datablick

RECOMMENDATION ENGINE FOR TABLEAU SERVER

A recommender system, which applies machine learning techniques - a type of statistical optimization - supports users to find workbooks of their interest by analyzing their past interactions (behaviors). With a recommender system, users provide an implicit binary feedback (score) for the recommender algorithm by clicking on a recommended item or not.

Personalized recommendations help users to find relevant information in an efficient way. To maintain user retention, and mitigate churn, the recommendations must be updated on a frequent cadence. Effective recommender engines have proven return in investment at the Enterprise as they add significant value to the end user.

To obtain the recommendations we rely on two sources. These can be divided into two groups:

1) (Explicit) Features, e.g. tags (meta-data); and

2) (Implicit) Interactions, e.g. ratings (historical data).

METHODOLOGY

The prediction problem is a very sparse user-item matrix. This matrix can be tested/queried for interactions. A value of zero in this matrix (item has not had an interaction) against a value of one or more becomes the target variable, where the goal is to predict missing probabilities.

The recommender system uses two approaches to find a solution:

1) (Meta-data) Content filtering; and

2) (Historical data) Collaborative filtering (e.g. Nearest neighbor clustering)

-hitherto mentioned with an output of recommendations.

MODEL EVALUATION

To assess model effectiveness and efficiency, evaluation of the model was performed - by implementing offline tests.

The offline test was time-banded, and input data was split into training and testing sample sets. A predicted score was calculated for probability for every cell of the user-item matrix.

Thus computing interactions results in an accuracy rate. This accuracy rate can be compared to a baseline (naïve) accuracy rate, e.g. recommendations of the top-N most popular items.

CHALLENGES

Processing scalability and robustness

Accessing and processing hundreds of thousands of items, users, and hundreds of millions of relationships with additional metadata about them like tags, languages, etc.

Potential data loss during processing.

Model extensibility

Integration of new interactions should not require model rebuild.

Implementation of new features into the model, such as Recurring Neural Network (Deep Learning Engine training).

Long tail problem

Defeating the "cold start/long tail" problem (i.e. limited or no historical interactivity).

OUR SOLUTION

Architectural overview

The architecture of the recommender system is shown at the figure one.

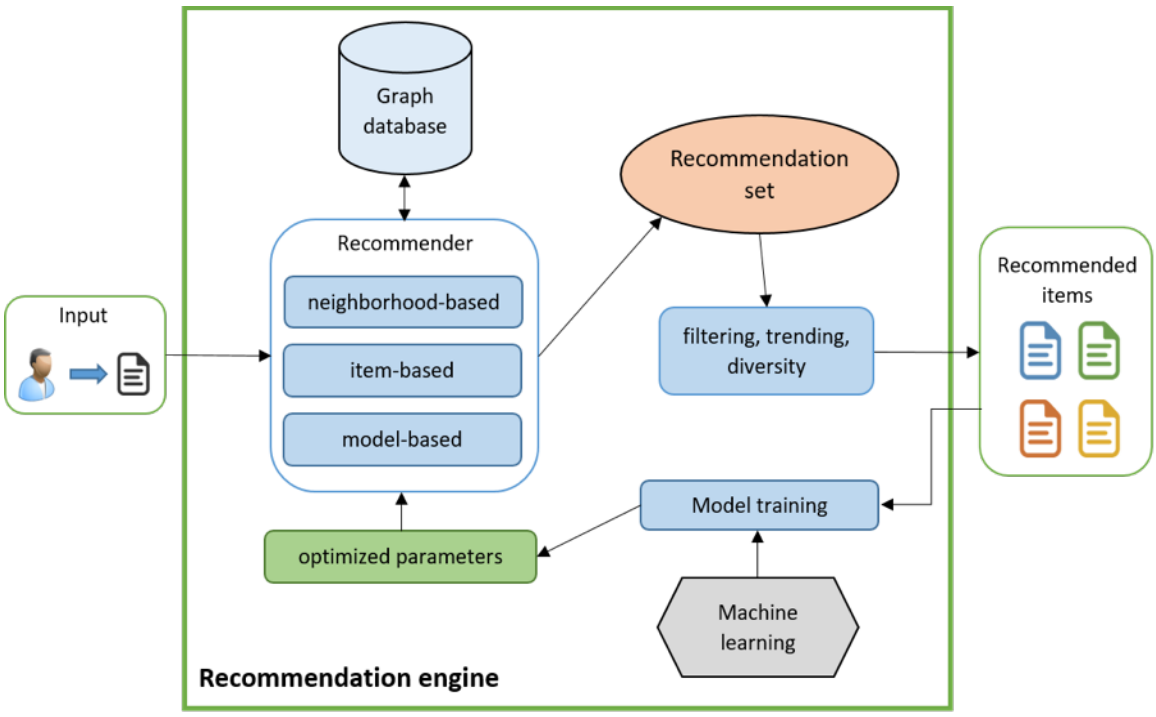

Figure 1 Architecture of recommender system

Database management

In a network users are connected to each other via the items they seen and via the features (e.g. tags) of those items. To address the challenges mentioned hitherto, a solution to use a graph database was applied.

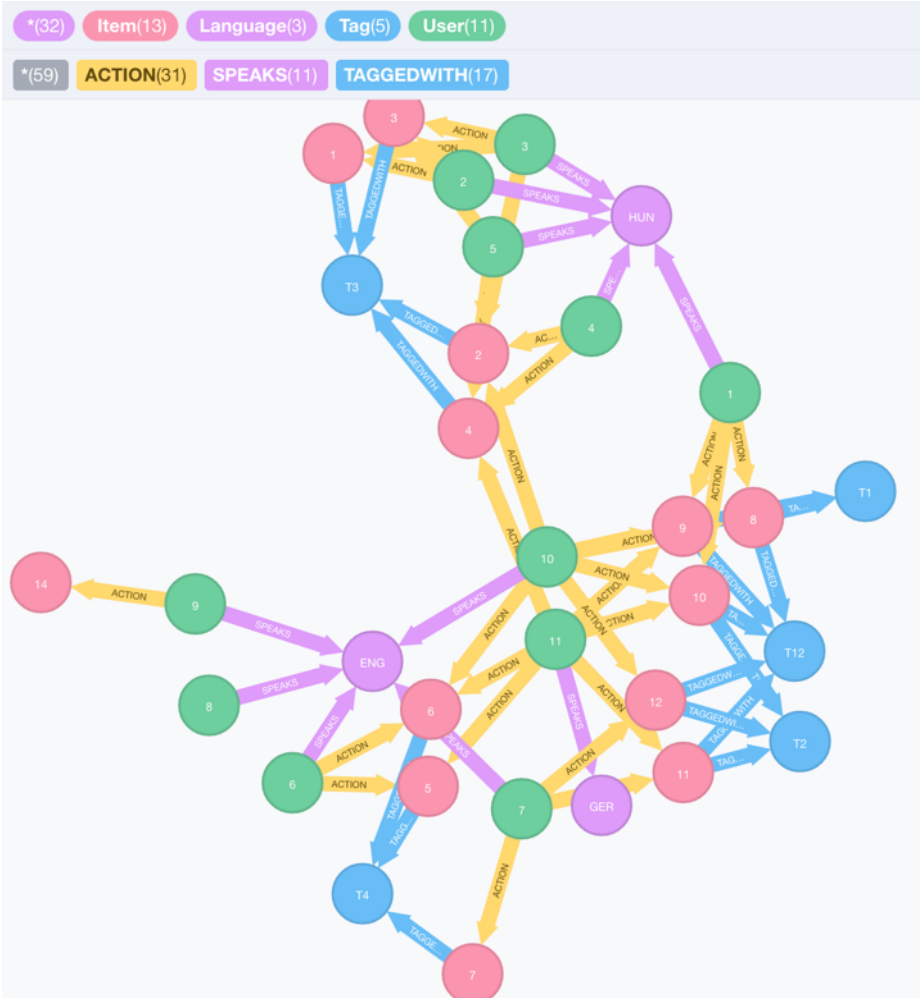

Figure 2 Structure of connected data represented by a complex graph