ToMeto: a Networks-based Approach to Recipe Recommendation

Albert Ge, Matthew Jin, Jonathan Joo, Boyu Tong - California Institute of Technology

Caltech

Introduction and Motivation

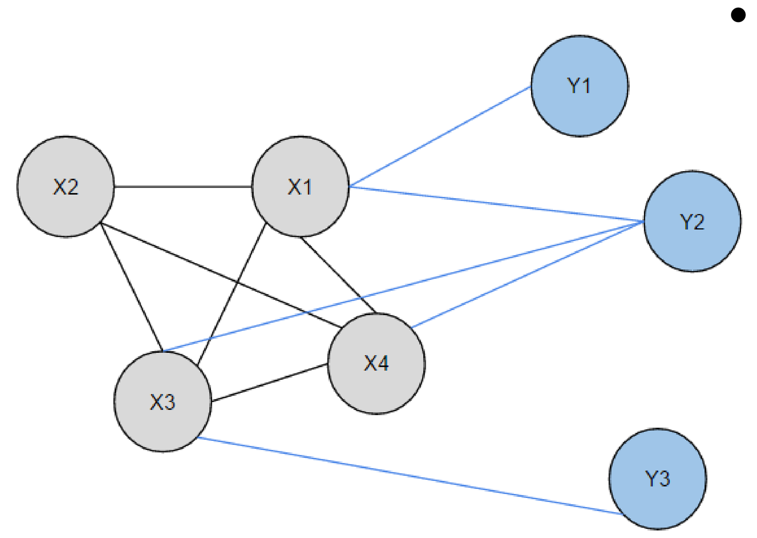
- Inspiration from several papers studying flavor and ingredient networks.
- Wealth of recipes stored in free online recipe databases to be gathered and analyzed.
- Wanted to create a food web application applying these networks to improve home cooking.
- Main Idea: **use ingredient network** extracted from a database of cooking recipes to **suggest new ingredients** to add to an existing recipe.
- Graph consists of ingredients, connected if they appear in the same recipe, weighted according to the number of occurrences.

Data Processing

- Data was scraped from two popular online recipe websites: **Allrecipes**, and **NYT Cooking**. 52195 and 13768 recipes were downloaded, respectively.
- Decided to focus efforts on the smaller NYT dataset, as the data format was easier to parse and work with.
- Parsed each downloaded recipe, extracting the title, ingredients, and cooking instructions.
- Improved data quality by **mapping** "low-quality" or "noisy" ingredient entries to actual ingredients:

medium-large shrimps (about 14 to 16) → shrimps

- Reduced the ingredient count by over 9000, and ended with 1070 most common ingredients that were eventually used in the network analysis.
- Methodology for mapping the ingredients involved **filtering** out the linguistic fluff from ingredients: parentheses, commas, plurality ("*shrimps*" vs "*shrimp*").



- **Supplemented** network with the Allrecipes recipes mapped to the NYT ingredients. The resulting network consisted of only NYT recipe ingredients but had edge weights based on both datasets.

Algorithms

Preliminary (naïve) approach:

- Weight of an edge (*a,b*) is the number of recipes the pair appears together.
- For each ingredient *a* in the recipe, compute, for each of its neighbors *b* outside the recipe: $\min\left(\frac{w(a,b)}{degree(a)}, \frac{w(a,b)}{degree(b)}\right)$

PMI (Pointwise mutual information):

- PMI of two ingredients *a,b* given by: $\log \frac{p(a,b)}{p(a)p(b)}$
- **Edge weights** between ingredients are their **PMI scores**. In this way, ingredients that are highly correlated (are in many recipes together) will have a stronger weight. Weights range from -∞ to +∞. Ingredients that never share a recipe do not have an edge between them.

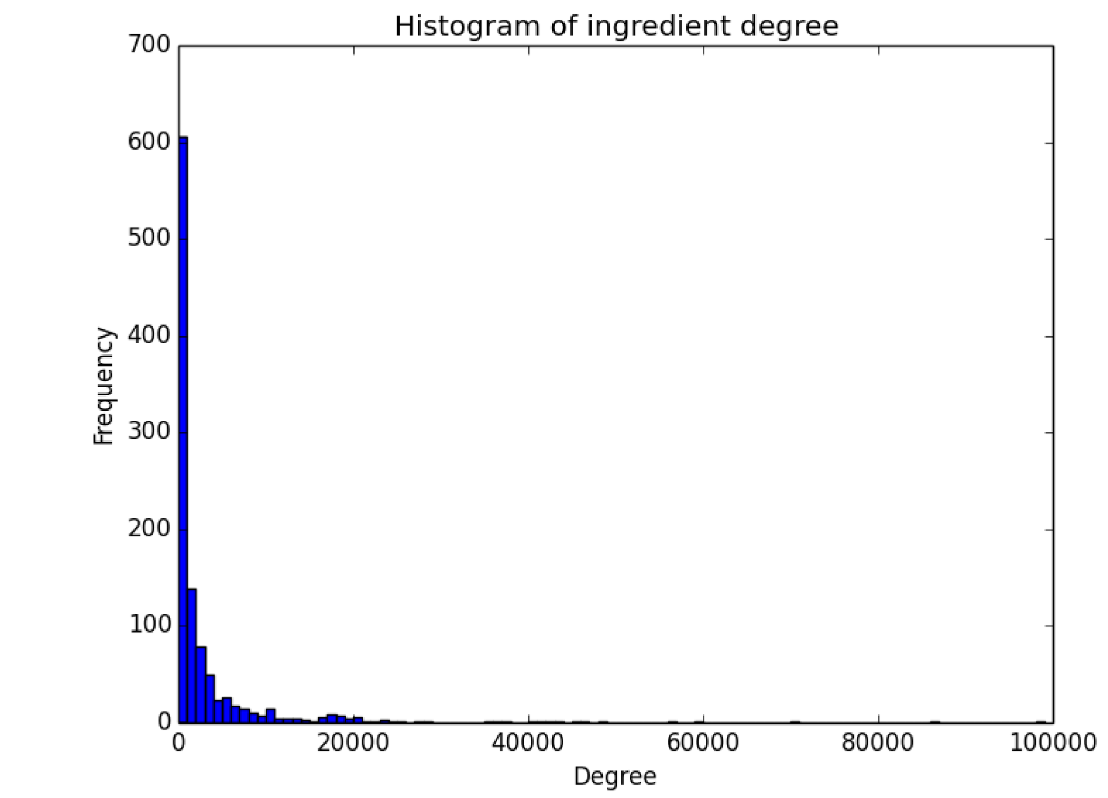
PMI (Pointwise mutual information) variants:

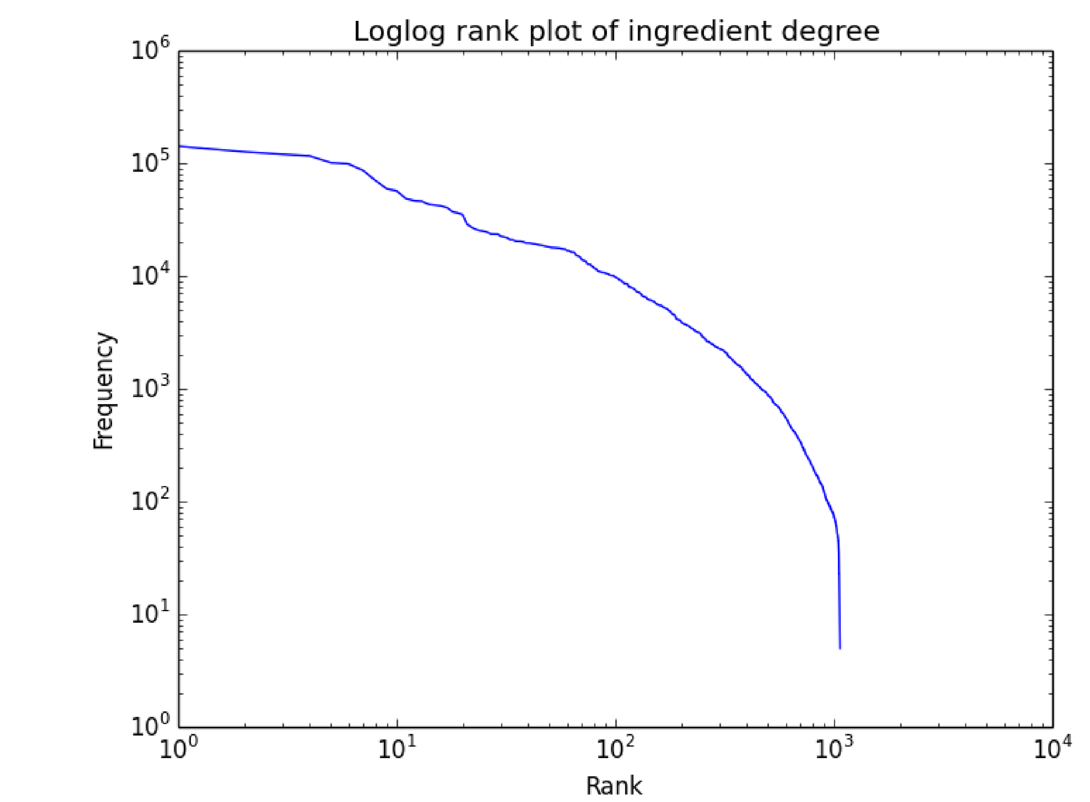
- Normalized PMIs (nPMI) – Normalize by degree to find "critical" ingredients.
- Weighted sum of PMIs (wPMI) – Weight by # of edges to recipe ingredients.
- Generalized PMI - 3 or more nodes (gPMI) – Extend PMI to 3+ nodes.
- Minimax PMI (mPMI) – Use minimax to find the most compatible ingredients.

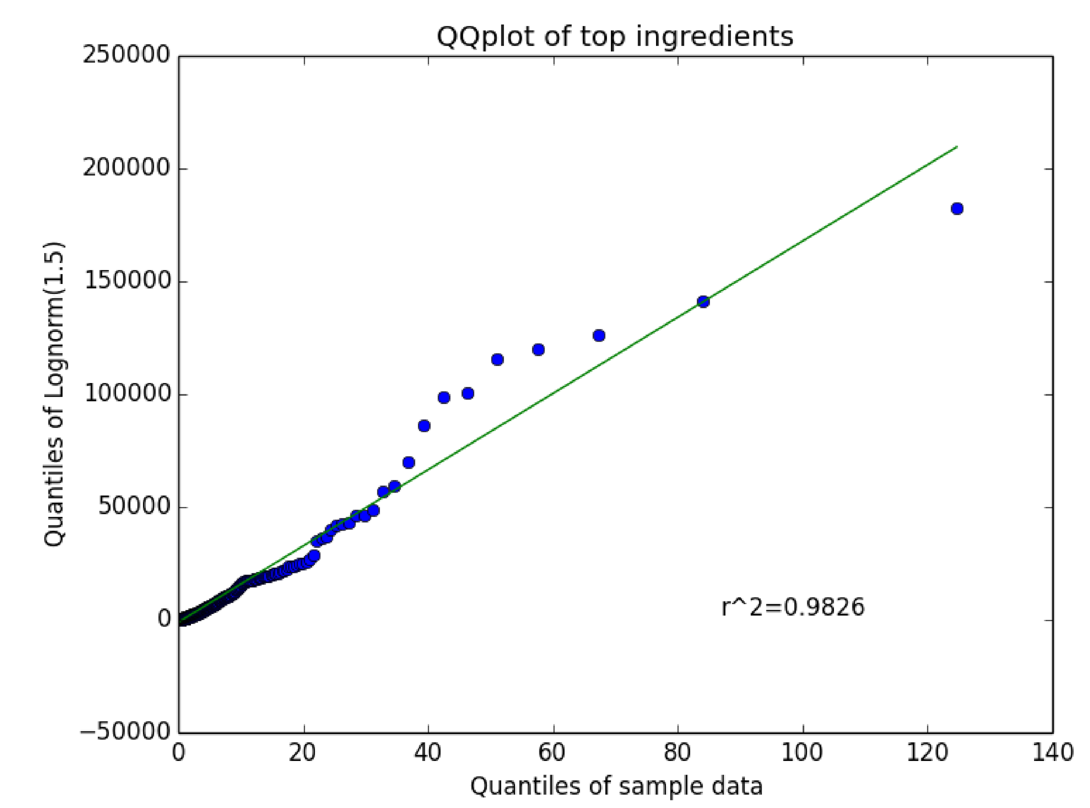
Analysis of Algorithms

- Required some objective way of comparing algorithms.
- Split the database of recipes into **learning** and **testing** sets.
  - Inspired by **machine learning** methods of testing.
  - Construct the ingredient network and run the algorithms using only the learning set (approximately 70% of the data).
  - For each of the recipes in the testing set, remove ingredient**s** and apply the recommendation algorithm on the remaining ingredients.
  - Does the algorithms suggest the missing ingredients?
- Testing still in progress.
- Rudimentary results: wPMI, sPMI seem to be the best algorithms.

Preliminary Network Analysis







References

1. Ahn, Y., Ahnert, S., Bagrow, J., and Barabasi, A. Flavor network and the principles of food pairing. Bulletin of the American Physical Society 56 (2011).
2. Teng, C.-Y., Lin, Y.-R. & Adamic, L. A. Recipe recommendation using ingredient networks. (2011). ArXiv:1111.3919 [cs.SI].