Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Matthias Schubert
Felix Borutta

Munich, June 27, 2019

Managing Massive Multiplayer Online Games

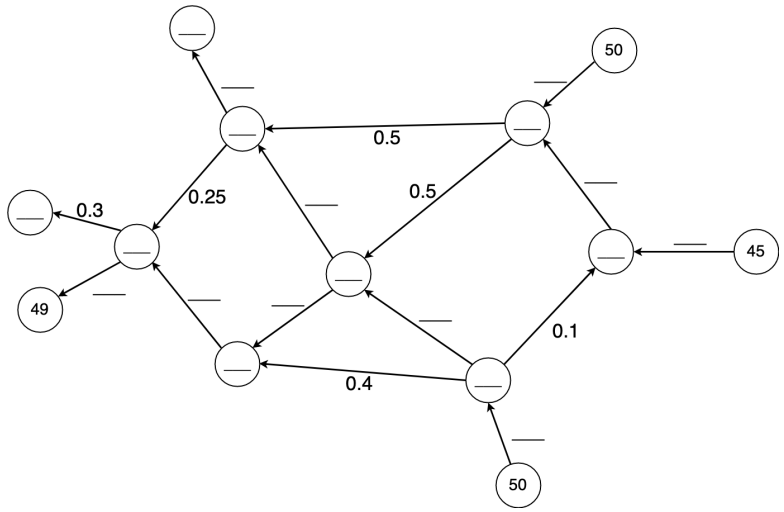
SS 2019

Exercise Sheet 9: Temporal and Spatial Behavior

The assignments are due July 3, 2019

Assignment 9-1 *Network Propagation Models*

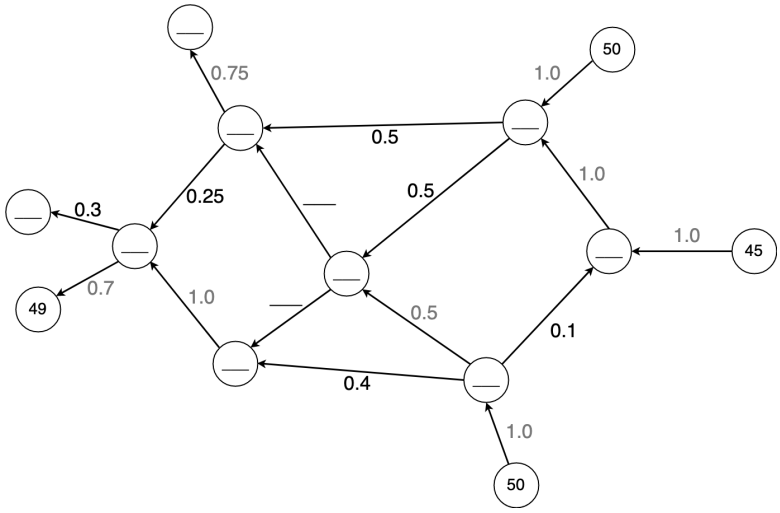
Let the following extract of a road network be given:



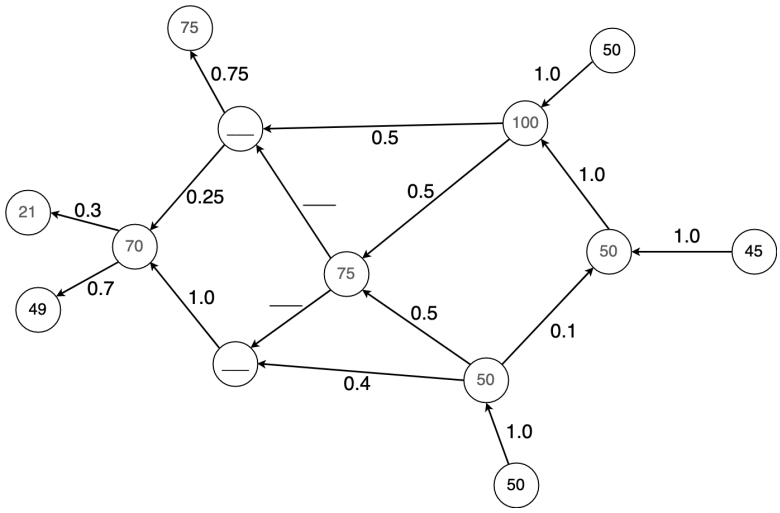
Let the nodes be intersections resp. dead ends and the edges denote road segments. Nodes are labeled with the number of entities, e.g., game characters in a computer game, and edge weights are labeled with the probabilities with which entities decide to travel across the corresponding road segment.

Calculate the missing transition probabilities and the expected number of entities that may be located at the intersections resp. dead ends. Transition probabilities per node shall sum up to 1.

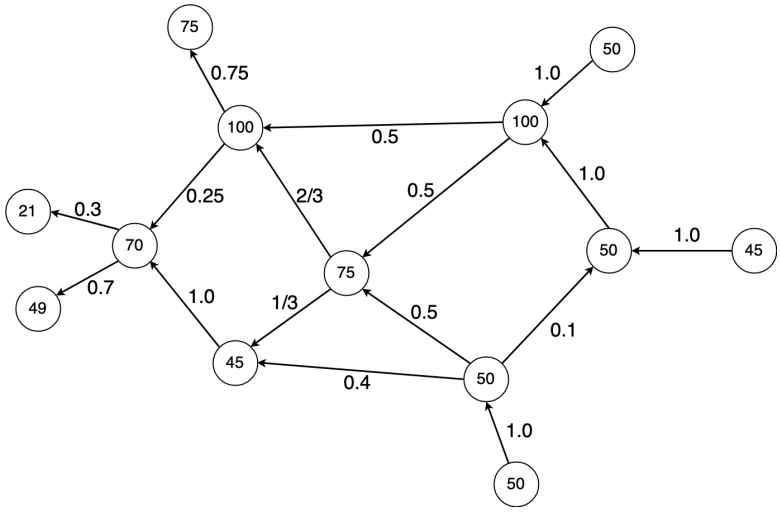
Given that the sum of transition probabilities per node shall be 1 we can fill in some of the missing transition probabilities as follows:



Given these transition probabilities, we can fill in some of the node labels by propagating the incoming entities through the network such that we get[1]:



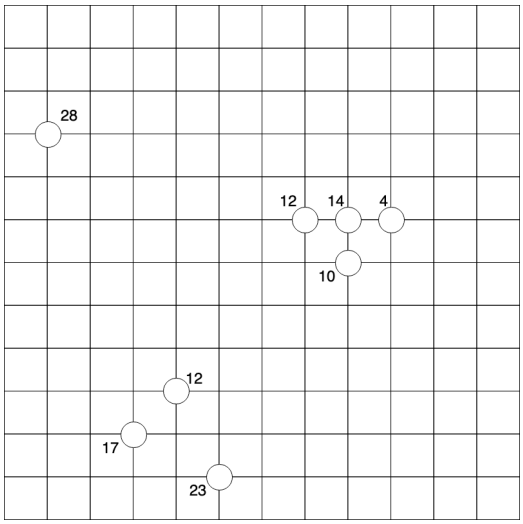
Given these information, we can finally calculate the missing values such that we get:



[1]You may want to note that all entities that enter this network should also have been propagated through the network entirely at some point which means that the number of entities at the source nodes, i.e., nodes that only have outgoing edges, should equal the number of entities in the sink nodes, i.e., nodes that do not have any outgoing edges. This for instance helps to infer the node weight for the sink node that is depicted in the upper left part of the network.

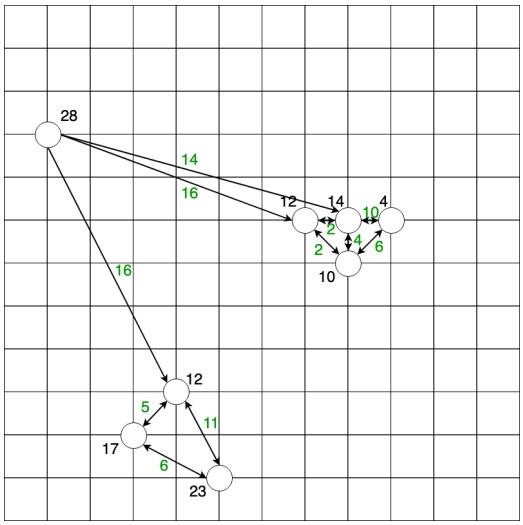
Assignment 9-2 *Spatial Outlier Detection*

Given the following positions as relevant spatial positions (e.g. starting positions in an FPS or frequent camp positions in an MMORPG). For every position a score that depicts semantic information about the quality of the position (e.g. the average number of frags in an FPS or the average count of EP/coins per hour in an MMORPG) is given additionally.

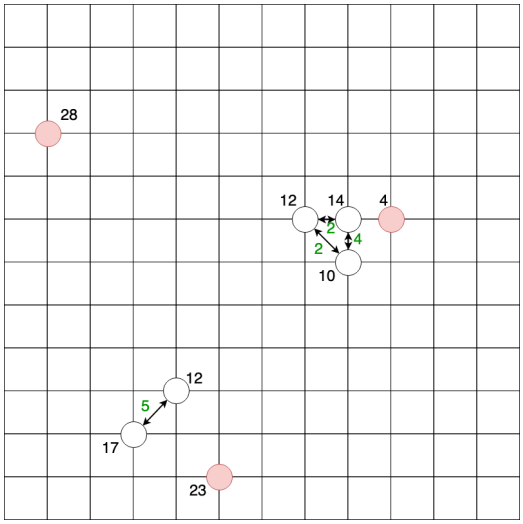


Find the three strongest outliers in this dataset by applying the Point Outlier Detection Algorithm with $k = 2$. Use the absolute score difference as weighting function.

The first step is to build the kNN-graph with $k = 2$:

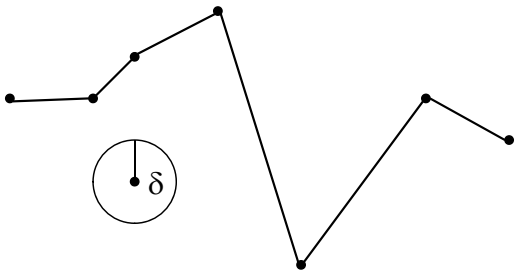


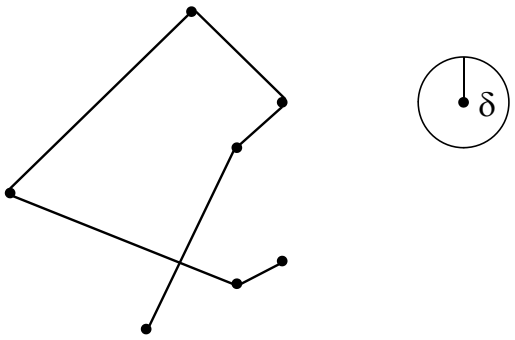
Then we iteratively remove the edge with the highest weight until we get the desired number of isolated data objects:



Assignment 9-3 *Compression of trajectories*

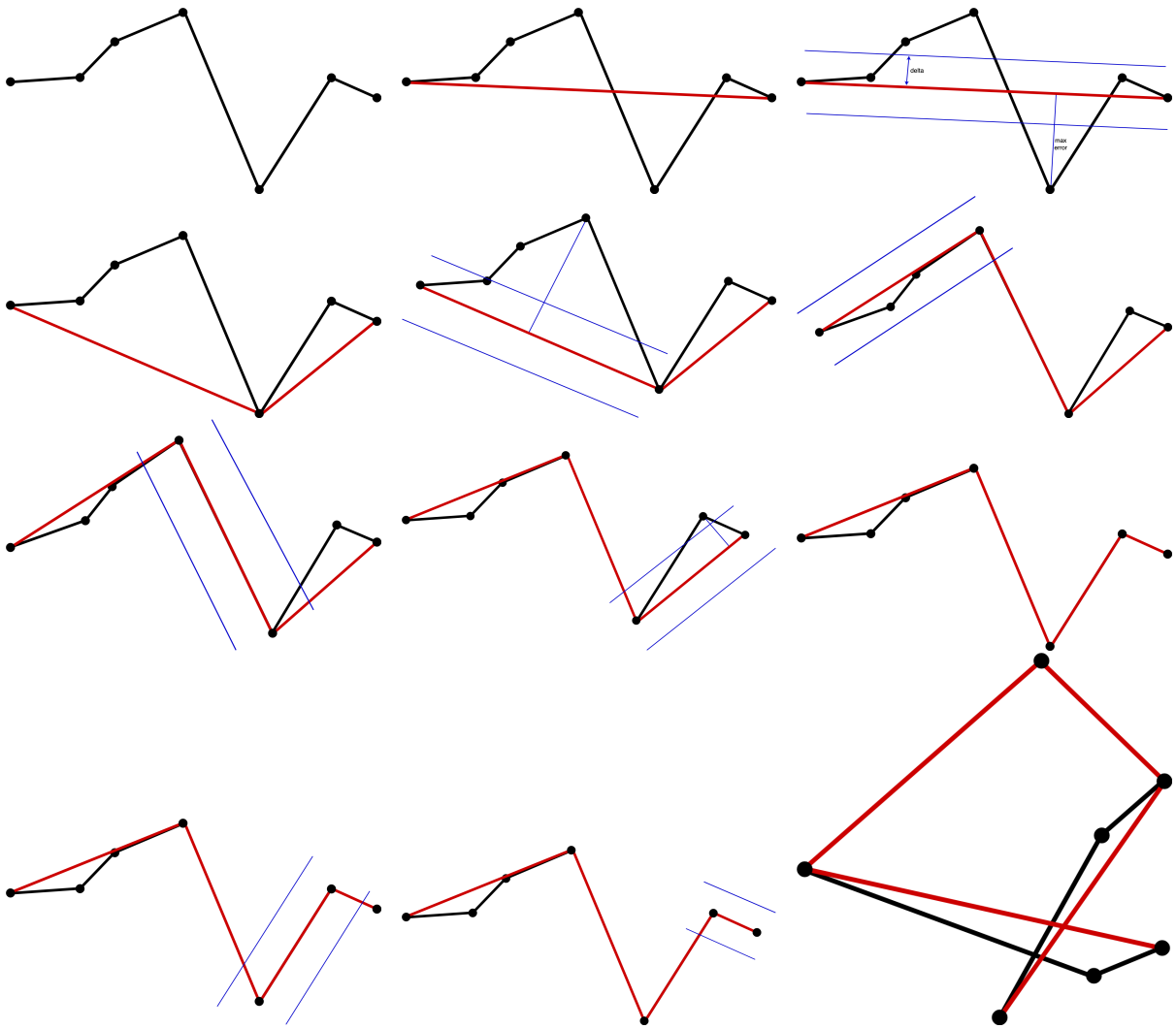
Approximate the following trajectories with the Douglas Peucker Algorithm.





The Douglas Peucker Algorithm can be found in Chapter 7: Spatial Analytics, slide 19. You can find the single steps for approximating the first trajectory with this algorithm in the figure below. For the second trajectory, I only added the final solution. Please note that the solutions are only sketches. The blue solid lines that are parallel[2] denote the considered δ-bounds and the blue solid lines that are perpendicular to the red lines denote the corresponding maximum errors.

[2]Though they are probably not exact, neither in terms of parallelism nor in terms of δ distance.


delta
max
error