Using Game Reviews to Recommend Games

Michael Meidl and Steven Lytinen
College of Computing and Digital Media
DePaul University
243 S. Wabash
Chicago, IL 60604

Kevin Raison
Chatsubo Labs
9708 1st Ave NW
Seattle, WA 98117

Abstract

We present a recommender system intended to be used by a community of gamers. The system uses free-form text reviews of games written by the members of the community, along with information about the games that a particular user likes, in order to recommend new games that are likely to be of interest to that user. The system uses the frequency of co-occurrence of word pairs that appear in the reviews of a game as features that represent the game. The pairs consist of adjectives and *context words*; i.e., words that appear close to an adjective in a review. Because of the extremely large number of possible combinations of adjectives and context words, we use *information-theoretic co-clustering* of the adjective-context word pairs to reduce the dimensionality. Games are represented using the standard information retrieval vector space model, in which vector features are based on the frequency of occurrence of co-cluster pairs. We present the results of three experiments with our system. In the first experiment, we use a variety of strategies to relate frequencies of co-cluster pairs to vector features, to see which produces the most accurate recommendations. In the second, we explore the effects of co-cluster dimensionality on the quality of our system's recommendations. In the third experiment, we compare our approach to a baseline approach using a bag-of-words technique and conclude that our approach produces higher quality recommendations.

Introduction

We have developed a game recommender system, which produces recommendations of games that are likely to be of interest to a user based on two factors: (a) the game reviews that a community of gamers have written about a collection of games, and (b) the games in the collection for which that user has written positive reviews. Our system represents a game using the traditional information retrieval *vector space model* (Salton, Wong, and Yang 1975). A game is represented as a vector made up of features that represent frequency of co-occurrence of word pairs which appear in the game's reviews. The word pairs consist of adjectives which are particularly salient in the description of a game, and *context words*; i.e., words which appear in a review within a window of up to two words before or after the appearance of an adjective.

Since the number of pairs of adjectives and context words in the collection of reviews is unmanageably large (over 3,500,000 pairs), our system uses *information-theoretic co-clustering* (Dhillon, Mallela, and Modha 2003) to produce clusters of adjectives, as well as clusters of context words. More precisely then, the features in a vector representing a game reflect the frequency of co-occurrence in its reviews of pairs of clusters: one a cluster of adjectives, and the other a cluster of context words. The use of co-clustering dramatically reduces the size of the feature space.

Our system produces recommendations of new games to a user by selecting at random a small number of "seed" games that the user has positively reviewed. The vector space representation is used to find games that are similar to the seeds, which are then recommended to the user. Similarity of any two games is measured in our system by calculating the cosine of the angle between the vectors which represent the two games. If our system generates n recommendations, the recommended games are the n closest games to the seeds, as measured by the cosine similarity.

The notion that adjectives are important in capturing the essence of a game review (or other types of reviews, for that matter) has been suggested, among others, by Zagal and Tomuro (2010). Our selection of which adjectives are salient to the description of games is taken from this work. We have used the co-clustering of adjectives and context words in our own previous work (Raison *et al.* 2012) to capture further information expressed in game reviews. Context words help to differentiate the dimensions along which adjectives can capture positive and negative sentiment about a game. For example, a game review might contain the phrases "incredible graphics" as well as "horrible gameplay". Therefore, the fact that "incredible" is used to describe one dimension of a game (and "horrible" another) is important for our system to represent.

Although our approach borrows from the work mentioned above, the previous work used these techniques in order to cluster games based on user reviews, and then to assess the quality of the clusters by examining the similarity of games that were clustered together. Thus, our current work is novel in its application of these techniques to the recommender task, and also in its quantitative assessment of the effects of

co-clustering on the accuracy of the recommendations produced by the system (as discussed in the *Results* section below).

In the remainder of this paper, we present in more detail the co-clustering technique and its use in generating the features that represent games in our system. We also discuss the use of the vector space model in our system's generation of recommendations. Finally, we present the results of three experiments we conducted, that demonstrate the quality of the recommendations produced by our system. In the first experiment, we compared the use of a variety of metrics for relating vector features to frequencies of co-cluster pairs, to see which metric produces the most accurate predictions. In the second, we explored the effects of co-cluster dimensions on the quality of our system's recommendations. Finally, we compared our use of word co-occurrence frequencies to a simpler "bag-of-words" approach. The results indicate that the use of co-clustering of word pairs is superior to bag-of-words.

Game Reviews and Co-clusters

The corpus of game reviews that our system uses is taken from the work of Zagal and Tomuro (2010). These reviews were collected from the *GameSpot* Web site (www.gamespot.com). The dataset contains approximately 400,000 reviews of 8,279 different games. For each game, our system examined all of the reviews of that game, and counted frequencies of the occurrence of selected adjectives and their context words. The adjectives are those determined to be the most relevant to the expression of sentiment in game reviews (see Zagal and Tomuro 2010 for details). Context words are "open class" words (e.g., nouns and verbs, but not prepositions) which appear in a review within a window of up to two words before or after an adjective. In total, the game reviews contain over 700 relevant adjectives and about 5,000 context words.

To illustrate the collection of adjective-context word pairs, consider this excerpt from a review of *Grand Theft Auto IV*:

> GTA 4 is a good game, the graphics are great, the gameplay is great, and the attention to detail is incredible.

The adjective-context word pairs extracted from this sentence are:

> good game
> great graphics
> great gameplay
> incredible detail

As a result of the extraction of adjective-context word pairs, our system produces a data file for each game, containing frequency counts of each adjective-context word pair in all of the reviews for that game. For example, here is a portion of the frequency counts for the adjective "incredible" in the *Grand Theft Auto IV* reviews (this adjective appears with too many context words to include all of them in the list below).

> incredible: funny 1, play 1, detail 7, achievement 2, soundtrack 1, look 2, details 3, graphics 8, character 1, game 12, sound 1, story 5

The word "detail", for example, appears as a context word for the adjective "incredible" a total of 7 times in the reviews of *Grand Theft Auto IV*.

Because of the extremely high number of adjective-context word pairs (there are potentially over 3,500,000 such pairs in the dataset), our system uses *information-theoretic co-clustering* (Dhillon, Mallela, and Modha 2003) to cluster both adjectives and context words into subsets, in the same way as in our previous work (Raison *et al.* 2012). The co-clustering technique selects subsets of adjectives and of context words in such as way as to minimize the loss of information resulting from replacing single words with co-clusters. Formally, mutual information $I(x, y)$ of two random variables x and y[1] as follows:

$$I(x, y) = \frac{log(p(x,y))}{p(x)p(y)}$$

Information loss is defined as:

$$L(x, y) = I(x, y) - I(c_1, c_2)$$

$I(x, y)$ is the mutual information of the lexical items x and y, and $I(c_1, c_2)$ is the mutual information of the clusters c_1 and c_2, where $x \in c_1$ and $y \in c_2$.

The co-clustering algorithm requires that the number of co-clusters for both dimensions (adjectives and context words) be specified as part of the input to the algorithm. In the first experiment described in the *Results* section, we discuss the effects of varying the number of co-clusters.

In Table 1, we show some examples of a few of the co-clusters that are generated for 30×100 co-clusters (i.e., 30 adjective clusters and 100 context word clusters). Intuitively, it appears that clusters capture some semantic commonalities among words in a cluster, although typically not all clustered words are synonyms or words that are obviously related to each other. For example, adjective cluster 1 in general contains words that express very positive sentiment ("incredible", "impressive", "astounding", etc.) although there are exceptions ("mediocre", "rounded"). In context word cluster 1, words such as "feeling", "mood", "manner", "creepy", and "disturbing", while not synonyms, all address an emotional aspect about how a game feels.

Adjective clusters sometimes contain words with opposite sentiment, such as the inclusion in cluster 2 of "easy" and "hard", "single-player" and "multi-player". A possible cause is that adjectives with opposite sentiments often can be used in similar contexts (e.g., "easy gameplay" vs. "hard gameplay", or "single-player mode" vs. "multi-player mode"). Additional *a priori* knowledge about positive vs. negative sentiment that adjectives can express would be helpful in minimizing this problem. Although use of *a priori* knowledge about positive or negative sentiment would be likely to improve our system's performance, our results indicate that the system performs well even with imperfect clusters.

Related Work

Previous work in sentiment analysis has focused on the use of adjectives in text, for example in (Hatzivassiloglou

[1] Here, x is an adjective and y is a context word.

Adjective cluster 1	incredible, decent, mediocre, impressive, spectacular, exceptional, unbelievable, astonishing, remarkable, acceptable, extraordinary, promising, astounding, rounded, unparalleled, adequate, noteworthy, exquisite
Adjective cluster 2	easy, multiplayer, hard, cooperative, single-player, difficult, offline, casual, tricky, impossible
Context word cluster 1	feel, feeling, eye, tune , scary, mood, manner, creepy, tone, emotion, representation, ear, authentic, device, photo, tuned, disturbing, tuning, footstep, thick, spooky, vibe, eerie, depiction, horrific, dandy
Context word cluster 2	animation, movement, rate, frame, framerate, pacing, ran, greatness, stream, steady, silky, sailing, acceleration, warioware

Table 1: Examples of adjective and context-word co-clusters in 30×100 co-clustering

and McKeown 1997; Pang and Li 2008). In other work, co-occurrence of adjectives with nouns (e.g., "wonderful ideas", "horrible taste") has been used for the purpose of extracting more accurate or fine-grained opinions (but not for generating recommendations). For example, Stylios *et al.* (2011) extracted adjective-nouns pairs, which are in the dependency/modifying relation, from the opinions posted at an online forum on eGovernment for the purpose of mining the public opinions on various government decisions. In (Archak, Ghosea, an Ipeirotis 2007), a similar approach was used to identify consumer preferences from product reviews posted at Amazon.

As discussed earlier, Zagal and Tomuro (2010) also used to occurrence of adjectives in game reviews in order to cluster games using the vector space approach, based on frequency of occurrence of adjectives. They then qualitatively analyzed the clusters in order to assess the similarities of games that were clustered together. In our own previous work (Raison *et al.* 2012), we used co-clustering of adjective and context words in the same way as in our current work, but again the resulting vector space representation was used to cluster games, and then a qualitative analysis of the clusters was performed. Our current work is distinct in its application of co-clustering to the recommender task, and in the quantitative assessment of the accuracy of recommendations.

Co-clustering has also been used by others to capture the dependency between two variables (or objects and features), which are typically represented by the rows and columns of a matrix. For example, Archak, Ghosea, and Ipeirotis (2007) and Bisson and Hussain (2008) both applied co-clustering to the task of document categorization, and showed improved results as compared to the generation of clusters on a single dimension. George and Merugu (2005) applied co-clustering in generating product recommendations; however, this work did not involve the analysis of product reviews as in our own current work.

Feature Representations in Vectors

After co-clustering, our system replaces specific adjectives and context words in the raw frequency counts with the co-clusters that the words are members of. Using these co-cluster pairs, a game is represented as a vector, the length of which is the product of the number of adjective co-clusters times the number of context word co-clusters. For example, in 70×100 co-clustering (the maximum dimensionality that we used), the length of each vector is potentially 7,000[2]. Note that even with this relatively large number of co-clusters, the vector length is reduced by a factor of approximately 500 from the more than 3,500,000 possible combinations of single adjectives and single context words. Each number in the vector reflects the frequency of co-occurrence of a word that is a member of a particular adjective cluster with a word that is a member of a particular context word cluster.

We have experimented with a variety of approaches to weighing each feature in the vectors, as described below.

boolean: Values in a game's vector are either 1 or 0, reflecting whether or not a particular adjective/context-word co-cluster combination appears in any review of that game.

tf: Vector values are determined by counting the frequency of co-occurrence of each adjective-context word co-cluster.

tf-idf: This is the approach that is most commonly used in information retrieval systems (Salton, 1991). Vector values are calculated by multiplying term frequencies by the inverse document frequency (i.e., the inverse of the number of game reviews in which a member of an adjective cluster co-occurs with a member of a context word cluster).

tf-normc: This is a normalized version of *tf*, in which each term in the vector is normalized by dividing by the maximum value of that term over all games. In other words, this is a column-wise normalization of vectors, as opposed to a row-wise normalization.

max-tf: This builds on *tf-normc* by applying a smoothing function to the values in a vector, as described in (Manning *et al.* 2009). The smoothing function mitigates the effect of the length of the reviews on term frequencies. For each document d, let $tf_{max}(d) = max_{\tau \in d} tf_{\tau,d}$, where τ ranges over all terms in d. Then

$$max\text{-}tf(G_i) = \alpha + [(1\text{-}\alpha) * \frac{G_i}{max(G_i)}]$$

where G_i is the vector representation of game i, and α is a variable which ranges between 0 and 1. The result of the smoothing function is that each term weight is between α and 1. Typically $\alpha = \frac{1}{2}$.

[2]Features whose values are 0 across all vectors are removed, thereby shortening the vector lengths.

In each approach described above, similarity between two games is measured by computing the cosine of the angle between the vectors which represent the games. This similarity metric is typically used in IR systems (Salton 1989), and seems appropriate in our task because our representation of games is formed from free-form text reviews (i.e., documents).

Generation of Recommendations

Once the system has constructed the vector space representations of the games, it generates recommendations for a particular user as follows. Of the set of games that the user has reviewed (call this set G), the system identifies those games that the user likes (L). Gamespot reviews include rankings of games on a scale of 1 to 10, which our system uses (rather than the content of the free-form text reviews) to determine which games are in L. We chose the cut-off between liked and disliked games to be the median ranking given by a user to all games that s/he reviewed.

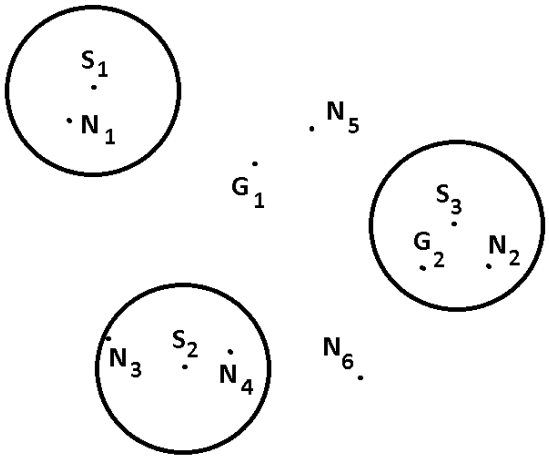

Figure 1: Illustration of the Recommendation Process

Results

We conducted three experiments in order to evaluate our system's performance. Because we were not in a position to perform live user tests, we conducted *offline* experiments (Herlocker *et al.* 2004). In the experiments, our system selected a small number of "seeds" (S) from the set of games L that a user liked,[3] and generated recommendations as described in the previous section. Under normal circumstances, these recommendations would not include games in G (i.e., games which the user has already reviewed). However, in our experiments, recommendations were limited *only* to games in G. In other words, the experimental task was: given S, can our system accurately predict which other games are in L and which are not?

[3]In our experiments, $|S| = 3$.

We evaluated the quality of our system's recommendations by measuring their precision, in the standard information retrieval meaning of this term (Salton 1989). In our context, precision is defined as follows:

$$\text{Precision} = \frac{|R \cap L|}{|R|}$$

High precision is achieved if a high percentage of games in R are games in L, although many games in L may not be included in R. Precision tends to vary inversely with n.[4]

For the experiments, we selected 10 users who had written the most reviews in our collection. We selected prolific reviewers so that we had ample data about which games those reviewers liked and disliked. On average the 10 users wrote reviews of 150-200 different games. For each user, we performed a standard k-fold cross-validation (Kohavi 1995). That is, we divided the set of games that the user liked (L) into k equal size folds. In each simulation a fold served as the seeds (S); therefore $k = |L|/|S|$. We measured average precision across the folds, and then calculated the average across the 10 users. [5]

In our first experiment, we varied the weighting scheme for feature dimensions in our vector space representations of games, using 10×100 co-clustering. Table 2 shows average precision over values of n ranging between 1 and 10. Despite the frequent use of *tf-idf* in many information retrieval systems, we found that *tf-normc* yielded superior performance (although *tf-idf* was second). The Boolean approach was third, and would have the advantage that vector feature values would not need to be recalculated as additional game reviews were added to the collection. Perhaps not surprisingly, *tf*, which is the only metric in which no normalization of any kind was applied, yielded the worst precision.

Metric	*Precision*
tf-normc	.86
tf-idf	.80
boolean	.74
max-tf	.70
tf	.66

Table 2: Average precision for 1-10 recommendations using various similarity metrics

Next, using *tf-normc*, we varied the dimensions used in co-clustering, ranging from 10×30 (10 adjective clusters and 30 context word clusters) to 70×100, in order to see the effects on system performance. Figure 2 shows precision vs. n for 10 adjective clusters, varying the number of context word clusters between 30 and 100. Similarly, Figure 3 shows precision vs. n for 30 adjective clusters. In general, co-clustering with 70 adjective clusters (not shown) produced worse performance. The fact that co-clustering with 10 adjective clusters produced the highest precision suggests that the vectors produced from larger numbers of adjective

[4]In all experiments, we tested for values of n between 1 and 10.

[5]In machine learning parlance, S is the "training set" and *L-S* is the "test set" for each fold.

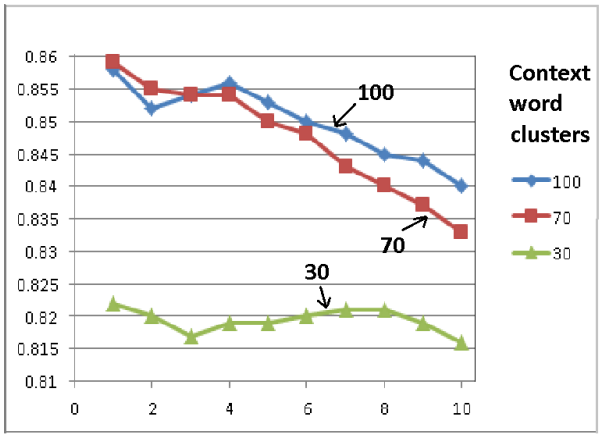

Figure 2: Precision vs. n with 10 adjective clusters

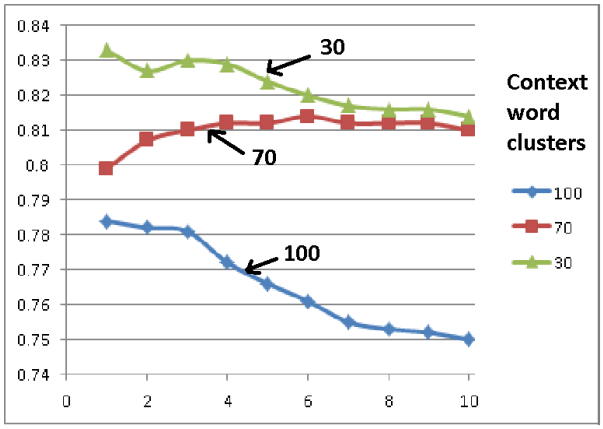

Figure 3: Precision vs. n with 30 adjective clusters

clusters are too sparse, and therefore fail to capture important commonalities among adjectives.

Finally, in the third experiment, we compared our use of word pair co-occurrence as features used to represent games with a simpler "bag of words" approach. In this baseline approach, the frequency of occurrence of single words, rather than word pairs, was used to build the vector representations of games. This seems to be an appropriate baseline with which to compare our system because of the prevalence of "bag of words" in information retrieval systems. Figure 4 shows a comparison of performance using 10×100 adjective-context word co-clustering (which achieved the highest precision), the average precision for co-clustering across all dimensions, and the bag-of-words approach. Co-clustering with 10 adjective clusters results in precision of 7-10 percentage points higher than bag-of-words, with the average precision for co-clustering using all dimensions higher than bag-of-words as well.

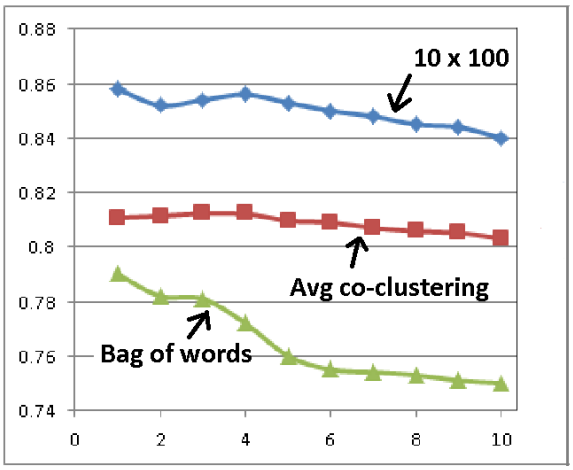

Figure 4: Precision vs. n, comparing co-clustering and bag-of-words

Conclusion

The results of our experiments suggest that our game recommender system produces high quality recommendations to users, based on a gaming community's reviews of games and a particular user's ratings of games s/he has played. As our experiments indicate, superior performance is achieved with the use of adjective-context word pairs (and then shrinking the dimensionality of the vectors using co-clustering) as opposed to the use of single terms in bag-of-words. Our system's performance is best when using small number (10) of adjective clusters. The optimal co-clustering dimensionalities (10×100) reduces the size of the vector space by a factor of about 300 when compared to no co-clustering of word pairs, and a factor of 5-10 compared to bag-of-words. The latter comparison indicates that the superior performance using co-clustering is achieved despite the reduction in dimensionality.

Using 10×100 or co-clusters, and when generating a small number of recommendations (between 1 and 10), we found that the co-clustering approach produces precision of .84-.86, while the precision of the bag-of-words approach is between .75 and .79.

In future work, we plan to apply our approach to produce recommendations in domains other than gaming. In principle our approach is applicable to any domain in which reviews (preferably detailed reviews) of products or services are available, such as restaurants, hotels, and so on.

User reviews reveal information about the content of games, and thus our approach is similar to other *content-based* recommendation systems (Pazzani and Billsus 2007). As stated earlier, including sentiment analysis in our approach would likely yield better results. We also plan to compare our approach to *collaborative filtering* techniques (Ricci, Rokach, and Shapira 2011).

References

Archak, N. Ghosea, A., and Ipeirotis, P. 2007. Show Me the Money! Deriving the Pricing Power of Product Features by Mining Consumer Reviews. In Proceedings of the Thirteenth ACM International Conference on Knowledge Discovery and Data Mining, 56-65. New York, NY: ACM.

Dhillon, I.S., Mallela, S., and Modha, D.S. 2003. Information-theoretic Co-clustering. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 89-98. New York, NY: ACM.

George, T. and Merugu, S. 2005. A Scalable Collaborative Filtering Framework Based on Co-clustering. In Proceedings of the Fifth IEEE International Conference on Data Mining, 625-628. Los Angeles, CA: IEEE Computer Society.

Hatzivassiloglou, V., and McKeown, V. 1997. Predicting the Semantic Orientation of Adjectives. In Proceedings of the Eighth Conference of the European Chapter of the Association for Computational Linguistics, 174-181. Stroudsburg, PA: Association for Computational Linguistics.

Kohavi, R. 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1137-1145.

Pang, B. and Lee, L. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* 2(1-2), 1-135.

Pazzani, M. J., Billsus, D. (2007). Content-based Recommendation Systems. In *The adaptive Web* (pp. 325-341). Springer Berlin Heidelberg.

Raison, K., Tomuro, N., Lytinen, S., and Zagal, J. 2012. Extraction of User Opinions by Adjective-context Co-clustering for Game Review Texts. In Proceedings of the Eighth International Conference on NLP, Kanazawa, Japan, 289-299. Berlin: Springer-Verlag.

Ricci, F., Rokach, L., and Shapira, B. 2011. *Introduction to Recommender Systems Handbook.* Springer U.S.

Salton, G. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Boston, MA: Addison-Wesley.

Salton, G. 1991. Developments in automatic text retrieval. *Science* 253, 974-979.

Salton, G., Wong, A,. and Yang, C. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM* 18(11): 613-620.

Stylios, G., Christodoulakis, D., Besharat, J., Kotrotsos, I., Koumpouri,A., and Stamou, S. 2011. Public opinion mining for governmental decisions. *Electronic Journal of Electronic Government* 8(2), 202-213.

Zagal, J., and Tomuro, N. 2010. The Aesthetics of Gameplay: A Lexical Approach. In Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments, 9-16. New York, NY: ACM.