ABSTRACT
A lot of current research on recommender systems focuses on objectives that go beyond the accuracy of the recommendations; for instance, ensuring that the list of recommended items is diverse. In this work we explore a particular beyond-accuracy objective — serendipity. Existing approaches to measuring serendipity rely on comparing the produced recommendations against a baseline recommender system. We take the first step toward a metric that would allow objectively measuring the surprise (or unexpectedness) of recommendations, without comparing them against an alternative system. We propose two ways to measure surprise, which constitutes the core component of serendipitous recommendations.

Through offline experiments we compare three state-of-the-art recommendation algorithms in terms of their ability to generate surprising recommendations. For one of the suggested metrics, the results validate the intuition that a matrix factorization approach generates the most accurate but also the least surprising recommendations, while a user-based neighbourhood approach performs best in terms of surprise.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Information Filtering

General Terms
Experimentation, Measurement

Keywords
Recommender systems; evaluation metrics; beyond accuracy; serendipity

1. INTRODUCTION
The goal of recommender systems (RS) research has traditionally been focused on accurately predicting users’ ratings for unseen items. However, accuracy is not the only important objective of recommendation [12]. In recent years the focus of RS research has shifted to such objectives as ensuring the recommended items are novel and the set of items is diverse [17]. These objectives are of particular importance since in real life systems users are most likely to consider only a small set of recommendations. It is therefore crucial to make sure that this set of items is as interesting and engaging as possible.

In this paper we investigate serendipity — an objective denoting the ability of a system to generate recommendations that are both surprising and relevant [9]. Although serendipity is frequently mentioned in the RS literature as an important complement to recommendation accuracy, until now there have been few attempts to formally define and measure it. A common intuition in the RS literature suggests that, of the state-of-the-art recommendation techniques, matrix factorization approaches and the item-based neighbourhood technique are less likely to make serendipitous recommendations than the user-based neighbourhood technique [6]. However, to the best of our knowledge, this intuition has not been confirmed by experimental evaluation.

Existing approaches to measuring serendipity rely on comparing the produced recommendations against a baseline recommender system [13, 8, 1]. We propose metrics that do not require two systems, but allow evaluating the surprise of recommendations produced by any existing techniques.

In the rest of this paper, we first review research that addresses the serendipity objective and position our work with respect to the previous efforts. We then propose two alternative ways of measuring surprise, which constitutes the core component of serendipity. Our main and final contribution lies in comparing the performance of three state-of-the-art recommendation algorithms in terms of their ability to generate serendipitous recommendations. Our findings exemplify the trade-off between accuracy and serendipity, and provide a basis for further research on measuring and optimizing recommendation serendipity.

2. RELATED WORK
The term serendipity, meaning a “happy and unexpected discovery by accident”, was coined in the 18th century [2]. In the information retrieval (IR) literature, Toms [16] was among the first to recognize the importance of facilitating serendipity in information systems. Other IR studies have also analyzed the process of serendipitous information discovery and the paradox of designing for unexpected results [7, 11]. In the RS literature, Herlocker et al. [9] in-
formally defined serendipitous recommendations as “surprisingly interesting items the user might not have otherwise discovered”.

### 2.1 Increasing serendipity

The first attempts to increase the serendipity of retrieved results were reported in the IR literature, e.g., Campos and Figueiredo designed a software agent to support serendipitous information discovery through web crawling [4]. More recently, André et al. [2] suggested viewing serendipity as a combination of chance discovery and the usefulness of discovered information. The authors provided guidelines to design information systems with better support for both components of serendipity: supporting chance encounters and enhancing the user’s ability to recognize serendipitous content.

In the RS literature, Iaquinta et al. were among the first to optimize a recommender system for serendipity [10]. They considered items with low similarity to a user’s profile as potentially serendipitous recommendations and modified a content-based system to include such items alongside the standard recommendations. Onuma et al. [15] proposed a graphs-based algorithm for supporting surprising recommendations. The authors introduced the idea of computing a ‘bridge score’ of item nodes in the user-item bipartite graph. Nodes connecting separate interconnected areas in the graph receive high bridge scores as they bridge different subspaces in the item information space. The bridge score may be combined with an item relevance score when generating recommendations.

More recently, Oku and Hattori [14] presented a system that induced possibly serendipitous recommendations by selecting items whose content is a mixture of the content features of two items from the user’s profile. Adamopoulos and Tuzhilin [1] presented an approach to generate unexpected recommendations by maximizing a utility function which combines an item’s relevance and its distance from a user’s profile (although the authors do not use the term ‘serendipity’, but present it as a personalized novelty metric) [17].

A formulation of serendipity requires a distance measure between an item and a set of items. While the previous works used the average distance computed on item ratings or content features [17, 1], we hypothesize that averaging an item’s distance from the user’s profile items is not the best way to measure the item’s surprise. Intuitively, a user perceives a recommendation as surprising if it is sufficiently different from any item the user has seen before. This information may be lost when averaging the computed distances, especially if the user has been exposed to diverse items. Therefore, we propose using the lower bound item distance from the user’s profile items as an indicator of surprise. We believe this to be a reasonable estimate of how users perceive unexpected recommendations with respect to their past interactions with the system.

Note that we do not consider item relevance in our metric definitions. While relevance is also important to serendipity, we leave it to be measured separately by accuracy metrics. Furthermore, similarly to Adamopoulos and Tuzhilin [1], we also do not require a surprising item to be novel, although in principle, if an item is dissimilar to a user’s profile, it is likely to be unknown to the user.

### 3. SURPRISE METRICS

We propose two alternative definitions of surprise — one based on users’ rating behaviour and the other based on item content descriptors. Given the target item and the user’s profile (a set of items rated by the user), both metrics produce a score that indicates the level of surprise the target item brings to the user.

#### 3.1 Co-occurrence-based surprise

The first definition is based on the probability for the item to be seen (i.e., rated) together with the items in the user’s profile. To measure the pairwise co-occurrence of
items we employed normalized point-wise mutual information (PMI) [3], which measures the probability of observing specific outcomes of two independent random variables together. Given a pair of items $i$ and $j$, we compute their PMI value as:

$$\text{PMI}(i, j) = \log_2 \frac{p(i, j)}{p(i)p(j)} / - \log_2 p(i, j)$$  \hspace{1cm} (1)$$

where $p(i)$ and $p(j)$ represent the probabilities for the items to be rated by any user, and $p(i, j)$ is the probability for the same user to rate both items. PMI values range from $-1$ (in the limit) to 1, with $-1$ meaning the two items are never rated together, 0 signifying independence of the items, and 1 meaning complete co-occurrence of the items.

In order to measure the surprise of a recommended item $i$, we compute its PMI with each item in the user’s profile, and take the maximum value as the overall surprise value:

$$S_{\text{co-occ}}(i) = \max_{j \in P} \text{PMI}(i, j)$$  \hspace{1cm} (2)$$

where $P$ is the user’s profile (i.e., her set of rated items).

Since higher values of PMI$(i, j)$ signify higher co-occurrence of items $i$ and $j$ (and therefore low surprise of seeing the two items together), taking the maximum value represents the lower bound of the surprise perceived by the user when item $i$ is recommended.

Furthermore, to enable a comparison with the ideas in [17, 1], we conducted experiments with a variant where, instead of relying on the lower bound of surprise, we take the average PMI value across items in the user’s profile:

$$S^{avg}_{\text{co-occ}}(i) = \frac{\sum_{j \in P} \text{PMI}(i, j)}{|P|}$$  \hspace{1cm} (3)$$

For both $S_{\text{co-occ}}$ and $S^{avg}_{\text{co-occ}}$, lower values signify better surprise results.

**Limitations.**

We note that the co-occurrence-based definition of surprise metric may be sensitive to rare items, since the point-wise mutual information measure is known to be biased toward rare item pairs [3].

### 3.2 Content-based surprise

Our second surprise metric is based on distance applied to item content labels. We employed the complement of Jaccard similarity metric for comparing the items:

$$\text{dist}(i, j) = 1 - \frac{|L_i \cap L_j|}{|L_i \cup L_j|}$$  \hspace{1cm} (4)$$

where $L_i$ and $L_j$ are sets of labels describing items $i$ and $j$.

To measure the surprise of a recommended item $i$, we take the minimum distance between the target item $i$ and the user’s profile items:

$$S_{\text{cont}}(i) = \min_{j \in P} \text{dist}(i, j)$$  \hspace{1cm} (5)$$

Taking the minimum distance value as the overall surprise represents the lower bound of how surprising the item is with respect to the seen items.

Similarly to the co-occurrence-based surprise, we also evaluated in our experiment a variant that takes the average distance:

$$S^{avg}_{\text{cont}}(i) = \frac{\sum_{j \in P} \text{dist}(i, j)}{|P|}$$  \hspace{1cm} (6)$$

For both $S_{\text{cont}}$ and $S^{avg}_{\text{cont}}$, higher values signify better surprise results.

**Limitations.**

Note that the content-based metric is sensitive to the quality of item content labels. Low-quality labels may fail to capture the difference between items. Therefore, particular care should be taken when applying the metric to datasets with noisy data, e.g., user-generated labels.

### 4. EXPERIMENTS

#### 4.1 Datasets

We tested the proposed surprise metrics on two benchmark datasets for offline recommender system evaluation — the MovieLens 1M dataset and LastFM 1K dataset.

The MovieLens dataset contains ~1 million ratings, 6040 users and 3706 movies. The movies are annotated using a vocabulary of 18 genres (on average 1.65 genres per movie). To obtain richer content descriptors for the movies, we additionally scraped IMDb plot keywords for each movie, which resulted in an average of 81 labels per movie.

The LastFM dataset contains the listening events for 992 users and more than 100K artists. As the dataset is extremely sparse, we cleaned the set of artists by leaving only those for which we could obtain at least three LastFM tags and discarding artists that were listened to by fewer than 20 users. This resulted in 992 users and 7280 artists, with a total of ~500K ratings. The listening frequencies of the artists were transformed into ratings from 1 to 5 using the standard approach of converting implicit feedback into numerical ratings [17]. Since LastFM content labels for artists are user-generated (and not editorial as in the case of IMDb), to avoid noisy data we retrieved a maximum of the 10 most popular labels for every artist. This resulted in 9.2 labels per artist on average.

We used three state-of-the-art recommendation algorithms in our experiments: PureSVD — a matrix factorization approach (with 50 factors) that has been shown to perform well for top-N recommendation tasks [5] — and two standard neighbourhood methods — user-based and item-based collaborative filtering (with a neighbourhood size of 50) [6].

#### 4.2 Evaluation methodology

In recent years, rating-based accuracy metrics for offline RS evaluations have been replaced by precision-oriented metrics that more closely reflect the users’ interaction with the system — considering only a small set of top-ranked recommendations, ignoring the lower-ranked items.

In accordance with these state-of-the-art evaluation strategies, in this work we adopt the ‘one plus random’ methodology [5]. The methodology is based on splitting the dataset into a training set $M$ and probe set $P$, constructing the test set $T$ by selecting one highly-rated item per user from the probe set. Then, for each user $u$ we make predictions for 1000 unrated items plus the one test item. The set of 1001 items is ranked according to the recommender’s predicted score and the top-N recommendations are selected (we used $N=10$ in our experiments). If the test item is among the top-N items, we have a hit. The overall performance of the system — recall — is calculated as the ratio of the number of hits over the total number of test users. The underlying assumption that the 1000 unseen items are irrelevant is clearly
undervaluing the performance, as certain items among the 1000 may be actually relevant for the user.

We believe this methodology to be appropriate when measuring the surprise of recommendations as it involves items the user has not discovered (i.e., unrated items), whereas other offline evaluation strategies employ items the user has had no trouble discovering (i.e., already rated items).

As described in Section 3, given an item and the user’s profile, the co-occurrence-based and content-based surprise metrics produce a score in $[-1; 1]$ and $[0; 1]$ respectively. To obtain a single surprise value for a user’s top-N recommendation list, we average the surprise values of all recommended items.

We computed the results using 5-fold cross validation with 80%-20% training/probe set split.

5. RESULTS AND DISCUSSION

As a first step in the evaluation, to make sure that the two surprise metrics are not redundant, we compared them using the following procedure: for each user in the dataset, we generated a list of top-10 recommendations using the PureSVD matrix factorization approach. We then ranked the 10 items using each metric and computed Spearman’s rank correlation for the two rankings. Averaging the results over all users we obtained values close to zero for both MovieLens and LastFM datasets. This indicates that the two metric definitions capture different aspects of surprise.

In accordance with this finding, the two surprise metrics give different results in our main experiment.

5.1 Surprise value comparison

Table 1 lists the average recall and surprise values across test users for the evaluated algorithms — matrix factorization approach (MF), user-based approach (UB), and item-based approach (IB) — on both datasets. All reported metrics produce significantly different means when applied to different algorithms ($p < 0.001$ in a one-way ANOVA test).

As expected, MF produces the most accurate recommendations, outperforming both neighbourhood approaches.

For the content-based surprise metric (where higher values are better), the results are consistent for both minimum distance (Eq. 2) and average distance (Eq. 3) variants. In case of the minimum distance $S_{\text{co-occ}}$ variant, the MF approach performs as well as the UB and IB approaches, even outperforming both of them on the LastFM dataset. Contrasting, the average distance variant $S_{\text{avg}}$ shows MF to be outperformed by both UB and IB techniques.

An explanation for results of the co-occurrence-based surprise metric may lie in its sensitivity to rare items. In our experiments, the matrix factorization approach tends to rank popular items higher than both neighbourhood approaches, which can influence the values of point-wise mutual information. When one of the items in PMI$(i, j)$ is rare, the normalized PMI metric can take extreme values — close to 1 if the two items frequently co-occur in the dataset, or close to −1 if they almost never occur together. Consequently, the average distance metric variant (which takes the maximum PMI value as the lower bound of an item’s distance to a user’s profile) tends to produce higher values when applied to rare items, while the average distance metric variant tends to produce low overall scores as it gets reduced by the negative PMI values. More experiments are needed to confirm this intuition. If confirmed, the $S_{\text{co-occ}}$ metric could be modified to avoid a bias toward rare items [3].

5.2 The impact of user’s profile size

The size of a user’s profile is an important factor to consider when measuring surprise of recommendations. Intuitively, the more items a user has been exposed to, the more difficult it is to surprise the user. To test this intuition, we plotted the surprise values obtained for individual users against their profile size. For both surprise metrics, we plotted two variants of results: one obtained using the minimum distance from the user’s profile, and the other using the average distance from the profile. In both cases we measured the surprise of the recommendations generated using the most accurate recommendation method — the PureSVD approach.

Results for the minimum distance metric variant for both datasets are shown in the left half of Figure 1. As the user profile size increases, surprise values for the $S_{\text{co-occ}}$ metric tend to increase and the values for the $S_{\text{cont}}$ metric decrease, i.e., in both cases the surprise of recommendations is decreasing. This finding confirms the intuition that a large user’s profile, there is a higher chance of finding an item similar to the recommended item, therefore, the chance of surprising the user is smaller.

The same trend is not observed when computing surprise using the average item’s distance from a user’s profile (right half of Figure 1), as the surprise values are independent of the profile size. We therefore believe that the minimum distance is more appropriate when measuring an item’s surprise with respect to a user’s profile, since averaging item distances results in losing information about the user’s past interactions with the system.

5.3 Qualitative analysis

For a qualitative insight into the results, we looked at the recommendations generated for individual users. Table 2 shows for one user from the LastFM dataset 10 items from her profile and the top-10 recommendations generated using each algorithm. A subset of content labels is displayed for

<table>
<thead>
<tr>
<th>Metric</th>
<th>MovieLens 1M</th>
<th>LastFM 1K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>0.334 ± 0.022</td>
<td>0.065 ± 0.051</td>
</tr>
<tr>
<td>$S_{\text{co-occ}}$</td>
<td>0.349 ± 0.354</td>
<td>0.407 ± 0.407</td>
</tr>
<tr>
<td>$S_{\text{cont}}$</td>
<td>0.915 ± 0.928</td>
<td>0.613 ± 0.613</td>
</tr>
<tr>
<td>$S_{\text{avg}}$</td>
<td>0.177 ± 0.146</td>
<td>0.097 ± 0.097</td>
</tr>
<tr>
<td>$S_{\text{cont}}$</td>
<td>0.976 ± 0.981</td>
<td>0.928 ± 0.928</td>
</tr>
</tbody>
</table>

As expected, MF produces the most accurate recommendations, outperforming both neighbourhood approaches.

For the content-based surprise metric (where higher values are better), the results are consistent for both minimum distance (Eq. 5) and average distance (Eq. 6) metric variants — MF performs significantly worse than both UB and IB, while UB performs better than IB.
each item. (The full set of labels is omitted due to space limitations.) As can be seen from the content labels, all three algorithms generate items that are related to the user’s musical preferences — independent rock and electronic music. However, user-based recommendations include more artists that may be surprising discoveries for the user, e.g., a jazz musician and a string quartet playing covers of popular songs in classical style. The item-based approach generates one such recommendation — a heavy metal band playing rockabilly songs.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we investigated two metrics – co-occurrence-based \( S_{\text{co-occ}} \) and content-based \( S_{\text{cont}} \) – for measuring surprise in recommender systems. We evaluated each metric with three state-of-the-art recommenders on two benchmark datasets.

For both co-occurrence- and content-based metrics, two variants were evaluated — one using the minimum item’s distance from the user’s profile items as an indicator of surprise, and the other using the average distance from the profile items.

Using the \( S_{\text{cont}} \) surprise metric, the experimental results confirm that there is a trade-off between recommendation accuracy and serendipity [8] and the intuition that the matrix factorization approach, while being the most accurate, produces the least surprising recommendations with respect to the user’s profile. Conversely, the user-to-user collaborative filtering approach produces the most surprising recommendations.

The above finding is confirmed for both \( S_{\text{cont}} \) metric variants — the minimum distance and average distance. However, the analysis of user’s profile size influence on surprise values indicates that some information may be lost when using the average distance metric variant. Therefore, we believe that using the minimum distance to profile items is the better approach for measuring surprise.

The findings of \( S_{\text{cont}} \) are not fully confirmed using the alternative, co-occurrence-based surprise metric \( S_{\text{co-occ}} \). We believe the reason for this lies in the metric’s sensitivity to rare items. However, the metric deserves further investigation, since in contrast to the content-based metric, it does not rely on item content labels and may capture a different aspect of surprise than the content-based metric.

The next steps in our research include measuring how the state-of-the-art recommendation algorithms perform with respect to other beyond-accuracy objectives, such as diversity, novelty, and coverage, in addition to serendipity. Additionally, we want to investigate the correlations and trade-offs between these objectives. Finally, we would like to conduct a live user study, since no evaluation of beyond-accuracy objectives can be complete without obtaining feedback from real users.

7. ACKNOWLEDGMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

8. REFERENCES

Table 2: A sample user profile and the generated recommendations from the LastFM dataset. The light-shaded item is the recovered hidden test item. The dark-shaded items are musicians belonging to music genres not present in the user’s profile — jazz, classical, and heavy metal music.

<table>
<thead>
<tr>
<th>Profile</th>
<th>PureSVD (MF)</th>
<th>$\text{R}_{\text{content}}$</th>
<th>$\text{R}_{\text{coll}}$</th>
<th>User-based (UB)</th>
<th>$\text{R}_{\text{content}}$</th>
<th>$\text{R}_{\text{coll}}$</th>
<th>Item-based (IB)</th>
<th>$\text{R}_{\text{content}}$</th>
<th>$\text{R}_{\text{coll}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Death Cab for Cutie</td>
<td>Bloc Party</td>
<td>0.318</td>
<td>0.333</td>
<td>Liam Finn</td>
<td>0.357</td>
<td>0.750</td>
<td>Gram Parsons</td>
<td>0.497</td>
<td>0.574</td>
</tr>
<tr>
<td>indie rock, alternative, emo, pop</td>
<td>indie rock, alternative, britpop</td>
<td>indie, new wave, alternative, britpop</td>
<td>indie, singer-songwriter, folk, pop</td>
<td>pop, punk, rock, alternative, emo</td>
<td>pop, punk, rock, alternative, emo</td>
<td>pop, punk, rock, alternative, emo</td>
<td>pop, punk, rock, alternative, emo</td>
<td>pop, punk, rock, alternative, emo</td>
<td>pop, punk, rock, alternative, emo</td>
</tr>
<tr>
<td>John Mayer</td>
<td>The Smiths</td>
<td>0.381</td>
<td>0.333</td>
<td>Quetedrive</td>
<td>0.395</td>
<td>0.574</td>
<td>Ralph Myers</td>
<td>0.390</td>
<td>0.667</td>
</tr>
<tr>
<td>singer-songwriter, acoustic, rock, blues</td>
<td>indie rock, alternative, britpop</td>
<td>indie rock, alternative, britpop</td>
<td>indie rock, alternative, britpop</td>
<td>indie rock, alternative, britpop</td>
<td>indie rock, alternative, britpop</td>
<td>indie rock, alternative, britpop</td>
<td>indie rock, alternative, britpop</td>
<td>indie rock, alternative, britpop</td>
<td>indie rock, alternative, britpop</td>
</tr>
<tr>
<td>Damien Rice</td>
<td>MB3</td>
<td>0.371</td>
<td>0.667</td>
<td>Steve Aoki</td>
<td>0.379</td>
<td>0.574</td>
<td>The Enemy</td>
<td>0.466</td>
<td>0.333</td>
</tr>
<tr>
<td>The New Pornographers</td>
<td>The Smiths</td>
<td>0.379</td>
<td>0.46</td>
<td>Jan Garbarek</td>
<td>0.446</td>
<td>0.889</td>
<td>To My Boy</td>
<td>0.365</td>
<td>0.750</td>
</tr>
<tr>
<td>indie rock, alternative, power pop</td>
<td>indie, alternative, rock, pop, american</td>
<td>indie rock, alternative, power pop</td>
<td>indie rock, alternative, power pop</td>
<td>indie rock, alternative, power pop</td>
<td>indie rock, alternative, power pop</td>
<td>indie rock, alternative, power pop</td>
<td>indie rock, alternative, power pop</td>
<td>indie rock, alternative, power pop</td>
<td>indie rock, alternative, power pop</td>
</tr>
<tr>
<td>The Postal Service</td>
<td>Pink Floyd</td>
<td>0.264</td>
<td>0.571</td>
<td>Bloc Party</td>
<td>0.318</td>
<td>0.333</td>
<td>Seasick Steve</td>
<td>0.458</td>
<td>0.571</td>
</tr>
<tr>
<td>indie, electronic, pop, alternative, emo</td>
<td>rock, progressive, psychedelia, classic</td>
<td>rock, progressive, psychedelia, classic</td>
<td>rock, progressive, psychedelia, classic</td>
<td>rock, progressive, psychedelia, classic</td>
<td>rock, progressive, psychedelia, classic</td>
<td>rock, progressive, psychedelia, classic</td>
<td>rock, progressive, psychedelia, classic</td>
<td>rock, progressive, psychedelia, classic</td>
<td>rock, progressive, psychedelia, classic</td>
</tr>
<tr>
<td>Toby Keith</td>
<td>Rataatat</td>
<td>0.337</td>
<td>0.462</td>
<td>Ralph Myers</td>
<td>0.390</td>
<td>0.667</td>
<td>Elf Power</td>
<td>0.422</td>
<td>0.333</td>
</tr>
<tr>
<td>The Hold Steady</td>
<td>Green Day</td>
<td>0.246</td>
<td>0.571</td>
<td>Olivia Ruiz</td>
<td>0.394</td>
<td>0.750</td>
<td>Bloc Party</td>
<td>0.318</td>
<td>0.333</td>
</tr>
<tr>
<td>indie rock, alternative, post-punk</td>
<td>punk rock, alternative, pop punk</td>
<td>punk rock, alternative, pop punk</td>
<td>punk rock, alternative, pop punk</td>
<td>punk rock, alternative, pop punk</td>
<td>punk rock, alternative, pop punk</td>
<td>punk rock, alternative, pop punk</td>
<td>punk rock, alternative, pop punk</td>
<td>punk rock, alternative, pop punk</td>
<td>punk rock, alternative, pop punk</td>
</tr>
<tr>
<td>Clap Your Hands Say Yeah</td>
<td>GIRL TALK</td>
<td>0.347</td>
<td>0.667</td>
<td>Love And Rockets</td>
<td>0.355</td>
<td>0.182</td>
<td>buffalo</td>
<td>0.341</td>
<td>0.402</td>
</tr>
<tr>
<td>The View</td>
<td>MSTRKRFT</td>
<td>0.329</td>
<td>0.571</td>
<td>The String Quartet</td>
<td>0.258</td>
<td>0.889</td>
<td>Boys Night Out</td>
<td>0.345</td>
<td>0.571</td>
</tr>
<tr>
<td>indie rock, britpop, garage rock, punk</td>
<td>electronic, dance, house, indie</td>
<td>electronic, dance, house, indie</td>
<td>electronic, dance, house, indie</td>
<td>electronic, dance, house, indie</td>
<td>electronic, dance, house, indie</td>
<td>electronic, dance, house, indie</td>
<td>electronic, dance, house, indie</td>
<td>electronic, dance, house, indie</td>
<td>electronic, dance, house, indie</td>
</tr>
<tr>
<td><strong>Average values</strong></td>
<td><strong>0.330</strong></td>
<td><strong>0.459</strong></td>
<td><strong>0.364</strong></td>
<td><strong>0.527</strong></td>
<td><strong>0.352</strong></td>
<td><strong>0.534</strong></td>
<td><strong>0.352</strong></td>
<td><strong>0.534</strong></td>
<td><strong>0.352</strong></td>
</tr>
</tbody>
</table>


