

Generalized Subgraph Querying for Relationships

Michael Aiello

Inspiration:

Malcolm Gladwell's "Tipping Point" and Albert-Laszlo Barabasi's "Linked" discuss the importance and structure of connectivity in social and other relationship based networks. Gladwell explores the sociological importance of "connector individuals" while Barabasi defines properties of "connectivity" mathematically. I would like to create a tool which can be used to query a relationship based network for sub-networks by describing the structure of the relationships and properties of the nodes. The tool should essentially allow users to "draw" their description of connectivity and it will search an input network for this sub-network.

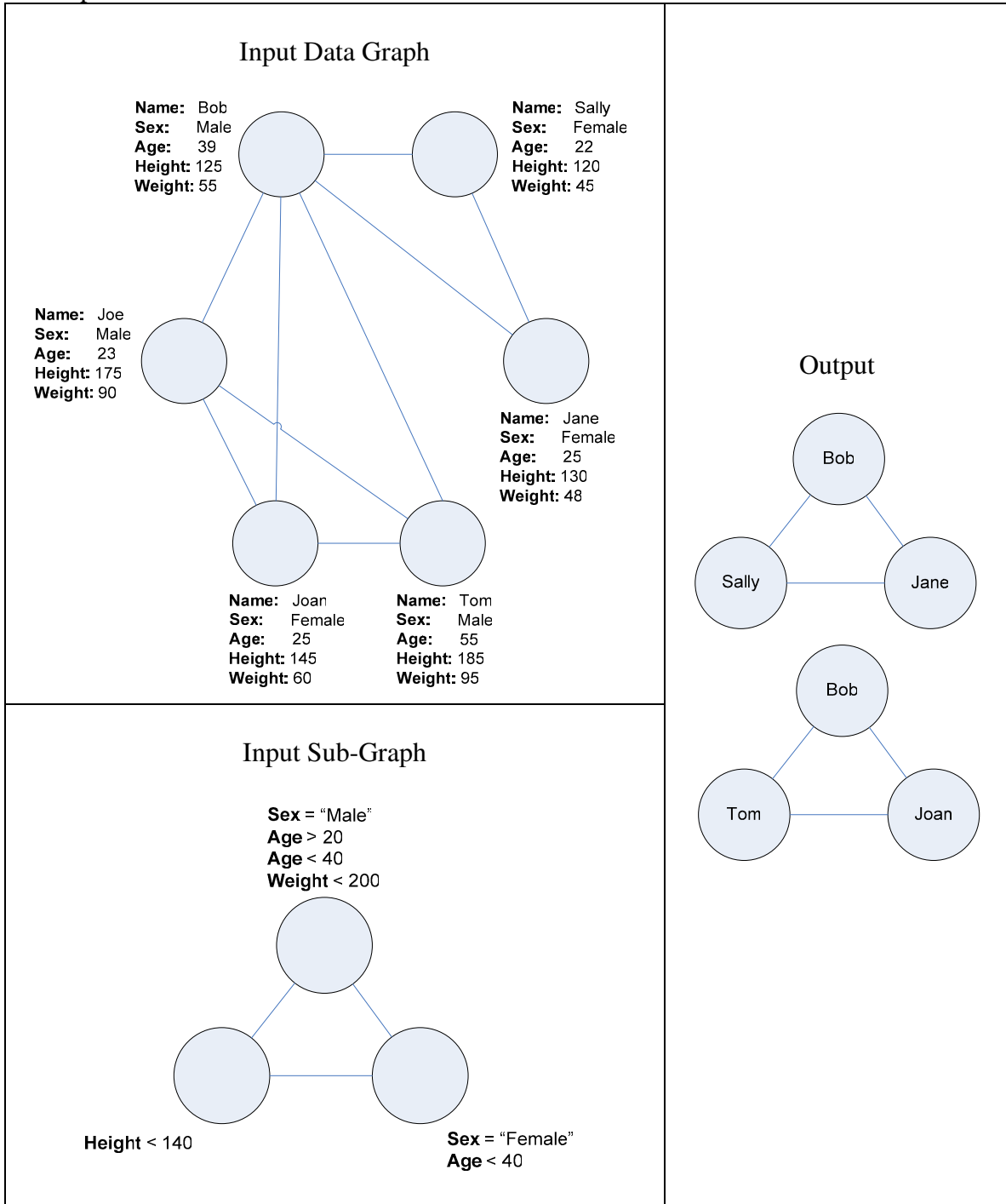
This type of tool has several implications, here are a few examples:

1. Marketing - being able to define and find subgroups of people that have the potential to be trend starters (see Tipping Point)
2. Anti-terrorism/correlation – Define what a terrorist network looks like based on relationships and properties of individuals, and search/retrieve them from existing communication databases. We now define the exact meaning of "chatter" and can describe it in terms of chatter based risk.
3. Financial fraud detection – "shapes" of complex fraudulent transactions are described and constantly searched for and reported.
4. Financial derivative searching – search a historical correlation graph for the shape of a derivative and return all such possible derivatives.
5. Biological Research – Searching graphs of chemical interactions for shapes of transaction patterns for classification/replication. (imagine defining a "NOR" gate with the tool, and searching a DNA strand for all instances)

Problem Definition:

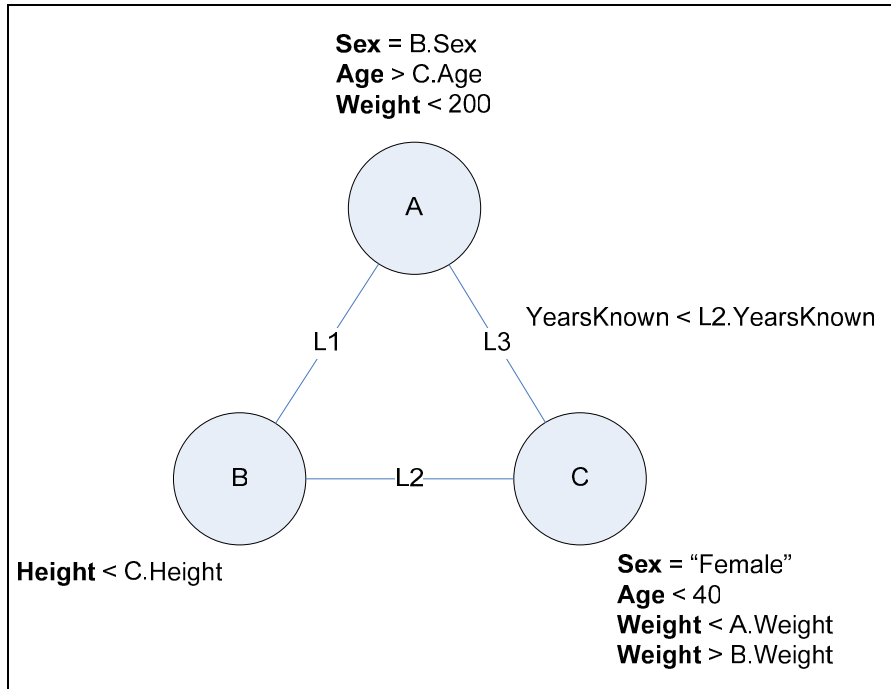
Consider a graph of connected nodes which have properties. A shaped sub-graph which has predicates on the properties of its nodes/relationships is given to the program as input and the program outputs all matching sub-graphs. What follows is the simplest type of query, an undirected graph with no predicates on the links between nodes. It also does not use abstract/random nodes or the description of an n-intermediate connection.

Example:

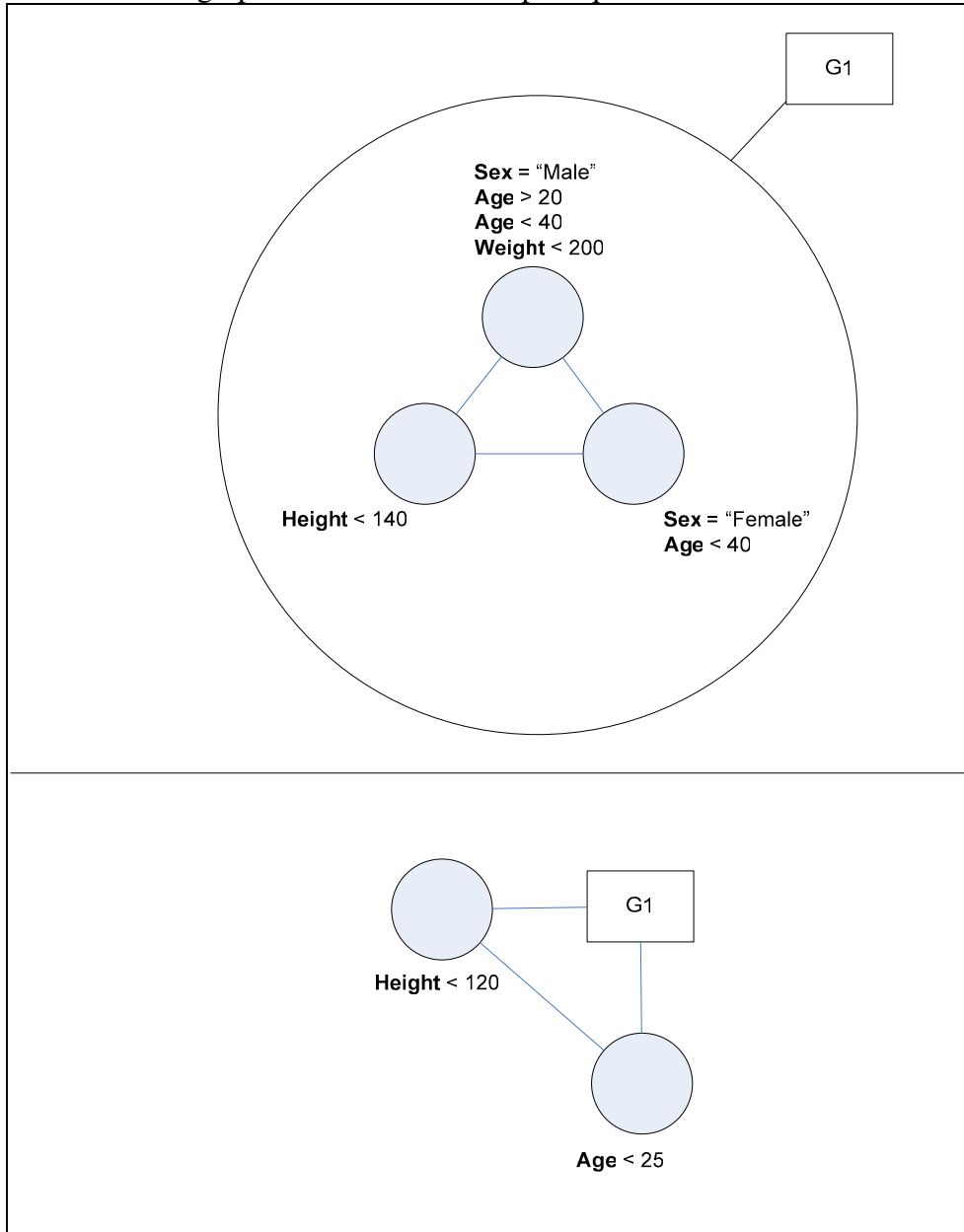


A more advanced query may include:

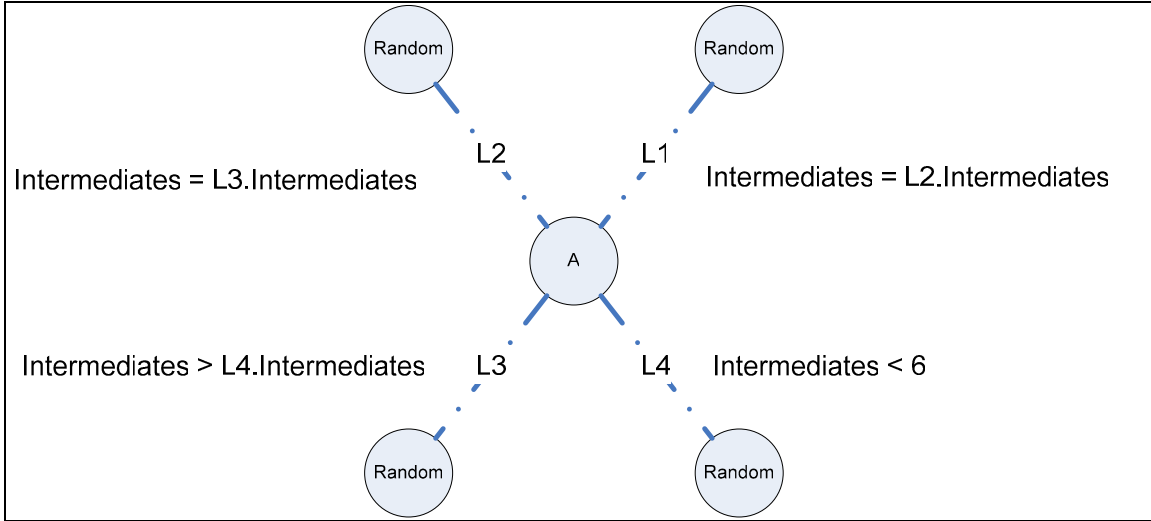
1. Define and search on other properties of connections themselves (weightings, direction, age)
2. Naming of nodes/links, predicates that involve relationships between nodes/links



3. Abstraction of sub-graphs to build more complex queries



4. Representative/repeater links (i.e. connected by at most 9 intermediate nodes, all satisfying a common predicate) and unification to the resulting number of intermediates
5. A representation of a random node



6. Ability to define “views” of graphs, and query them as if they were the original graph/Looping through result queries to find statistics/build derived graphs