1 Information Filtering System Design

Information filtering systems automatically distinguish relevant from irrelevant items (emails, news articles, intelligence information) in large information streams. They typically use a classifier trained on relevance feedback from past items. However, when filtering for new users, or when item contents or user interests change, sufficient training data may not be available. In such "cold-start" situations, it may be beneficial to actively explore user interests by forwarding those items whose relevance we wish to learn, but too much exploration degrades short-term performance. This is an example of the so-called exploration vs. exploitation tradeoff.

In this project, we plan to implement an algorithm for the information filtering system using dynamic programming, similar to [1]. Dynamic programming is very computational intensive and typically suffers from the curse of dimensionality. We plan to implement efficient c++ code such that it can solve low dimensional dynamic programming efficiently. In order to solve the dynamic programming, we will first formulate the problem and write down the Bellman equation, then solve it using backward reduction. Some techniques that we plan to speed up our code are:

- Vectorization: Given the value function of T, different state space in T-1 are actually independent with each other. Thus using vectorization could speed up our calculation.
- Memory Alignment: For example, when solving a two dimensional dynamic programming, how to arrange the state space so that we can minimize our chance for cache miss?
- OpenMP: This is similar to the vectorization. Key idea here is different state space is independent.

Along the way, we will also explore various complier flags and try to benefit from the profiling reports.

References

[1] Xiaoting Zhao & Peter Frazier, Exploration vs. Exploitation in the Information Filtering Problem, arXiv.org.