Improving the relevance of personalized marketing using graph databases

Rex Wagenius and Alexander Wilczek

Royal Institute of Technology KTH, Stockholm, Sweden

Abstract. Today many websites use personalized marketing, however, as most have experienced, this is not as personalized, nor intelligent, as it could be. A common problem is that the product or service marketed is something that the user already have bought. In this report we suggest a solution to improve the relevance of the personalized marketing, increase the intelligence of the systems used, as well as provide an expansion of today's personalized marketing using graph databases.

The idea is to create one graph for each user that links to areas of interest. Then by linking each area of interest to products and services relevant to that area the precision of the personalized marketing can be improved. To further improve this, areas of interest can be linked to other areas that are either similar or that several users of this interest have in common. Even though relational databases can be used for this a graph database is more suitable to keep the information relevant over time and thus the ever-changing interests of a user can be kept up to date.

Keywords: Graph Databases · Personalized Marketing · Column Store · Interest Score · Graph Records

1 Introduction

Most people today have probably encountered personalized marketing in one form or the other. However, sometimes personalized ads appear irrelevant, or even incorrect. In short, the personalized ad generation-systems of today appear to be lacking in intelligence. A simple field test shows that today's systems are generally based on cookies and browser history which is non-optimal.

However there are ways to improve such systems. By implementing and using graph databases in order to better monitor and match data such as personal interests, prior purchases as well as time spent on a web page; we not only lessen the importance of cookies in this type of system but also significantly increase its scope of interest.

2 System Model

2.1 Defining the graphs

The idea is to create a system where each user has their own graph that represents their areas of interest. Each node in the graph represents an area of interest, except for the root which represents the user itself. As one traverses the graph outwards the areas become more specific. The reason behind letting each user have their own graph, instead of one large graph for all users, is that it is easier to track user specific data for each area of interest. Thus allowing us to store said data in the nodes rather than the edges.

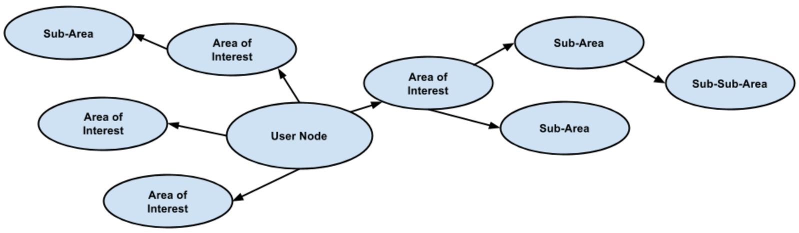

Fig. 1. Visual representation of a user graph.

Similarly, companies define interest graphs of their own. These graphs can be added to user graphs upon shown interest. It is however worth noting that the companies are not free to define graphs entirely as they please. When a company wishes to define a graph, it is presented with an empty graph template that needs to be filled in with certain interest data. Such a template always begins with an empty node and can be expanded as further areas of interest are added. Forcing companies to abide by the same graph format is done in order to ensure graph structure consistency as different looking graphs (in terms of the data stored) can cause issues. Finally, once a company graph has been submitted to the system it needs to be internally evaluated and approved before being incorporated in the system. The reason for this is to ensure that the graphs actually are relevant to the company, of the appropriate size and without duplicate nodes.

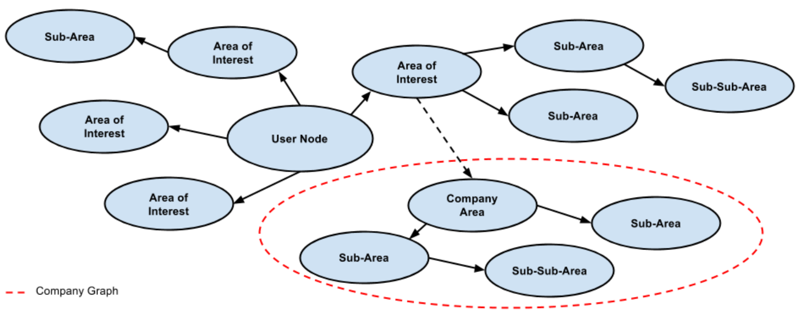

Fig. 2. A predefined company graph being added to a user graph.

Limiting the graph size means that a company can submit several graphs relevant to different parts of their business rather than have one larger graph. Dividing the business into several graphs provides the company with the opportunity to link certain sub-pages with certain parts (graphs) of their business. Once a user visits and thus shows interest in a website relevant to a company, the company's graph is added to the users as a sub-graph. After the company graph has been connected to the user graph, relevant ads can be generated and presented to the user.

Nodes & Edges. As previously mentioned the data stored in the user graph nodes is user specific. As a company graph is added to a user graph, the metadata of each respective node is automatically entered. A suggestion of node metadata is presented below:

- Last visited: Date
- Latest purchase: Date
- Re-advertisement: Date[1]
- Number of purchases: Integer

However, a system using simply the data above would have no means of determining a user's level of interest. In order to address this issue, we introduce an interest score stored in the edges. An algorithmic suggestion for how this calculation can be done is presented below:

$$R_i = \alpha \cdot \sum_{i=1}^{k} R_i + \frac{T}{\beta} + \frac{P}{\gamma} + \frac{V}{\delta}. \tag{1}$$

In this suggestion, R_i is the interest score of a related area. A related area is defined as the previous nodes in the path. T is the time spent on the website in minutes, P is the number of previous purchases and V is the number of sub-pages visited. α, $\beta\gamma$ and δ are constant factors whose value intervals are defined below:

$$\alpha = [0.2, 0.8], \beta = [5, 15], \gamma = [0, 10], \delta = [20, 60]\ ^2. \tag{2}$$

2.2 Column-Oriented Storage

A study conducted by Bleco and Kotidis has shown that storing the data within columns rather than the traditional rows significantly increases system performance for this type of systems. By storing data in this fashion the system only needs to read the rows for relevant fields rather than read all data for each entry regardless of its relevance. This reduces the worst case scenario in regards of field reads and removes the need for expensive joins between tables[1].

[1] The re-advertisement date is the date which a product or service is supposed to be advertised again after a purchase.

[2] The values of these constants are merely an example and are, as such, subject to change.

2.3 Materialized Views

In regular row-based databases, materialized views are a powerful tool used for improving system performance for frequently used queries. Graph databases are no different, Bleco and Kotidis presents an implementation of materialized views for graph databases. In this implementation so called graph records consist of measurements, which are values stored in the nodes, and a bitmap that stores booleans stating whether or not a specific edge exists in this record. Finally, certain common queries can be pre-calculated and stored within views, an example of such a query could be "*Does a path exist between nodes A and F?*". Doing this helps minimize query run-times[1].

Table 1. Example graph records with pre-calculated views.

	Measurements			Bitmap			Views		
r_i	m_1	m_2	m_3	b_1	b_2	b_3	v_1	v_2	v_3
r_1	4,0	2,3	NULL	1	1	0	0	NULL	0
r_2	5,5	6,0	6,4	1	1	1	1	6,4	1

3 Conclusion

By taking all of the above into account and designing a system accordingly, we can conclude that although the system will still not be perfect, it should provide a significant intelligence increase compared to today's systems. Furthermore, the highly adaptable, more modular nature of the system also means that it can be even further developed and perfected for the future. This not only benefits companies with regard to marketing and sales, but also the consumer as users no longer will have to suffer through unnecessary and redundant ads.

4 Strengths & Weaknesses

While this system should prove to significantly increase the intelligence and quality of personalized marketing it is limited by certain factors. Considering how it is infeasible for a system to gather information about every company there is, the system is dependant on the companies supplying relevant data. Therefor, a symbiotic relationship between the companies and the ad-distributor, where the companies supply information and the ad-distributor presents relevant ads to the user, has to be established. However, the abovementioned limitation also proves to be one of the system's greatest strengths. By having companies supply the system with relevant data, most redundancy in the system data is removed.

References

1. Bleco, D., Kotidis, Y: Graph Analytics on Massive Collections of Small Graphs. 17th International Conference on Extending Database Technology (EDBT). Athens University of Economics and Business (2014).