Identifying Connectors, Mavens and Salesmen among users of Last.fm

Machon Gregory (mbg [at] umd.edu)
Roman Stanchak (roman [at] cs.umd.edu)
CMSC 734 Application Project
February 26, 2007

Introduction
In Malcolm Gladwell's book, *The Tipping Point*, he describes three types of people that have the power to cause "social epidemics." Social Epidemics a phrase coined by Gladwell refers to the rapid transition from something being unique to it being common, the point where the transition occurs is the tipping point. The three types of people that have the ability to cause this change are Connectors, Mavens, and Salesmen. Mavens are those who seed new ideas and trends. They are characterized as obsessive gatherers of information who can instantly recall the pros and cons on a given topic. They may not have wide social networks or be particularly charismatic, but their expert opinion can be trusted. Salesmen, on the other hand, use their charisma and negotiation skills to influence others within their own social group. Finally, Connectors are those who carry ideas between disparate social groups. They typically have wide social circles that tend to bridge many different types of groups.

The goal of our application project is identify these types of people in the Last.fm community. Last.fm is a social networking site that is centered around music. The Audioscrobbler system is a database that tracks the listening habits of the users in the Last.fm community. It provides web services to access data such as users profiles, a list of friends, recommendations received, recent tracks listened to, a users top fifty tracks, etc. The goal of our application project is use these web services to collect data about the Last.fm users that will allow us to identify the Connectors, Mavens, and Salesmen using visual analysis tools.

Data Set
The data set consists of 809 users, a small subset of the overall Last.fm population. For each user, the following information was collected

<table>
<thead>
<tr>
<th>Data Feed</th>
<th>Fields Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Profile</td>
<td>Username, Gender, Total Tracks Played</td>
</tr>
<tr>
<td>Manual Recommendations</td>
<td>Sender Username, Recommended Song (Artist,Title) *</td>
</tr>
<tr>
<td>Friends List</td>
<td>Friends added to the user's profile (Username)</td>
</tr>
<tr>
<td>Recent Weekly Track Charts</td>
<td>Tracks (Artist, Title, Number of Plays)</td>
</tr>
<tr>
<td>Recent Loved Tracks</td>
<td>Recently Loved Tracks (Artist, Title)*</td>
</tr>
<tr>
<td>Top Tracks</td>
<td>Most played tracks (Artist, Title, Number of Plays)**</td>
</tr>
</tbody>
</table>

*Last 10
**Top 50

In order to classify each user as either a Maven, Connector, or Salesman data had to be collected from the web services and processed to be used by the visual analysis tools. For each user a user profile was collected along with a list of friends, their recent weekly track charts and loved tracks, and their top 50 tracks. For each of the friends in the user's friends list we collected the recommendations they received, the recent weekly track charts and loved tracks, and their top 50 tracks. We assumed that a user only makes recommendations to their friends. The data was processed by tallying the number of time a users recommended an artist to a friend and whether the receiver ended up listening to the recommended song. In order to determine whether the receiver listened to it, we scanned the receiver's recent weekly tracks chart, recently loved, and top 50 tracks list. This created three metrics that could be used to identify a user as a Maven, Connector or Salesman. Unfortunately, two of the calculated metrics were unusable. The tally of the recently loved tracks and top 50 tracks that were recommended by the user were zero for all 809 users that we collected data for.
We hypothesized that Mavens could be identified as users who (1) tended to listen to music before their friends, and (2) when they made recommendations, their friends tended to listen to them. Connectors would be identified as those users who connected disparate groups of friends. Finally, Salesmen would be identified as users who made lots of recommendations within their social group.

Analysis

Our analysis used Spotfire and Treemap. We originally planned to use HCE, but we became frustrated with its tendency to crash.

In our attempt to examine identify the Mavens and the Salesmen we looked at a Treemap of all the recommendations. The recommendations are labeled with name of the receiver, grouped by the sender, and the receivers that listened to the track were colored green. From this Treemap we can see the number of recommendation that were sent out was not correlated to the number of recommendations that were listened to by the receiver. We can begin to identify the potential Mavens and Salesman, but we can not come to any conclusions with out knowing the expertise of the users. We attempted to measure expertise by looking a whether a senders recommendation appeared in their Top 50 tracks, loved tracks, or recent weekly charts, but unfortunately none of the 809 users had listened any of the recommended tracks enough to them to appear in any of those list.
Upon further examination the above Treemap is not accurate as we thought, by just looking at just the recommendation that were taken we can see that there are some people that sent recommendations to themselves an listened to them.

**Identifying Connectors**

We planned to identify the connectors by identifying users that connected disparate social groups. Unfortunately, none of the tools we used easily allowed us to do this. We looked into graph visualization tools, but the most refined were not freely available for download. We ended up hacking together a rough visualization using the Prefuse Information Visualization Toolkit (prefuse.org). The result is unfortunately rather unmanageable: although the application is interactive, it is still nearly impossible to untangle strongly connect users from their surroundings. While this toolkit holds promise for this type of analysis, the lack of a frontend to filter data severely hampers its usability.
Miscellaneous Discoveries

Although we were unable to identify the Mavens, Connectors, and Salesmen, we were able to make numerous interesting discoveries about the data we collected. Below is an interesting look at listening trends and gender.

**Users by Gender**

This chart to the left shows that Last.fm users are predominantly male, though a small percentage do decline to provide their gender.

**Songs listened by Gender**

This chart to the right shows that as expected, Males make up most of the total songs plays to on Last.fm.
When the total number of recommendations made by users is categorized by gender (top), we again see that males dominate the ratios. However, when it is examined whether or not the receiver listened to the recommended song or artist (bottom), the chart changes significantly. The fraction of recommendations that were taken from female users jumps to 29% of the total as compared to only being about 20% of the total population. In other words, this chart seems to indicate that LastFM users are more likely to listen to recommendations coming from a female user than from a male user.
The picture becomes even clearer when the ratio of recommendations taken versus total recommendations made is compared across gender. Here we see that on average users listen to recommendations from females about twice as often as males, and recommendations from unspecified genders around 3 times as often as those from males.
Evaluation of the Tools

The tools were evaluated using a subset of Schneiderman's 8 golden rules of interface design, along with three implementation specific rules relevant to our analyses. Universal usability refers to the ability of the implementation to run on different operating systems. We are both Linux users, so this was an important factor in our impressions of the applications. Presentation quality refers to the application's ability to produce images, charts, etc useful for presentations. Since we ultimately wanted to display our findings, the tools needed to do this effectively. Finally, data transformation refers to the ability to manipulate and transform data from within the application. Our source data wasn't organized particularly well, so being able to transform it on the fly was potentially useful.

Treemap

Design dialog to yield closure.
The design of the Main/Legend/Filters/Hierarchy pane is ordered strangely. The first action was often to set the Hierarchy, then set the Legend, then Filter, then do silly things like set the font size.

Error handling.
Treemap does not open the common CSV format, so data had to be preprocessed. Additionally, it offered to replace some missing data but not others, forcing us to again preprocess the data.

Universal Usability
Being a Java program, Treemap ran without a hitch on Linux or Windows.

Presentation Quality
We were disappointed to find that Treemap provides no facility to output an image or figure of the generated treemap.

Data Transformation
Treemap did not provide any facility to transform data. It could have been useful to avoid more data preprocessing.

Spotfire

Design dialog to yield closure.
Dialogs in Spotfire tended to yield closure.

Error handling.
The error messages in Spotfire were informative.

Universal Usability
Spotfire is Windows only. This led to some frustrations with a Windows installation that hadn't been booted in 6 months, and the ensuing onslaught of automatic updates, out of date virus definitions, etc. That wasn't the tool's fault, however.

Presentation Quality
Setting the colors, labels and titles of plots in Spotfire was easy. Spotfire has the capability to export Powerpoint slides only if Powerpoint is installed. Only single charts could be exported as images, though the resulting quality was excellent.

Data Transformation
The ability to transform data in Spotfire is excellent. An expression editor allows users to enter mathematical and logical expressions to transform the original data.