Turning Experience Products into Search Products: Making User Feedback Count

Joachim Selke and Wolf-Tilo Balke
Institut für Informationssysteme
Technische Universität Braunschweig
Germany

***Abstract*—Online shopping sites are faced with a significant problem: When offering experience products, i.e., products that lack a helpful description in terms of easily accessible factual properties (e.g., wine, cigars, and movies), a lot of work and time needs to be invested to provide such information. Two very popular approaches are the introduction of sophisticated categorization systems (e.g., fruity, woody, and peppery for wines) along with manual product classification performed by experts and the addition of user feedback mechanisms (e.g., ratings or textual reviews). While user feedback typically is easy to collect, for purposes of product search, it cannot be used as easily as this is possible with a systematic categorization scheme. In this paper, we propose an effective method to automatically derive product classifications of high quality from many different kinds of user feedback. Our semi-supervised method combines advanced data extraction methods with state-of-the-art classification algorithms and only requires a small number of training examples to be created manually by experts. We prove the benefits of our approach by performing an extensive evaluation in the movie domain.**

I. Introduction

Marketing theory traditionally distinguishes between two major types of products: Search products and experience products [1]. The difference lies in how much information about major product properties can be acquired by customers without actually using the product. While search products typically can be described completely in terms of their *factual properties* (e.g., technical specifications), experience products are dominated by intangible, *perceptual properties*, which are difficult to assess without direct product experience. Typical examples of experience products are books, cars, wine, and movies. Here, emotional aspects such as fun, amusement, fantasy, and enjoyment are clearly more important than the respective product's objective utility [2].

However, this distinction is not strict. As aesthetics has become a driving factor in industrial product design, many traditional search products gradually turn into experience products. Norio Ohga, former CEO and chairman of Sony Corporation, is quoted as follows [3, p. 134]: "At Sony, we assume that all products of our competitors have basically the same technology, price, performance, and features. Design is the only thing that differentiates one product from another in the marketplace."

The difficulty of assessing perceptual product properties prior to purchase poses significant problems for online shopping. Research shows that selling experience products over the Internet is much more difficult than selling search products. In particular, customers perceive more risk when buying experience products [4], [5] and strongly prefer to buy through channels that accurately portray the characteristics of a product [6], [7].

To tackle these issues, online shopping sites try to transform experience products into search products [8], [9] by providing additional information. Probably the most popular approaches are (1) the introduction of a sophisticated categorization system to model perceptual properties explicitly (e.g., movie genres), and (2) allowing users to provide feedback about individual products (e.g., by means of textual reviews).

However, each of these approaches has its drawbacks. The use of a categorization system typically implies a tedious process of manual product classifications to be performed by experts, in particular if products need to be reevaluated continuously due to shifting opinions or perceptions in the general public [10], [11]. In addition, manual classifications are highly prone to inconsistencies [12]. However, a categorization system can directly be exploited by the shopping site to provide highly useful faceted search capabilities [13] to its customers. In contrast, user feedback perfectly reflects the perceptual properties of products [14], but usually cannot be exploited for effective product search.

In this paper, we will tap the wisdom of the crowd to resolve the above issues. After reviewing different approaches to categorization and user feedback, we show that the tedious and error-prone manual classification process can be replaced by an automatic classification system that analyzes and exploits user feedback. Our system requires just a very small number of training examples for each category, and yields a classification accuracy comparable to human judgments.

The paper is structured as follows: After reviewing related work in Section II, we carefully analyze the problem at hand in Section III. As a result of this analysis, we propose our approach in Section IV and perform an extensive experimental evaluation in Section V. We conclude in Section VI.

II. Related Work

To our knowledge, there are basically three areas in information systems research related to the handling of experience products: recommender systems, opinion mining, and multimedia databases.

Recommender systems deal with the problem of generating personalized product recommendations from user profiles collected in the past [15]. There are two major research directions: content-based and collaborative approaches. While content-based recommender systems derive their recommendations from product descriptions in structured form (e.g., by finding notebook computers made by manufacturers the user liked in the past), collaborative recommender systems analyze recurring patterns of taste and user behavior (e.g., Amazon's well-known "users who bought *x* also bought *y*" statements). Consequently, content-based approaches typically target search products, while collaborative approaches are better suited for experience products. Although there are collaborative recommender systems that try to exploit manually created categorization schemes, e.g. [14], [16], or different types of user feedback for making better recommendations, e.g. [17], [18], we are not aware of any approaches trying to infer a product's classification from such data.

Opinion mining tries to identify sentiments about a product's properties, typically from textual product descriptions or reviews [19]. Applied to a large collection of textual user feedback about a specific product, opinion mining may be used to create a condensed overview of the product's main features, where each feature is assigned a number indicating the proportion of people who spoke positively about it, see e.g. [20], [21]. However, this approach is orthogonal to product classification. While opinion mining tries to rate those features shared by products in a given domain on a positive–negative scale (e.g., "60% of all people like this phone's built-in camera"), classification aims to divide a product domain into groups (e.g., "this phone is a business phone"). Moreover, opinion mining strictly requires textual product descriptions, while our work covers different types of user feedback.

Multimedia databases are linked to three prominent types of experience products: music, pictures, and videos [22]. What makes these product types so special is that one can provide a (near-)complete representation of each product in digital form. The goal of multimedia databases is to enable content-based search on multimedia objects (e.g., finding similar songs or images showing a sunset at the beach). This is usually done by deriving a compact representation of each multimedia object by means of so-called feature extractors and using only this representation in all search algorithms. Feature extractors aim to identify those properties of objects that closely resemble human visual and auditory perception [23], [24]. Although there exists work on performing an automatic classification of multimedia objects based on feature representations, e.g. [25], [26], these approaches are always bound to specific media types and corresponding feature extractors, and thus cannot be generalized to other kinds of experience products. As we will see below, the approach presented in this paper also is well-suited for multimedia content, while avoiding analysis of multimedia files, which typically is computationally expensive.

In summary, although there is a plethora of different specialized approaches to deal with experience products, we are not aware of any domain-independent work on classifying experience products based on user-provided feedback.

III. Problem Analysis

For the scope of this paper we assume that we are given some domain of experience products and a database containing a list of actual products. To keep things simple, we assume that apart from product IDs (e.g., product names), no additional product information is available in the database.

As an illustrating example, we use the domain of movies throughout this paper. This is for several reasons: First, movies are particularly good example of experience products as they appeal to a wide range of customers and are almost purely experiential. It has been shown that, when selecting movies, consumers rely far more on perceptual movie features (funny, romantic, scary, . . .) than factual ones (actors, directors, release year, . . .) [27]. Second, the movie domain provides sophisticated categorization schemes and extensive manual movie classifications we can use to evaluate our proposed approach. Third, there are large collections of user-provided feedback publicly available, which also are required to evaluate our approach properly. Fourth, driven by companies such as Netflix[1] and LOVEFiLM[2], online video rental and on-demand streaming has become a mass market, still undergoing significant growth. In their first-quarter 2011 financial results, market leader Netflix reports to have about 23 million customers in the United States and Canada, with an average monthly revenue of about 250 million dollars.[3] However, as we do not make any assumptions being specific to the movie domain, all methods and results presented in this paper can be directly transferred to other types of experience products.

To make perceptual movie features available to customers, companies introduced corresponding categorization schemes paired with manual classification of each individual product by domain experts. In addition, public feedback mechanism have been made to customers. Next, we give examples of how these two approaches have been established in the movie domain.

A. Categorization Schemes

Traditionally, genres are used to describe a movie's most important perceptual features. One typically distinguishes about twenty generally accepted genres such as Comedy, Documentary, Horror, and Science Fiction [10], [28]. However, there is no generally accepted genre categorization scheme. Usually, a movie may belong to multiple genres at once.

Faced with the need to differentiate between many hundred thousands of movies, many movie databases decided to go beyond the major genres and introduced more complex categorization schemes. For example, the rental service Netflix expanded its simple genre list into a taxonomy covering 485 genres.

Some companies even felt the need to completely leave genres behind and introduce more specific categories. For example, in addition to using a sophisticated genre taxonomy,

[1] http://www.netflix.com

[2] http://www.lovefilm.de

[3] http://netflix.mediaroom.com/index.php?s=43&item=390

the commercial metadata provider AllMovie[4] manually classifies its 440,000 movies with respect to more than 5,000 different moods, themes, tones, and types (Ensemble Film, Haunted By the Past, Intimate, ...). Probably the most extreme approach to movie categorization has been implemented by the recommendation service Clerkdogs[5]. In addition to using about 200 genres, human experts at Clerkdogs rate each of their movies with respect to 37 different attributes on a 12-point scale (Character Depth, Geek Factor, Violence, ...).

Given the huge number of movies covered by these databases, introducing new categories becomes a major issue, as it currently requires the manual evaluation of all existing database entries. For example, the Internet Movie Database[6] (IMDb) currently is faced with the problem of classifying its 1.9 million titles with respect to a number of new genres.[7]

B. User Feedback

Apart from making perceptual product features explicit through the use of categorization schemes, many online shopping sites allow their customers extend company-provided product descriptions with detailed feedback. In the movie domain, there are three major types of user feedback.

The first and most simple type of user feedback are *product ratings*, which usually can be provided by registered customers on a numerical scale (e.g., 1–5 stars or thumbs-up/thumbs-down). Popular examples are Netflix, IMDb, and Flixster[8]. Companies record this information as rating triples of the form (movieID, userID, rating). Ratings are typically used for displaying the public's average opinion about a movie or for providing personalized movie recommendations.

The second popular type of feedback are *tags* as they are used e.g. by the recommendation service MovieLens[9]. Here, customers may assign arbitrary terms to each movie, which in turn are used to generate a tag cloud for the respective movie. Moreover, tags can be used for navigational movie search, as it has been demonstrated by Movie Tuner [29].

Third, some movie portals allow users to write *textual reviews* about their products. This approach is featured by IMDb and Rotten Tomatoes[10], among others. Movie reviews usually are only displayed to customers as part of the respective product description. A notable exception is Nanocrow[11]d, which analyzes reviews by means of information retrieval methods to generate three-word groups used to describe movies, so-called nanogenres.

To get a better understanding of what amount of user feedback is available, we downloaded and analyzed the data made public by three different movie databases. The results are shown in Table I. There is a clear trend: Users are much more likely to provide ratings than tags, the same is true for the relationship between tags and reviews. Intuitively, this immediately makes sense, as proving a rating rarely requires more than a single mouse click, providing a tag in addition requires some cognitive effort, and writing a whole review may take a significant amount of time and effort. On the other hand, we would expect a single review to be more informative than a single tags, which in turn is expected to be more informative than a single rating, which essentially tells nothing about a movie's perceptual properties. Therefore, the amount of data available from each type of user feedback is inversely related to the amount of information given by each single feedback item.

This raises some interesting questions: How useful is each type of user feedback for purposes of automatic classification? What is more important: The number of users who contributed feedback or the wealth of information given by each piece of feedback? We will address these questions in our experimental evaluation in Section V.

IV. Making User Feedback Count

The above problem analysis shows that the main problem with product categorization is the manual work required to classify all products accordingly. On the other hand, there is plenty of user feedback available, which could be used to fuel an automatic classification system. This way, we would be able to tap the wisdom of the crowd, without even asking a single user to perform a tedious classification task with respect to some complex categorization scheme. Given a sufficient accuracy in classification, such a system would significantly reduce the manual work to be performed by operators of online shopping sites, while retaining all the benefits of product classification.

Without loss of generality, let us assume that we are dealing with only a single product category C, with each product either belonging to this category or not. We define the task to be performed as follows: Given the product IDs of n products clearly belonging to C and n products clearly not belonging to C, use the available user feedback to determine, for each remaining product, whether it is contained in C or not. As the size-$2n$ list of training examples must be created manually by some domain expert, n is typically very small, e.g. $n = 10$.

As our goal is to provide a systematic solution that can cope with different kinds of product domains as well as different types of user feedback, but also is able to incorporate state-of-the art classification techniques in a flexible way, we decided to break down our problem into the following two steps:

1) Given a set of products along with user feedback for each product, create a *semantic space* of products, i.e., assign a point in d-dimensional coordinate space to each product such that products with (dis)similar user feedback have (dis)similar coordinates.
2) Given n positive and n negative training examples, automatically classify all remaining products by applying a standard semi-supervised classification algorithm that relies only on the semantic space representation of products.

[4] http://www.allmovie.com
[5] http://www.clerkdogs.com
[6] http://www.imdb.com
[7] http://www.imdb.com/board/bd0000167/thread/165524169
[8] http://www.flixster.com
[9] http://www.movielens.org
[10] http://www.rottentomatoes.com
[11] http://www.nanocrowd.com

Service	#movies	#ratings	#tags	#reviews
IMDb	1,900,000	198,000,000	3,600,000	1,900,000
Rotten Tomatoes & Flixster	190,000	2,312,000,000	–	580,000
MovieLens	16,000	18,000,000	261,000	–

TABLE I
AMOUNT OF USER FEEDBACK COLLECTED BY DIFFERENT MOVIE DATABASES.

The first step transforms the user feedback into a unified semantic space representation, where the model parameter d is chosen in advance; here $d \approx 100$ is a typical value. Semantic spaces are a common tool in cognitive psychology to model human conceptual understanding [30], [31]. They have also become popular in information retrieval, pattern recognition, and statistics to describe complex objects in a condensed way [32]–[34]. Moreover, for all three types of user feedback discussed above, there already are methods available to create informative semantic spaces. We will briefly discuss these methods below.

The second step perfectly corresponds to the task performed by general-purpose algorithms for semi-supervised classification [35]. Therefore, also this step of our approach can be handled easily by existing methods, which enables a very high level of flexibility.

To create a semantic space representation from the three types of user feedback discussed in the previous section, we recommend the following methods:

- *Ratings.* Research in recommender systems recently made significant progress through the introduction of so-called factor models [36]. These models implicitly use semantic spaces to generate personalized recommendations. In previous work [37], we found that these semantic spaces contain enough information to be also useful for purposes different from recommendations. However, there are a lot of different factor models to choose from, which all rely on different methods to create semantic spaces. In recent studies we found Euclidean embedding [38] to be most effective [12]. This technique represents both users and items in a shared space such that the Euclidean distance between a user and an item is inversely proportional to the strength of preference the user indicated when rating the item.
- *Reviews.* To turn reviews into a semantic space representation, information retrieval research offers a wide range of methods. Probably the most popular ones are Latent Semantic Indexing (LSI) [32] and Latent Dirichlet Analysis (LDA) [34]. While LSI uses a technique from linear algebra to project the term–document space into an optimally information-preserving subspace of lower dimensionality, LDA is based on a probabilistic generative model that tries to explain how documents and terms are related to a small number of different topics. However, in our setting we are not interested in mapping review documents into semantic space but products. Therefore, we recommend to merge all reviews of a given product into a single review document. This can be done e.g. by just appending the individual reviews to one another.
- *Tags.* For the analysis of tags, we can think of two different approaches: If the individual users who provided each tag are unknown, the list of tags assigned to a product could simply be treated as a text document and handled in the same way as a review. If tagging information is available as (product, user, tag) triples, factor models can be applied. However, unlike numerical ratings, tags cannot be ordered on a single scale. Therefore, more advanced factor models have been proposed in the area of tag recommendation: so-called tensor models [39], [40]. Tensor models could also be applied to textual reviews if individual authorship is known and each user tends to review multiple products [41].

Regarding the implementation of the classification step, we found support-vector machines [42] and k-nearest neighbor classifiers [43] to be very effective [37]. However, depending on the product domain and the algorithm used to create the semantic space, other methods might be more appropriate. As it is common practice in machine learning research, for any specific real-world application, the optimal choice of step-1 and step-2 methods should be determined empirically.

Another advantage of our modular two-step approach is that is can easily be modified to suit new requirements. For example, instead of classifying products in a binary fashion, one might want to apply a scoring system as it is used by the movie database Clerkdogs (see Section III-A). Then, the problem to be solved becomes a regression task: given a small training set of products that have been scored manually with respect to a given criterion, predict the score of all remaining products. In a preliminary study [44], we found support-vector regression [45] to be helpful. However, many other standard regression algorithms should be suitable as well.

V. EXPERIMENTAL EVALUATION

To evaluate the benefits of the proposed approach, we put it on test with real-world data retrieved from different movie databases on the Web. We downloaded and cross-referenced the following data:

- manual genre classifications from IMDb, Netflix, and Rotten Tomatoes (RT),
- customer reviews from IMDb,
- ratings from Netflix (published as part of the Netflix Prize[12] in 2006), and
- manual scorings from Clerkdogs.

[12] http://www.netflixprize.com

Unfortunately, we have not been able to retrieve any tagging data, as this type of information is not made available by any major movie database in form of (movie, user, tag) triples. Therefore, we only evaluated our approach for rating and review feedback data.

In summary, our unified data sets consists of 10,562 movies and has the following properties:

- All movies have a genre classification in each of the three databases, with 7 genres being shared by all databases.
- All movies have been reviewed at least once in IMDb, resulting in a total number of 918,193 reviews.
- Apart from four exceptions, each movie has been rated at least 100 times in Netflix, in a total number of 85,651,367 ratings.
- 6,067 movies have been scored by Clerkdogs with respect to at least one of 37 criteria, resulting in a total number of 63,361 scorings on a 12-point scale.

First, we tested the accuracy of automatic genre classification. For the first step of our method, we implemented the following algorithms: Euclidean embedding [38] on ratings (RatEE), Latent Semantic Indexing [32] on reviews (RevLSI), and latent Dirichlet allocation [34] on reviews (RevLDA). For all these algorithms, standard parameters as reported in the literature have been used. As dimensionality of the semantic spaces to be created we chose $d = 100$. In previous work, we did not find any significant benefits from using larger dimensionalities [37]. For the second step of our method, we implemented support vector machines [42] with RBF kernels (SVM) and 5-nearest neighbor classification [43] based on Euclidean distance (5NN). The parameters of the support vector machines have been kept the same for all three semantic spaces. We chose these methods because they are based on an intuitive geometric ideas that fit those underlying semantic spaces. Moreover, in a preliminary study, we found these methods to be highly useful [37].

As the manual genre assignments made by the three movie databases show a significant amount of inconsistencies [12], we needed to derive a reference genre assignment from this data. Therefore, for each of the seven genres, we defined our ground truth to be the majority's opinion. That is, we consider a movie to belong to genre X if and only if at least two of the three databases classify it as X.

For each genre, we randomly picked n clearly positive and n clearly negative examples of the respective genre, where a movie is considered as a clear example if all three databases agree on the classification. To evaluate how the amount of available training data affects the classification performance, we independently tested $n = 10$, $n = 25$, and $n = 50$.

To measure classification performance, we could not apply the popular accuracy measure (relative number of correct classifications). This is because there is substantial imbalance between positive and negative genre assignments. Using accuracy would result in the strange situation that a naive classifier, which classifies every movie as *not Sport* would achieve a near-perfect accuracy of 96.3%, as only 3.7% of all movies are seen as Sport movies by the majority of databases. Several alternative measures have been proposed to evaluate classification performance in presence of class imbalance [46], [47]. A popular choice is the g-mean, which is the geometric mean of sensitivity (accuracy on all movies truly belonging to the genre) and specificity (accuracy on all movies truly not belonging to the genre). The above naive classifier achieves 0% g-mean, as the g-mean punishes significant differences between sensitivity and specificity.

Table II and Table III show our results, where each entry is the mean over 20 random repetitions. The g-mean values corresponding to the original classifications made by IMDb, Netflix, and RT are listed for reference in Table IV. We can observe that RevLSI shows the best classification performance, the other two methods perform slightly worse, with RatEE being slightly better than RevLDA. When comparing the mean g-mean values to the reference values in Table IV, we see that there only is a very small difference. However, as the three databases contribute to our ground truth, the values shown in Table IV are higher than they we would expect them to be when comparing to an independent source of ground truth. Therefore, we can conclude that *automatic classification performs similarly good as manual classification*. This is even true for small training sets ($n = 25$). We can also observe that semantic spaces created from ratings enable a surprisingly good classification performance. We initially expected spaces generated from review data to be much more effective as reviews may explicitly contain information about genre assignments, but ratings cannot. Finally, we can see that SVM and 5NN show comparable performance, with SVM having a slight advantage.

We also wanted to know whether the classification performance can still be improved by combining the semantic spaces derived by different methods and from different sources of user feedback. To this end, we created different 200-dimensional semantic spaces by merging (i.e., appending) the coordinates from two 100-dimensional spaces. We created all three possible semantic spaces, which we evaluated in Table V using the SVM approach. We see that for neither space the overall performance is better than those seen with the original RevLSI space. We conclude that all relevant genre information is already contained in the RevLSI space, with the amount of information contained in the other two spaces being a strict subset.

Finally, we evaluated the effectiveness of our approach for regression tasks as required by Clerkdogs. We selected the ten scoring criteria most often used in Clerkdogs and, for each criterion, randomly chose n movies scored with respect to this criterion. Then, we used support vector regression (SVR) [45] to predict the scores of all remaining movies that have been scored by Clerkdogs with respect to the criterion. We used the same SVR parameters as in the previous SVM experiment. To measure the accuracy of predictions, we computed the Pearson correlation between the predicted scores and the ones assigned by Clerkdogs' experts. Our results are shown in Table VI. Again, each entry is the mean of 20 independent random repetitions. The most notable result is that all Pearson correlations are positive, that is, it was always possible to learn the target criterion correctly at least to a certain degree. While for some

	RatEE/SVM			RevLSI/SVM			RevLDA/SVM		
Genre	$n = 10$	$n = 25$	$n = 50$	$n = 10$	$n = 25$	$n = 50$	$n = 10$	$n = 25$	$n = 50$
Comedy	0.66	0.70	0.72	0.64	0.74	0.77	0.72	0.79	0.81
Documentary	0.68	0.75	0.76	0.80	0.86	0.88	0.65	0.73	0.75
Drama	0.66	0.68	0.69	0.64	0.69	0.72	0.69	0.72	0.73
Family	0.75	0.78	0.79	0.73	0.78	0.81	0.69	0.75	0.78
Horror	0.79	0.83	0.83	0.82	0.85	0.87	0.76	0.81	0.85
Romance	0.62	0.66	0.67	0.60	0.66	0.70	0.54	0.63	0.69
Sport	0.61	0.63	0.64	0.73	0.77	0.79	0.46	0.49	0.48
Mean	0.68	0.72	0.73	0.71	0.77	0.79	0.64	0.70	0.73

TABLE II
ACCURACY (G-MEAN) OF AUTOMATIC CLASSIFICATION (n POSITIVE AND n NEGATIVE EXAMPLES) BY MEANS OF SVM.

	RatEE/5NN			RevLSI/5NN			RevLDA/5NN		
Genre	$n = 10$	$n = 25$	$n = 50$	$n = 10$	$n = 25$	$n = 50$	$n = 10$	$n = 25$	$n = 50$
Comedy	0.62	0.66	0.68	0.68	0.74	0.75	0.70	0.74	0.76
Documentary	0.66	0.72	0.74	0.77	0.81	0.83	0.65	0.69	0.70
Drama	0.65	0.68	0.70	0.67	0.73	0.74	0.69	0.72	0.73
Family	0.76	0.80	0.81	0.69	0.79	0.81	0.65	0.71	0.73
Horror	0.71	0.76	0.78	0.82	0.87	0.88	0.72	0.77	0.79
Romance	0.61	0.65	0.66	0.65	0.70	0.72	0.57	0.62	0.64
Sport	0.62	0.62	0.60	0.78	0.83	0.83	0.51	0.50	0.48
Mean	0.66	0.70	0.71	0.72	0.78	0.79	0.64	0.68	0.69

TABLE III
ACCURACY (G-MEAN) OF AUTOMATIC CLASSIFICATION (n POSITIVE AND n NEGATIVE EXAMPLES) BY MEANS OF 5NN.

	IMDb	Netflix	RT	Mean
Comedy	0.94	0.78	0.94	0.89
Documentary	0.91	0.87	0.93	0.90
Drama	0.90	0.78	0.92	0.87
Family	0.85	0.89	0.84	0.86
Horror	0.91	0.83	0.94	0.89
Romance	0.91	0.78	0.64	0.78
Sport	0.87	0.90	0.39	0.72
Mean	0.89	0.83	0.80	0.84

TABLE IV
ACCURACY (G-MEAN) OF MANUAL CLASSIFICATION.

criteria we have been able to achieve a quite high accuracy (e.g., *character depth* and *suspense*), there also have been criteria which proved to be hard to learn (e.g., *fast pace*). We can also observe that for most criteria we can obtain a correlation between 0.2 and 0.3, even for a very small number of training examples. Given that the correlation coefficient of a perfect estimation is 1 and that of a naive baseline (estimating each score by the average score in the training set) is 0, the estimated scores do not seem to be very accurate. But this assessment relies on the assumption that the scores provided by Clerkdogs' experts are indeed objectively correct. In our analysis of Clerkdogs' data we found nine movies that occur twice in the movie database. In total we located 63 movie–criterion combinations which have been assessed twice, and used this data to estimate the inter-expert consistency. We found that the Pearson correlation between the rating pairs is only 0.60, which is an surprisingly low value [44]. Therefore, our results a definitely worth improving but the same is definitely true for manual score assignments.

VI. CONCLUSION

In this paper, we proposed a flexible and effective method to relief shopping sites from the burden of manually classifying or scoring experience products. By combining research from many different areas, our method ties loose end together and provides a solid foundation for further extensions. Our extensive evaluation on a large real-world data set proves that our approach is suited to completely replace manual classifications (and maybe even scorings) through the analysis of user feedback, which does not have to be related to any classification information at all.

In future work, we plan to evaluate whether our approach also can be applied successfully to multi-domain shopping sites such as Amazon. Moreover, we are currently investigating why the scoring performance is rather low for some criteria, and what other methods can be used to even better achieve our goal of making user feedback count.

REFERENCES

[1] P. Nelson, "Information and consumer behavior," *Journal of Political Economy*, vol. 78, no. 2, pp. 311–329, 1970.

[2] E. C. Hirschman and M. B. Holbrook, "Hedonic consumption: Emerging concepts, methods and propositions," *Journal of Marketing*, vol. 46, no. 3, pp. 92–101, 1982.

[3] T. J. Peters, *Re-Imagine! Business Excellence in a Disruptive Age*. Dorling Kindersley, 2003.

	RatEE+RevLSI/SVM			RatEE+RevLDA/SVM			RevLSI+RevLDA/SVM		
Genre	$n = 10$	$n = 25$	$n = 50$	$n = 10$	$n = 25$	$n = 50$	$n = 10$	$n = 25$	$n = 50$
Comedy	0.67	0.72	0.75	0.70	0.75	0.77	0.74	0.79	0.82
Documentary	0.75	0.79	0.81	0.71	0.77	0.79	0.75	0.83	0.84
Drama	0.67	0.70	0.71	0.69	0.71	0.72	0.70	0.73	0.74
Family	0.77	0.79	0.80	0.76	0.79	0.80	0.74	0.79	0.81
Horror	0.81	0.85	0.86	0.81	0.84	0.85	0.82	0.86	0.87
Romance	0.63	0.67	0.69	0.63	0.67	0.69	0.63	0.71	0.73
Sport	0.65	0.65	0.67	0.61	0.62	0.64	0.65	0.68	0.70
Mean	0.71	0.74	0.75	0.70	0.74	0.75	0.72	0.77	0.79

TABLE V

ACCURACY (G-MEAN) OF AUTOMATIC CLASSIFICATION (n POSITIVE AND n NEGATIVE EXAMPLES) BY MEANS OF SVM.

	RatEE/SVR			RevLSI/SVR			RevLDA/SVR		
	$n = 20$	$n = 50$	$n = 100$	$n = 20$	$n = 50$	$n = 100$	$n = 20$	$n = 50$	$n = 100$
Character Depth	0.57	0.63	0.67	0.46	0.60	0.65	0.64	0.69	0.70
Cinematography	0.25	0.32	0.36	0.23	0.34	0.41	0.30	0.41	0.45
Complexity	0.40	0.44	0.46	0.35	0.42	0.48	0.37	0.45	0.50
Downbeat	0.21	0.26	0.32	0.17	0.36	0.34	0.17	0.25	0.34
Fast Pace	0.06	0.09	0.12	0.06	0.08	0.13	0.05	0.10	0.13
Hollywood Feel	0.16	0.24	0.28	0.12	0.19	0.25	0.05	0.11	0.18
Offbeat	0.30	0.34	0.38	0.15	0.24	0.32	0.18	0.23	0.32
Romance	0.24	0.33	0.40	0.26	0.37	0.46	0.21	0.33	0.45
Suspense	0.30	0.40	0.45	0.34	0.46	0.53	0.24	0.35	0.43
Violence	0.28	0.40	0.46	0.32	0.46	0.53	0.19	0.35	0.45
Mean	0.28	0.35	0.39	0.25	0.34	0.41	0.24	0.33	0.40

TABLE VI

ACCURACY (PEARSON CORRELATION) OF AUTOMATIC SCORING (n EXAMPLES) BY MEANS OF SVR.

[4] A. V. Citrin, D. E. Stem, E. R. Spangenberg, and M. J. Clark, "Consumer need for tactile input: An internet retailing challenge," *Journal of Business Research*, vol. 56, no. 11, pp. 915–922, 2003.

[5] J. Peck and T. L. Childers, "To have and hold: The influence of haptic information on product judgments," *Journal of Marketing*, vol. 67, no. 2, pp. 915–922, 2003.

[6] R. R. Burke, "Technology and the customer interface: What customers want in the physical and virtual store," *Journal of the Academy of Marketing Science*, vol. 30, no. 4, pp. 411–432, 2002.

[7] K.-P. Chiang and R. R. Dholakia, "Factors driving consumer intention to shop online: An empirical investigation," *Journal of Consumer Psychology*, vol. 13, no. 1–2, pp. 177–183, 2003.

[8] L. R. Klein, "Evaluating the potential of interactive media through a new lens: Search versus experience goods," *Journal of Business Research*, vol. 41, no. 3, pp. 195–203, 1998.

[9] M. Nakayama, N. Sutcliffe, and Y. Wan, "Has the Web transformed experience goods into search goods?" *Electronic Markets*, vol. 20, no. 3/4, pp. 251–262, 2010.

[10] D. Chandler, "An introduction to genre theory," 1997, available from http://www.aber.ac.uk/media/Documents/intgenre.

[11] R. Stam, *Film Theory: An Introduction*. Blackwell, 2000.

[12] J. Selke and W.-T. Balke, "TEAMWORK: A data model for experience products," Institut für Informationssysteme, Technische Universität Braunschweig, Germany, ifis Technical Report, 2011.

[13] D. Tunkelang, *Faceted Search*, ser. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool, 2009.

[14] I. Pilászy and D. Tikk, "Recommending new movies: Even a few ratings are more valuable than metadata," in *Proceedings of RecSys 2009*, 2009, pp. 93–100.

[15] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds., *Recommender Systems Handbook*. Springer, 2011.

[16] Y. Zhang and J. Koren, "Efficient Bayesian hierarchical user modeling for recommendation system," in *Proceedings of SIGIR 2007*, 2007, pp. 47–54.

[17] N. Jakob, S. H. Weber, M.-C. Müller, and I. Gurevych, "Beyond the stars: Exploiting free-text user reviews to improve the accuracy of movie recommendations," in *Proceedings of TSA 2009*, 2009, pp. 57–64.

[18] K. H. L. Tso-Sutter, L. B. Marinho, M.-C. Müller, and L. Schmidt-Thieme, "Tag-aware recommender systems by fusion of collaborative filtering algorithms," in *Proceedings of SAC 2008*, 2008, pp. 1995–1999.

[19] B. Pang and L. Lee, *Opinion Mining and Sentiment Analysis*, ser. Foundations and Trends in Information Retrieval. Now Publishers, 2008, vol. 2.

[20] M. Hu and B. Liu, "Mining opinion features in customer reviews," in *Proceedings of AAAI 2004*, 2004, pp. 755–760.

[21] ——, "Mining and summarizing customer reviews," in *Proceedings of KDD 2004*, 2004, pp. 168–177.

[22] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "Content-based multimedia information retrieval: State of the art and challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 2, no. 1, pp. 1–19, 2006.

[23] ——, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, no. 1, pp. 262–282, 2007.

[24] C. Jörgensen, A. Jaimes, A. B. Benitez, and S.-F. Chang, "A conceptual framework and empirical research for classifying visual descriptors," *Journal of the American Society for Information Science and Technology*, vol. 52, no. 11, pp. 938–947, 2001.

[25] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[26] L.-Q. Xu and Y. Li, "Video classification using spatial-temporal features and PCA," in *Proceedings of ICME 2003*, 2003, pp. 485–488.

[27] E. Cooper-Martin, "Consumers and movies: Some findings on experiential products," in *Advances in Consumer Research*, 1991, vol. 18, pp. 372–378.

[28] C. Preston, "Film genres," in *The International Encyclopedia of Communication*, W. Donsbach, Ed. Blackwell, 2008.

[29] J. Vig, S. Sen, and J. Riedl, "Navigating the tag genome," in *Proceedings of IUI 2011*, 2011, pp. 93–102.

[30] R. L. Goldstone and A. Kersten, "Concepts and categorization," in

Experimental Psychology, ser. Handbook of Psychology, A. F. Healy and R. W. Proctor, Eds. John Wiley & Sons, 2003, vol. 4, pp. 599–621.

[31] P. Gärdenfors, "Conceptual spaces as a framework for knowledge representation," *Mind and Matter*, vol. 2, no. 2, pp. 9–27, 2004.

[32] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to Latent Semantic Analysis," *Discourse Processes*, vol. 25, no. 2–3, pp. 259–284, 1998.

[33] I. T. Jolliffe, *Principal Component Analysis*, 2nd ed., ser. Springer Series in Statistics. Springer, 2002.

[34] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[35] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Springer Series in Statistics. Springer, 2006.

[36] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*, 2011, pp. 145–186.

[37] J. Selke and W.-T. Balke, "Extracting features from ratings: The role of factor models," in *Proceedings of M-PREF 2010*, 2010, pp. 61–66.

[38] M. Khoshneshin and W. N. Street, "Collaborative filtering via Euclidean embedding," in *Proceedings of RecSys 2010*, 2010, pp. 87–94.

[39] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Tag recommendations based on tensor dimensionality reduction," in *Proceedings of RecSys 2008*, 2008, pp. 43–50.

[40] Y. Cai, M. Zhang, D. Luo, C. Ding, and S. Chakravarthy, "Low-order tensor decompositions for social tagging recommendation," in *Proceedings of WSDM 2011*, 2011, pp. 695–704.

[41] D. Cai, X. He, and J. Han, "Tensor space model for document analysis," in *Proceedings of SIGIR 2006*, 2006, pp. 625–626.

[42] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.

[43] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of improving k-nearest-neighbor for classification," in *Proceedings of FSKD 2007*, 2007, pp. 679–683.

[44] J. Selke, S. Homoceanu, and W.-T. Balke, "Conceptual views for entity-centric search: Turning data into meaningful concepts," *Computer Science: Research and Development*, to appear.

[45] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[46] Q. Gu, L. Zhu, and Z. Cai, "Evaluation measures of the classification performance of imbalanced data sets," in *Proceedings of ISICA 2009*, ser. Communications in Computer and Information Science, vol. 51. Springer, 2009, pp. 461–471.

[47] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.