Social Knowledge-Driven Music Hit Prediction

Kerstin Bischoff, Claudiu S. Firan, Mihai Georgescu,
Wolfgang Nejdl, and Raluca Paiu

L3S Research Center / Leibniz Universität Hannover
Appelstrasse 9a
30167 Hannover, Germany
{bischoff,firan,georgescu,nejdl,paiu}@L3S.de

Abstract. What makes a song to a chart hit? Many people are trying to find the answer to this question. Previous attempts to identify hit songs have mostly focused on the intrinsic characteristics of the songs, such as lyrics and audio features. As social networks become more and more popular and some specialize on certain topics, information about users' music tastes becomes available and easy to exploit. In the present paper we introduce a new method for predicting the potential of music tracks for becoming hits, which instead of relying on intrinsic characteristics of the tracks directly uses data mined from a music social network and the relationships between tracks, artists and albums. We evaluate the performance of our algorithms through a set of experiments and the results indicate good accuracy in correctly identifying music hits, as well as significant improvement over existing approaches.

Keywords: collaborative tagging, classification, hit songs, social media.

1 Introduction

The benefits of being able to predict which songs are likely to become hits is various and is of big interest for both music industry and artists, as well as for listeners. In their attempt to release only profitable music, producers may want to have an indication of the potential of the music songs they will work with. Artists can profit from the results of such techniques by identifying the most suitable markets for their songs, music lovers' niches and by choosing the best channels and targets. Last but not least, normal music listeners can enjoy good music as a benefit of accurate hit predictions on a daily basis – radio stations can use such methods in order to improve their program by playing only songs which are highly likely hits.

Most previous attempts to identify hit songs have focused on intrinsic characteristics of songs, such as lyrics and audio features. In the prevailing view it is all about musical quality, so the task is to reveal the audience's preferences about music – *e.g.* by finding the similarity to what they liked before. However, it is often neglected that people are not independently deciding on what they like, but rather they like what they think other people may also like [1]. Despite 'intrinsic quality' success seems also to depend on the already known or

R. Huang et al. (Eds.): ADMA 2009, LNAI 5678, pp. 43–54, 2009.

assumed popularity, *i.e.* we find a rich get richer effect (a.k.a. preferential attachment or cumulative advantage). Thus, subjective opinions of a few early-arriving individuals account for hit potential as well.

As social networks become more and more popular and some specialize on certain topics, information about users' music tastes becomes available and easy to exploit. The wisdom of the crowds has become a famous notion of the collective intelligence manifesting itself in such collaborative tagging systems. Most important in our case, these networks set and identify trends and hot topics. In the present paper we propose a method for predicting the success of music tracks by exploiting social interactions and annotations, without relying on any intrinsic characteristics of the tracks. We predict the potential of music tracks for becoming hits by directly using data mined from a music social network (*Last.fm*) and the relationship between tracks, artists and albums. The social annotations and interactions enable both measuring similarity (*i.e.* intrinsic quality) of songs and finding those critical early-stage effects of cumulative advantage. Our approach requires only the social data corresponding to a track's first week life in *Last.fm* (*i.e.* the track is released *only* to the *Last.fm* audience), in order to be able to make good predictions about its potential and future evolution[1].

2 Related Work

Below we discuss the most relevant existing work, structured according to the two main directions along which we develop our methodology for predicting music hits.

2.1 Music Hits Prediction

Some previous work focused on automatic prediction of hit songs: in [2], the authors explore the automatic separation of hits from non-hits by extracting both acoustic and lyrics information from songs using standard classifiers on these features. Experiments showed that the lyrics-based features were slightly more useful than the acoustic features in correctly identifying hit songs. As ground truth data the authors made use of the Oz Net Music Chart Trivia Page[2]. This set is somewhat limited as it only contains top-1 hits in US, UK and Australia and the corpus used in the experiments was quite small – 1700 songs. In our approach we use a larger corpus and a much richer ground truth data set – the *Billboard.com* charts. Besides, our algorithms do not rely on lyrics or acoustic information but exploit social network data.

[3] focus on a complementary dimension: given the first weeks' sales data, the authors try to predict how long albums will stay in the charts. They also analyze whether a new album's position in the charts can be predicted for a certain week in the future. One of the most prominent commercial products

[1] *Last.fm* offers to artists the possibility to upload their own music to the portal (`http://www.last.fm/uploadmusic?accountType=artist`).

[2] `http://www.onmc.iinet.net.au/trivia/hitlist.htm`

for music hit prediction HSS[3] employs Spectral Deconvolution for analyzing the underlying patterns in music songs, *i.e.* it isolates patterns such as harmony, tempo, pitch, beat, and rhythm. Users of this service can upload a song, the system then analyzes it and compares it against existing chart hits from its database. The drawback of this system is that by using low-level features only, it cannot correctly predict the success of completely new types of music.

[4] claim that the popularity of a track cannot be learned by exploiting state-of-the-art machine learning (see also [1]). The authors conducted experiments contrasting the learnability of various human annotations from different types of feature sets (low-level audio features and 16 human annotated attributes like genre, instruments, mood or popularity). The results show that while some subjective attributes can be learned reasonably well, popularity is not predictable beyond-random – indicating that classification features commonly used may not be informative enough for this task. We investigate whether user generated interaction and (meta)data can serve as the missing link.

2.2 Social Media and Collaborative Tagging

Social media data provides ground for a wide range of applications. In [5], the authors make use of social media data for identifying high-quality content inside the Yahoo! Answers portal. For the community question-answering domain, they introduce a general classification framework for combining the evidence from different sources of information. Their proposed algorithms prove the ability to separate high-quality items from the rest with an accuracy close to that of humans. Our algorithms have a similar goal, though applied to a different domain – music. Though, it does not use tags, to a certain extent, the work of [6] is similar to ours: the authors analyze the potential of blog posts to influence future sales ranks of books on *Amazon.com*. The authors showed that simple predictors based on blog mentions around a product can be effective in predicting spikes in sales ranks.

Especially user generated tags have been extensively exploited: for building user profiles, for improving (personalized) information retrieval, results' clustering, classification or ontology building. Focusing on trend detection, in [7] the authors propose a measure for discovering topic-specific trends within folksonomies such as *Delicious*. Based on a differential adaptation of Google's PageRank algorithm, changes in popularity of tags, users, or resources within a given interval are determined. Similarly, [8] measure the interestingness of *Flickr* tags by applying a TFxIDF like score for different intervals in time.

For music, [9] found that *Last.fm* tags define a low-dimensional semantic space which - especially at the track level highly organized by artist and genre - is able to effectively capture sensible attributes as well as music similarity. We use this valuable folksonomy information for predicting music hits. To our best knowledge, user tags have not been used so far to infer hit potential of songs.

[3] `http://www.hitsongscience.com`

3 Data Sets

3.1 *Last.fm*

The method we propose for predicting music hits relies on external social information extracted from the popular music portal, *Last.fm*, a UK-based Internet radio and music community website, founded in 2002 and now owned by CBS Interactive. Statistics of the site claim 21 million users in more than 200 countries are streaming their personalized radio stations provided by *Last.fm*.

One of the most popular features of *Last.fm* user profiling is the weekly generation and archiving of detailed personal music charts and statistics. Users have several different charts available, including Top Artist, Top Tracks and Top Albums. Each of these charts is based on the actual number of times people listened to the track, album or artist. Similar global charts are also available and these are created based on the total number of individual listeners. Another important feature of *Last.fm* and crucial for our algorithms is the support for user-end tagging or labeling of artists, album and tracks. Thus, *Last.fm* creates a site-wide folksonomy of music. Users can browse musical content via tags or even listen to tag radios. Tags can fall into many categories, starting from genre, mood, artist characteristics and ending with users' personal impressions (for a detailed description of the existing types of *Last.fm* tags see [10]).

We collected 317,058 tracks and their associated attributes, such as artist and song name, number of times the tracks have been listened to on *Last.fm*, the name of the albums featuring the tracks, as well as tags that have been assigned to the songs. Additionally, the crawl contained information about 12,193 *Last.fm* users, all of them having listened to at least 50 songs and having used at least 10 tags. We started from the initial set of 12,193 crawled users and for all of them we downloaded all their available weekly charts. For this task we made use of the Audioscrobbler[4] web services. As not all of the 12,193 users from our initial set have been active since May 2007, we could gather charts for only 10,128 of them. A weekly chart consists of a list of songs that the user has listened during that particular week. The weekly charts we could gather span over 164 weeks and our final data collection consisted of 210,350 tracks, performed by 37,585 unique artists. 193,523 unique tags are associated with the tracks, 163,483 of these tags occurring as well along with artists.

3.2 Billboard Charts

For being able to asses the quality of our predictions, we also needed a good ground-truth data set. The most suitable for our purposes was the data exposed by *Billboard.com*. Billboard is a weekly American magazine devoted to the music industry, which maintains several internationally recognized music charts that track the most popular songs and albums in various categories on a weekly basis[5]. The charts, based on sales numbers and radio airplays are released as

[4] `http://www.audioscrobbler.net`
[5] `http://www.billboard.biz/bbbiz/index.jsp`

.html pages and represent the top tracks of the previous week. Every chart has associated a name, an issue date, and stores information about the success of the songs in form of rank, artist name and album/track name. Moreover, each chart entry has a previous week rank, as well as a highest rank field – *i.e.* the highest Billboard position ever reached by that song.

There are 70 different charts available for singles and 57 different ones for albums and a detailed list can be found at `http://www.billboard.biz/bbbiz/charts/currentalbum.jsp` and `http://www.billboard.biz/bbbiz/charts/currentsingles.jsp`, for albums and singles, respectively. We collect all these Billboard charts and aggregate the information, the resulting charts thus spanning over a range of almost 50 years, namely between August 1958 and April 2008. In total, the aggregated Billboard single chart contained 1,563,615 entries, 68,382 of them being unique songs. With respect to albums, the aggregated chart had 1,200,156 entries and among those, only 49,961 proved to be different albums.

The final set of tracks and the corresponding information around these tracks (artist, album, Billboard rank, *etc.*) is represented by the intersection of the set of unique tracks gathered from the *Last.fm* users' weekly charts and the set of tracks included in the Billboard charts. This intersection resulted in 50,555 unique music songs, on which we will perform our experiments.

4 Predicting Music Hits

We make use of the social information around the tracks, which we gather from the popular music portal *Last.fm*. This information is processed and transformed into a list of features, which is fed to a classifier for training it to discover potentially successful songs. The approach we propose relies on the following assumptions:

- The initial popularity (*i.e.* the popularity among listeners after only one week after the upload) of a track is indicative of its future success.
- Previous albums of the same artist have a direct influence on the future success of the songs.
- The popularity of other tracks produced by the same artist and included on the albums we consider has also an impact on the future success of a song.
- Popularity of the artist performing a track, in general, has a direct influence on the future success of new songs.

With these hypotheses fitting perfectly to the principles of preferential attachment/cumulative advantage, we now proceed describing the details of our music hit prediction algorithm based on social media data.

4.1 Feature Selection

The features used for training the classifiers are chosen such that the assumptions listed above are supported. It is thus natural to build a model where the main

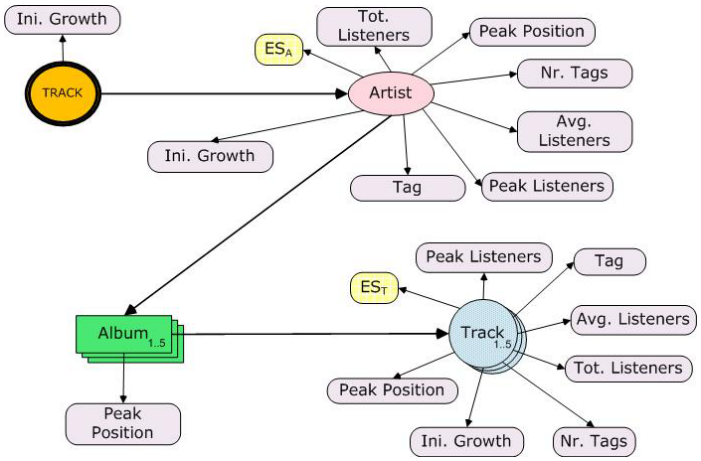

Fig. 1. Features used for training the classifiers

entities correspond to the interpreting *Artist*, previous popular *Albums* of the same artist and *Tracks* included on the albums considered. Moreover, each of these entities has associated a set of attributes, which are as well taken as input features for our classifiers. In Figure 1 we present the complete set of features considered.

All entities and their associated attributes related to a particular track, for which we would like to predict whether it will be a hit or not, form a tree having the $TRACK$ as root. Each of the features can be reached by starting from the root of the feature-tree and following the corresponding branches. We now discuss in detail the main feature entities and their associated attributes composing the feature tree.

Artist-Relevant Features. Artists as the performers of the songs we make predictions for are likely to have an influence on their hit potential. Usually, artist entities have associated a set of tags assigned by *Last.fm* users – we consider the top 5 most used tags, $Tag_{1..5}$. In case the artist does not yet have 5 tags, we exploit as many as available. Besides, we also include the total number of tags available for an artist, $Nr.\ Tags$, as well as its overall number of listeners, $Tot.\ Listeners$. $Ini.\ Growth$ represents the number of listeners for this artist during the first week it appeared in *Last.fm* charts (Note that these are *Last.fm* user charts.). The higher this number, the bigger the probability that this artist is quite popular and his future songs will become hits with a high probability. The $Peak\ Listeners$ feature measures the maximum number of listeners in one week over all weeks and all *Last.fm* user charts. With $Avg.\ Listeners$ we capture the average number of listeners over all *Last.fm* user charts and the value is computed as:

$$Avg.\ Listeners = \frac{Tot.\ Listeners}{\#weeks\ in\ Last.fm\ user\ charts} \tag{1}$$

The *Peak Position* represents the highest Billboard position this artist reached so far (for new artists, this value will not be known).

An *Artist* is directly connected to an *Album*-entity, since the performer might have produced several albums already. Thus, we also include as features the artist's top-5 albums, or as many as currently available.

Track-Relevant Features. In the model presented in Figure 1, containing the features used for training the classifiers, the *Track* entity occurs twice. It is important to distinguish between the two different instances: $TRACK$ represents the song for which we aim to automatically predict whether it will belong to the class "HIT" or "NHIT", which is also the root of the tree resulted from the complete set of features. Beside this, we also consider top-5 tracks, $Track_{1..5}$, appearing on the albums we include as feature for the given $TRACK$.

For $TRACK$, the track for which we want to make the predictions, only the $Ini.\ Growth$ feature is considered (the maximum number of *Last.fm* listeners after the first week this song appeared on the *Last.fm* portal). The artist of the track, *Artist* represents an entity directly connected to $TRACK$ and for the case that the song has more authors (*e.g.* Madonna featuring Justin Timberlake) we consider only the first artist.

For $Track_{1..5}$, the tracks associated to other albums of the same artist, we include as feature the overall number of listeners on *Last.fm*, $Tot.\ Listeners$, as a strong indicator of its popularity. The tags given by the *Last.fm* users to a track are as well good popularity indicators. We consider the top-5 tags, $Tag_{1..5}$, or like in the case of the artists, if there are less than 5 tags, we consider as many as available. $Peak\ Position$, $Avg.\ Listeners$, $Peak\ Listeners$ and $Ini.\ Growth$ have the same meaning as the corresponding artist-related features.

Album-Relevant Features. Similar to the case of artist-like entities, albums also have associated a series of features: their popularity can be measured based on the highest position reached in Billboard, the *Peak Position* feature. Since for some artists the previously released albums can be quite many, we include only the top-5 albums which reached positions in the Billboard charts. Besides, from each album we also consider the top-5 Billboard listed tracks, $Track_{i_{1..5}}$.

Additional Features. In addition to the direct features discussed above, we also extract some implicit features for the artist and track entities. We associate ES-Entity Scores features, as a combination of the entities' Billboard top reached position and their HITS [11] scores – computed by applying HITS on a graph using artists, tracks and tags as nodes. Given an artist A, a track T and a tag TG, we create links as follows:

- From A to T, if track T is played by artist A;
- From T to TG, if track T has been tagged with tag TG;
- From A to TG, if artist A has been tagged with tag TG.

On the resulted graph we apply the HITS algorithm and compute the corresponding hub and authority scores. We present below the formulas for computing the HITS scores for artists, HS_A and tracks entities, HS_T:

$$HS_{A|T} = \begin{cases} 0, \text{ if } hubS_{A|T} == 0 \wedge authS_{A|T} == 0; \\ authS_{A|T}, \text{ if } hubS_{A|T} == 0 \wedge authS_{A|T}! = 0; \\ hubS_{A|T}, \text{ if } hubS_{A|T}! = 0 \wedge authS_{A|T} == 0; \\ authS_{A|T} \cdot hubS_{A|T}, \text{ otherwise.} \end{cases} \quad (2)$$

$hubS_{A|T}$ and $authS_{A|T}$ represent the hub- and authority scores of the artist and track (represented as subscripts A or T respectively).

The final Entity Scores (ES) will be based both on the outcome of calculating the HITS scores and the corresponding best positioning in any Billboard charts ever. This score will give an estimation of the popularity of certain artists and tracks in relation to the tags used, between themselves and in the opinion of a recognized authority in the domain, as the Billboard charts are. The formula for computing ES_A and ES_T, the entity scores for artists and tracks is given below:

$$ES_{A|T} = \begin{cases} \frac{1}{1000} \cdot HS_{A|T}, \text{ if } PeakPos_{A|T} \text{ is missing;} \\ \\ \frac{1}{PeakPos_{A|T}} \cdot HS_{A|T}, \text{ otherwise.} \end{cases} \quad (3)$$

$PeakPos_{A|T}$ represents the best reached position by the artist or track in all considered Billboard charts. If these entities do not occur in any of the charts (they never got that successful as to be included in the music tops), we consider a large number (1000) to substitute their missing Billboard rank. The inverse of this number or of the best Billboard position is considered for the computation of the final Entity Score (see Equation 3). The resulting $ES_{A|T}$ scores for artists and tracks will be used as features for our music hits prediction algorithm. They will be attached to the corresponding entities depicted in the feature graph from Figure 1.

4.2 Music Hit Prediction Algorithm

The core of our music hit prediction method is a classifier trained on the *Billboard.com* ground truth and using as features social media data extracted from *Last.fm* or inferred from it. We experiment with a number of different classifiers (Support Vector Machines, Naïve Bayes, Bayesian Networks and Decision Trees) and for building the classifiers we use the corresponding implementations available in the open source machine learning library Weka[6] [12]. Given the four hypotheses mentioned above, the classifiers learn a model from a training set of data. Once the model is learned, it can be applied to any unseen data from *Last.fm* and predict whether the corresponding songs have the potential of becoming hits or not.

The set of songs described in Section 3 is split into two partitions: one partition for training and one for testing the classifiers. We train classifiers for several rank ranges, such that the partitioning of the data satisfies the following: For hit class $1-1$, we consider as hit songs only those tracks which have reached top-1 in Billboard charts. All other songs starting with the second position in Billboard are considered non-hits. Similarly, other hit rank ranges are considered: $1-3$

[6] `http://www.cs.waikato.ac.nz/~ml/weka`

(*i.e.* tracks which have reached top-3 Billboard positions are regarded hits, while the rest, starting from position 4, are non-hits), $1-5$, $1-10$, $1-20$, $1-30$, $1-40$ and $1-50$. The number of hit and non-hit instances is approximately the same for all classifiers. We select as many songs as available from the rank ranges considered as hits. For non-hits, we randomly pick about the same number of songs from the set of music tracks with Billboard positions greater than the right margin of the hit class or from the set of tracks not appearing at all in the Billboard charts (*i.e.* "clear" non-hits). We summarize in Table 1 the resulting number of instances for each of the hits' rank ranges.

Each classifier is trained and tested on the total set of instances (both hits and non-hits), corresponding to each of the hit class ranges. For the songs in the training set, we build the set of corresponding features according to the attributes attached to the main entities (artist, albums, tracks) as depicted in Figure 1. The classifier is trained on the resulting set of features and a model is learned from it. After this step, the model is applied to all songs from the test data and a prediction is made.

5 Experiments and Results

For measuring the performance of our prediction algorithm we use the following metrics:

- Accuracy (Acc) – Statistical measure of how well the classifier performs overall;
- Precision (P) – Probability for items labeled as class C of indeed belonging to C;
- Recall (R) – Probability of all items belonging to class C of being labeled C;
- F1-measure (F1) – Weighted harmonic mean of Precision and Recall;
- Area under ROC (AUC) – Probability that a randomly chosen example not belonging to C will have a smaller estimated probability of belonging to C than a randomly chosen example indeed belonging to C.

We experimented with several multi-class classifiers: Support Vector Machines, Naïve Bayes, Decision Trees and Bayesian Networks with 1 or 2 parents, but given the space limitations only the best results are presented – this was the case of Bayesian Networks with 2 parents. In Table 1 we also present the averaged results of the 10-fold cross validation tests.

As observed from Table 1, the best results are obtained for the classifier built for detecting top-1 music hits. For this case, we obtain a value of 0.883 for the AUC measure, 0.788 precision and 0.858 recall for hits, while the overall accuracy is 81.31%. In [2] the authors reported AUC values of 0.69 for the best performing classifiers, trained to recognize top-1 hits from charts in Unites States, UK and Australia. Having similar data sets' sizes and song sets with no bias on any particular music genre (though the tracks might be different), our results for class $1-1$ are comparable with the ones reported by [2]. Our approach performs better, providing $\approx 28\%$ improvement in terms of AUC values over the

Table 1. Classifiers' evaluation for predicting Hits/Non-Hits, considering different rank intervals for the hit-classes

Hits' Range	#Hits	#Non-Hits	Acc[%]	Hits				Non-Hits			
				P	R	F1	AUC	P	R	F1	AUC
1 – 1	2,335	2,331	81.31	0.788	0.858	0.821	0.883	0.844	0.768	0.804	0.883
1 – 3	3,607	3,594	79.73	0.768	0.852	0.808	0.875	0.833	0.742	0.785	0.875
1 – 5	4,354	4,339	79.57	0.765	0.854	0.807	0.871	0.834	0.737	0.783	0.87
1 – 10	5,553	5,515	79.24	0.771	0.835	0.801	0.857	0.818	0.75	0.783	0.856
1 – 20	7,016	6,913	75.84	0.804	0.688	0.741	0.848	0.724	0.83	0.773	0.848
1 – 30	8,035	7,897	75.87	0.808	0.684	0.741	0.85	0.722	0.835	0.774	0.85
1 – 40	8,744	8,538	75.28	0.802	0.679	0.735	0.843	0.716	0.829	0.768	0.843
1 – 50	9,024	8,807	75.19	0.803	0.676	0.734	0.84	0.714	0.83	0.768	0.84

methods described [2]. It has been argued that AUC values between 0.5 – 0.7 are an indicator of low accuracy, while AUC values between 0.7 – 0.9 indicate good accuracy [13].

For all other classifiers, the results present as well characteristics which indicate good classification accuracy. In terms of AUC values, the performance is a bit worse than for the very restrictive case of hits taken only from top-1 Billboard charts (class $1-1$). The main reason for this is the fact that as we increase the rank range for what we call hits, the tracks begin to have a more heterogeneous set of features making it more difficult for classifiers to distinguish the correct hits from the rest of the songs. However, as we increase the interval ranges, precision improves in the detriment of recall, the best value being achieved for hit predictions from the interval $1-30$.

For the scenarios we consider, precision is actually more important than recall: a music label would be interested in promoting as far as possible only those music tracks which definitely have the potential of becoming hits; most radio stations try to play only music tracks which are already popular and on their way to top positions in the music charts. The main advantage of relying on such an approach is the fact that they can easily identify new and fresh sounds after just one week of letting the song "in the hands" of the *Last.fm* users.

In addition to the experiments described above, we also tested the accuracy of the built classifiers on a concrete scenario: we created a set of 100 songs, all having reached position-7 in Billboard, as their best rank. The resulted set of tracks was afterwards used for testing all classifiers (the set of 100 rank-7 songs was removed from all training sets of all classifiers). In Figure 2 we present the average probabilities for the 100 rank-7 tracks as assigned by the different classifiers and indicating the likelihood of the tracks to belong to the particular hit range class. The thick line at the 50% average probability corresponds to random class assignment. We observe that classifiers corresponding to classes $1-1$, $1-3$ and $1-5$ all have probabilities below the threshold, which is perfectly correct since all tested tracks have rank position 7. Starting with the classifier for the range $1-10$, the average probabilities are showing the track position to be included in the respective intervals.

Regarding the features we used for classification, we investigated which features were especially valuable for classification. We analyzed Information Gain and

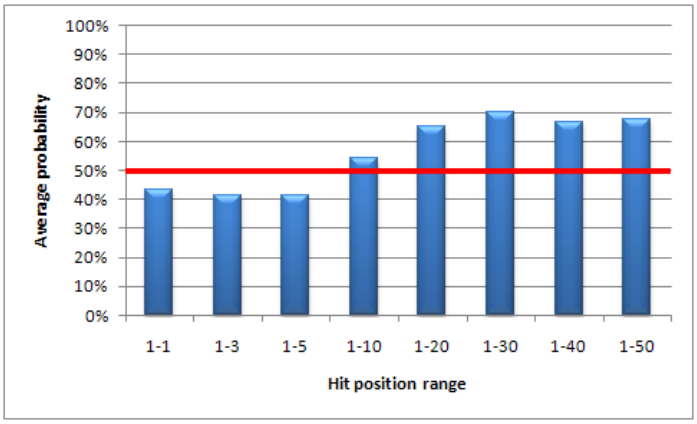

Fig. 2. Classification probability for chart position 7 averaged over 100 songs

Chi Square values for all features and found out that entity scores for artists and tracks ($ES_{A|T}$, see Equation 3) were particularly useful, ES_T being the top feature. Thus, the (prior) popularity of the artist and his earlier tracks and albums – measured as embeddedness in the *Last.fm* graph (artist-tracks-tags)– is the most distinct indicator of success for new songs. We clearly find a rich-get-richer effect: Once artists are popular chances for subsequent success are high.

6 Conclusions and Future Work

Previous attempts to identify music hits relied entirely on lyrics or audio information for clustering or classifying song corporas. By using data from a Web 2.0 music site, our approach adds a new dimension to this kind of research. Our algorithms exploit social annotations and interactions in *Last.fm* that enable both measuring intrinsic similarity of songs and finding critical early-stage effects of cumulative advantage for tracks assumed to be popular. In order to be able to make accurate predictions about evolution and hit-potential of songs, it only requires those tracks to be inside the portal for one week. The large scale experiments we performed indicate good classification accuracy for our method and compared with previous comparable work we achieve $\approx 28\%$ improvement in terms of AUC. The applications of our algorithm are manifold: record companies, radio stations, the artists themselves and last but not least, the users.

As future work we plan to experiment with an extended set of input features for the classifiers, including besides social attributes also audio and lyrics information. Since marketing is known to have a great impact on the future success of songs, we intend to study other online information sources, such as advertisements, blogs, or forums, which could as well give strong indications of a songs' hit potential and possibly underlying mechanisms of preferential attachment.

Acknowledgments. This work was partially supported by the PHAROS project funded by the European Commission under the 6th Framework Programme (IST Contract No. 045035).

References

1. Watts, D.J.: Is justin timberlake a product of cumulative advantage? New York Times, April 15 (2007)
2. Dhanaraj, R., Logan, B.: Automatic prediction of hit songs. In: 6th International Conference on Music Information Retrieval (ISMIR 2005), pp. 488–491 (2005)
3. Chon, S.H., Slaney, M., Berger, J.: Predicting success from music sales data: a statistical and adaptive approach. In: 1st ACM workshop on Audio and music computing multimedia (AMCMM 2006), pp. 83–88. ACM, New York (2006)
4. Pachet, F., Roy, P.: Hit song science is not yet a science. In: 9th International Conference on Music Information Retrieval (ISMIR 2008) (2008)
5. Agichtein, E., Castillo, C., Donato, D., Gionis, A., Mishne, G.: Finding high-quality content in social media. In: 1st ACM International Conference on Web Search and Data Mining (WSDM 2008), pp. 183–194. ACM, New York (2008)
6. Gruhl, D., Guha, R., Kumar, R., Novak, J., Tomkins, A.: The predictive power of online chatter. In: 11th ACM SIGKDD international conference on Knowledge discovery in data mining (KDD 2005), pp. 78–87. ACM, New York (2005)
7. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Trend detection in folksonomies. In: Avrithis, Y., Kompatsiaris, Y., Staab, S., O'Connor, N.E. (eds.) SAMT 2006. LNCS, vol. 4306, pp. 56–70. Springer, Heidelberg (2006)
8. Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., Tomkins, A.: Visualizing tags over time. In: 15th international conference on World Wide Web (WWW 2006), pp. 193–202. ACM, New York (2006)
9. Levy, M., Sandler, M.: A semantic space for music derived from social tags. In: 8th International Conference on Music Information Retrieval (ISMIR 2007), pp. 411–416 (2007)
10. Bischoff, K., Firan, C.S., Nejdl, W., Paiu, R.: Can all tags be used for search? In: 17th ACM Conference on Information and Knowledge Management (CIKM 2008), pp. 193–202. ACM, New York (2008)
11. Kleinberg, J.: Authoritative sources in a hyperlinked environment. Journal of the Association for Computing Machinery 46(5), 604–632 (1999)
12. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco (2005)
13. Fischer, J.E., Bachmann, L.M., Jaeschke, R.: A readers' guide to the interpretation of diagnostic test properties: clinical example of sepsis. Intensive Care Medicine Journal 29(7), 1043–1051 (2003)