Exploring smitten kitchen: what's in a recipe?

11/12/2013
University of Maryland, College Park

Tiffany Chao (tchao@cs.umd.edu)

1. Dataset

smitten kitchen is Deb Perelman's food blog, where she shares shiny pictures and recipes. I occasionally stumble onto it when I'm looking for something new to try.

I was curious about what was in there (I've only read a few entries), so I grabbed all fall and winter recipes (58 fall, 32 winter, 5 shared), extracted ingredients from ingredient lists, standardized common synonyms (e.g. "salt" and "table salt"),¹ and created a 232-node, 7994-edge ingredient co-occurrence network.²

More about the network:

→ Nodes are ingredients.
   ↦ They have types (vegetable, herb, spice, etc.)³ and flavors (sweet, savory, or sweet + savory).⁴
→ Edges represent co-occurrences.
   ↦ An edge from A to B means A appears with B in at least one recipe. All edges are thus reciprocated.
→ Edge weights indicate how often A appears with B, relative to all A pairings.
   ↗ If A occurred once with B, twice with C, then A → B's weight would be 1/3, and A → C's, 2/3.

2. Findings

This is admittedly a tiny, biased sliver of the recipe universe, so generalizability is very questionable. It still provides reasonable suggestions for my future pantry, however.

¹ By hand.
² I also generated an ingredient substitution network, but there wasn't enough data to draw any meaningful conclusions.
³ By hand. Borrows loosely from USDA food groups and wikipedia herb/spice divisions (see, for example, the entry on tarragon). The final list: alcohol, cheese, dairy (other), egg, fat/oil, fruit, grain, herb, leavening agent, legume, meat/poultry, nut, salt/pepper, seasoning (other), spice, starch, stock, sweetener, vegetable, vinegar.
⁴ By hand. Describes whether recipes using the ingredient were always sweet, always savory, or sometimes sweet, sometimes savory.
Figure 1. The raw graph.

Darker, bluer nodes have higher betweenness centralities. Darker, thicker edges have larger edge weights (stronger connections). Larger nodes have higher degrees.

Removing node labels and varying opacity with edge weight helps tidy the bird's nest, but not by much.

The graph at left already excludes single-count edges, but it's still illegible. All I see: two dark blue nodes; maybe four other bluish nodes; some dark, but occluded, edges. Things to look into.

Everything’s one giant connected mass.
Figure 2. Grouped by flavor.

From Figure 1 to Figure 2a (top): group by flavor, add group labels, lightly curve edges, manually reposition. From Figure 1 to Figure 2b (bottom): enable group-in-a-box, combine inter-group edges, add labels, lightly curve edges, repeat Fruchterman-Reingold until it settles, filter out weaker edges, manually reposition.

Encodings are the same as Figure 1’s: node color for betweenness, node size for degree, edge color/width/opacity for edge weight.

# Favor flexible ingredients.

I couldn't find a sweet spot for type groups, but flavor groups made some sense.

The edges to sweet + savory (top) vastly outweigh the edges from sweet + savory: most of the flexible, high-betweenness ingredients (bottom, in dark blue) appear in both sweet and savory recipes (garlic and olive oil being notable exceptions). As someone who has yet to stock a pantry, this translates to very practical, if unsurprising, advice: start with ingredients that can be used in a wide variety of recipes.

# Avoid sweets (initially).

Sweet ingredients seem more isolated: though both sweet and savory's inter-group edges are comparably powerful, savory has stronger intra-group connections. Add that to sweet's lack of influential nodes, and it seems I should avoid sweet ingredients for now.
Figure 3. The 10 most "between" ingredients.

From Figure 1 to Figure 3a (top): filter to the 10 highest betweenness centralities, add node labels, manually reposition. From Figure 3a (top) to Figure 3b (bottom): remove weaker edges.

Nodes are arranged from highest to lowest betweenness centrality: for the pentagon, clockwise from the top; for the column, from top to bottom.

Encodings are the same as Figure 1's: node color for betweenness, node size for degree, edge color/width/opacity for edge weight. Note also the recurring one-sidedness in Figure 3b (bottom).

# Whatever you do, don't forget salt. And butter.

All these ingredients look like staples—and among them, table salt is king. Interestingly, unsalted butter isn't far behind. This might just be because Perelman bakes fairly often. Or because butter is delicious. It's hard to say.

All edges are all relatively weak (compare them to the dark edges from Figure 2). The strongest among these are all baking-related: this, again, may because of Perelman's baking proclivities.
3. Critique

NodeXL seems powerful. I just wish it didn't crash quite so often.
+ Simple things.
  - Loading data is fairly straightforward.
  - Graph metrics compute reasonably quickly.
  - Rendering delays usually make sense.
  - Auto-filling is extremely handy.
+ Customization.
  - I especially appreciate the graphical options. Though: please let me set discrete color steps.
- Complex interactions.
  - I still don't understand why auto-filling clears spreadsheet formatting. Is this an Excel thing?
  - I still can't predict whether a particular combination of layout, spreadsheet filter, graph filter, and grouping will work.
  - I still don't understand what group-in-a-box can or cannot handle. Collapsed groups don't obey boundaries. Grid layouts revert to one box. Edge filters don't affect aggregate inter-group edges. All of these weren't what I expected: some warning would've been nice.
  - Grouping and regrouping seems problematic. Repeating the same process (group by vertex attribute, recalculate metrics, auto-fill encodings, refresh graph) sometimes produces unpredictable exceptions (if I'm lucky).
- Cancels/undos.
  - Not being able to cancel an expensive redraw means I either (1) walk away for half an hour (and hope it doesn't die) or (2) kill the program (and lose work).
  - I'd really like undo/redo for the graph. Losing node positions—annoying, but recoverable. Losing a chain of exploratory actions—there's no getting that back. I know this is an Excel thing, but it's still painful.
  - Generally, the above complexity wouldn't bother me so much if I could confidently undo/redo.