Recommender Tips, Mortar and DocGraph

Jonathan Packer wrote on Mortar's blog about flexible recommender models. Jonathan articulates that "from a business perspective the two most salient advantages of graph-based models: flexibility and simplicity."

Some of salient points made in the article are:

- graph-based models are modular and transparent
- simple graph-based model will allow you to build a viable recommender system for your product without delaying its time-to-market
- Graphs can be visualized, explained, discussed, and debugged collaboratively in a way that sophisticated machine learning techniques cannot.

Jonathan ends with "My opinion is that the next big advances to be made in recommender systems will be made by combining automated tools with human—possibly crowdsourced—editorial judgement and writing talent. They will be made in finding more engaging ways to present recommendations to users than cloying sidebars and endlessly scrolling lists."

DocGraph
"The average doctor has likely never heard of Fred Trotter, but he has some provocative ideas about using physician data to change how healthcare gets delivered." This was from a recent Gigaom article. You can read more details about DocGraph from Fred Trotter's post. The basic data set is just three columns: two separate NPI numbers (National Provider Identifier) and a weight which is the shared number of Medicare patients in a 30 day forward window. The data is from calendar year 2011 and contains 49,685,810 relationships between 940,492 different Medicare providers.

The current DocGraph social graph was built in Neo4J. With new enhancements in Neo4J 2.0 (primarily labels), now was a good time to rebuild the social graph, add in data about each doctor, their specialties and their locations. Finally, I've added in some census income data at the zip code level. Researchers could look at economic indicators to see if there are discernable economic patterns in the referrals.

Recommendation Engine
The combination of the Neo4J social graph, the medical data and the capability to build a recommendation engine in Mortar makes a compelling use case. I believe that this use case will address Jonathan's premise that the new engaging recommendation engines can be built to help give patients a sense of which doctors are most respected by their peers. Additionally, the graph data could help hospitals understand the referral patterns associated with poor care coordination, and provide health IT startups with a map of the most plugged-in doctors in each city.

Next steps

Over the next couple of weeks, I'll be writing on how I used Mortar, Pig and Neo4J to build the updated DocGraph data set.