Pixie-Inspired Algorithms

Pixie-inspired algorithms are algorithms that leverage random walks to conduct their calculations. In the case of recommender systems, you can have a system that recommends items to users based on this random walk pixie-inspired algorithm. These algorithms are especially useful when navigating large and complex graphs is necessary. While not very complex, pixie-algorithms can be very powerful.

A random walk algorithm essentially tries to capture the concept that an agent walks randomly through a graph based on probability. The agent at any given node may choose to walk to any one of its edges and/or adjacent nodes based on equal probability. In the context of recommendation systems, these nodes can be items (movies, books, products, etc) and the edges between them can be relationships between these items (similarity, user interactions, etc). By using a random walk algorithm, or pixie-inspired algorithm, you can explore the network in a random manner, allowing you to suggest items that are connected with a given user's interest even if it isn't directly related.

In this kind of algorithm, the random walks help in identifying relevant recommendations by simulating the user's interactions, or potential future interactions. For instance, when a user interacts with an item, that item is now an interest for the user and can be a potential starting point for the pixie algorithm. The algorithm then randomly starts interacting with similar items by choosing a random related item to visit, and repeating the same process from that item for an arbitrary amount of steps. The algorithm keeps track of the items that it visits. At the end of the algorithm, you can see which items were visited the most during the runtime, and these can then be recommended to the user. This can help simulate the user's actual exploration of the item space, and is unlikely to explore too far out of the user's preferences (depending on what iteration limit you set). The nice thing about this algorithm though is that it can easily extend to related interests. For instance, if the algorithm starts at an action movie, it is unlikely that it will visit a RomCom the most, but it may visit a thriller many times or another action movie. This can help diversify the recommendations without diverging too far.

This kind of algorithm can be utilized in the real world in many different ways. One easy and obvious application is online retail sites and streaming services. These types of sites and applications have a large variety of items with a large variety of categories of items. Interestingly, each category may have a more strongly related category than others, like action and thriller movies are strongly related, but action and romcom are not. Similarly with online retail, a person who buys a batman action figure may be interested in a batman comic, but would not be interested in a Barbie doll/action figure. Even though a Barbie doll and Batman action figure may be in a similar category, they would not be strongly related. So the random walker is unlikely to visit it more than it visits a Batman comic or something similar.

Another application that may not be as obvious is its use in recommending webpages. We have been learning in class about Trust Rank and Page Rank, which use a similar random-waker approach to evaluating the importance and/or trustworthiness of a webpage. These aren't directly recommendation systems, but the underlying concept is similar and demonstrates how powerful and flexible this kind of algorithm is. Any application that involves large and interconnected datasets could likely implement random-walker algorithms like pixie in some interesting and useful way.