Pre-release Box-Office Success Prediction for Motion Pictures

Conference Paper · July 2013

DOI: 10.1007/978-3-642-39712-7_44

2 authors:

Rohit Parimi
Bloomberg LP
11 PUBLICATIONS **51** CITATIONS

SEE PROFILE

Doina Caragea
Kansas State University
161 PUBLICATIONS **2,079** CITATIONS

SEE PROFILE

Pre-release Box-Office Success Prediction for Motion Pictures

Rohit Parimi and Doina Caragea

Computing and Information Sciences,
Kansas State University,
Manhattan, KS, USA 66506
{rohitp,dcaragea}@ksu.edu
http://www.cis.ksu.edu

Abstract. In the recent past, machine learning algorithms have been used effectively to identify interesting patterns from volumes of data, and aid the decision making process in business environments. In this paper, we aim to use the power of such algorithms to predict the pre-release box-office success of motion pictures. The problem of forecasting the box-office collection for a movie is reduced to the problem of classifying the movie into one of several categories based on its revenue. We propose a novel approach to constructing and using a graph network between movies, thus alleviating the movie independence assumption that traditional learning algorithms make. Specifically, the movie network is first used with a transductive algorithm to construct features for classification. Subsequently, a classifier is learned and used to classify new movies with respect to their predicted box-office collection. Experimental results show that the proposed approach improves the classification accuracy as compared to a fully independent setting.

Keywords: Transductive Approach, Classification, Motion Pictures, Business Intelligence, Features.

1 Introduction

Movies have become an integral part of our lives as a means of relaxation and entertainment. Movies have also been a significant medium for culture exchange between different countries and regions and are thus an indispensable asset to the world. Given this, the movie industry has become a business and it has huge market profit and potential [9]. As a consequence, the knowledge and research about the movie industry is becoming deeper. Ability to accurately predict the box-office returns for a movie will help the cinema line determine the propaganda cost and period of showing the movie to maximize the profit.

The problem of predicting the box-office gross of a pre-release motion picture has been widely tackled in the past from a statistical point of view. There are many factors influencing the box-office of a movie, for example, number of screens for the movie, advertising, time of the year, number of movies that are released

P. Perner (Ed.): MLDM 2013, LNAI 7988, pp. 571–585, 2013.

and so on, making the problem challenging. Some of the prior work on this problem was aimed at identifying features that influence the outcome of a movie and finding if they are positively or negatively correlated to the outcome [3].

With the success of machine learning algorithms in improving managerial decision making, researches have started using techniques to build predictive models when addressing the problem of predicting the box-office gross of a movie. Sharda et al. [1] have reviewed the past research on this problem and re-introduced the problem from a machine learning perspective. In their work, Sharda et al. [1] addressed the gross prediction problem by converting it to a classification problem and building a predictive model based on artificial neural networks. They analyzed 7 different features that influence a movie's gross and used them as inputs to a multilayer perceptron neural network. The output from the model is one of 9 classes (selected based on outcome range) to which the movie might belong. In their approach, the authors make the assumption that each movie is independent of the other movies - a basic assumption made by traditional classification algorithms.

However, in the case of movie gross prediction problem, movies are generally not independent. In fact, there is an underlying graph structure that we could identify among movies. For example, a movie can be connected to another movie if they share actors and/or directors, if they have the same genre, if one is a sequel to the other, or if they are released around the same time. If we consider common actors or directors, the intuition is that the reputation of an actor or a director who worked in a movie can be transferred to a different movie in which the actor or the director took part. Thus, we believe that the reputation of *Steven Spielberg* as the director of a yet to be released movie will have positive effect on the success of that movie compared to the success of a yet to be released movie directed by a rookie director.

Traditional classification models assume data instances are independent and identically distributed (i.i.d.) and fail to capture dependencies among instances, in our case movies. To address this limitation, there has been some prior work in the area of link-based classification. Getoor et al. [10] emphasized the importance of link information for classification and proposed a framework to model link distributions. Neville et al. [11] presented a relational Bayesian classifier with different estimation techniques to learn from linked data. Parimi et al. [12] addressed the link prediction problem in *LiveJournal* social network by combining link information with user interest features. Zhu et al. [5] proposed a matrix factorization technique to capture the structure of the graph for web-page classification. The success of the prior work in using link information for classification motivated us to construct a movie dependency network when addressing the gross prediction problem. We use the matrix factorization approach proposed by Zhu et al. [5] to generate network-based features for classification.

The main contributions of our work are as follows: a) an approach to create a graph network that captures dependency relations among movies; b) a custom weighting scheme to compute the weights on the edges; c) generation of features from the network; d) experimental results on a movie dataset showing that the

best performance is obtained when movie independent features are combined with dependency features extracted from the network.

The rest of the paper is organized as follows: Section 2 describes prior work in the areas of movie gross prediction and link-based classification. We provide the details of the data used in this work and its categorization in Section 3. Section 4 presents the details of our methodology, specifically the relational setting, the baseline approach and the experimental design. In Section 6, we explain the results of the experiments. Finally, Section 7 discusses the overall contribution of this study and future research directions.

2 Related Work

Much better marketing strategies can be designed and better choices can be made by the cinema production companies in the presence of a strong estimator of a movie's anticipated success. With this in mind, researchers in the past have tried to identify factors that influence the success of a movie and computed correlations between those variables and a movie's box-office gross. Moon et al. [3] used the information from the entire lifetime of a movie to improve their gross predictions over time. One factor that they considered was the word-of-mouth, as it can be indicative of the demand associated with a movie. In addition, they identified correlations between critic ratings and advertisement, and the movie revenue. Their results show that the opening weekend collections for a movie are the strongest estimators for the lifetime gross of the movie. However, it is worth noting that the problem of predicting the gross before a movie's release, that we address, is harder than extrapolating the gross based on first week collections.

Sharda et al. [1] approached the movie gross prediction problem from a machine learning perspective. They reviewed past research on the variables influencing the success of a movie and used seven of those in their work. They converted the problem of predicting the revenue of a movie into the problem of classifying a movie based on revenue ranges. For example, a movie can be considered a *flop* if its revenue is in the range *100,000 to 200,000* and a *blockbuster* if its revenue is in the range *5 million to 10 million*. A multilayer perceptron network is used to classify a movie's gross in one of 9 possible classes. However, a drawback of this approach is that it assumes movies to be independent from each other.

The work by Zhang et al. [4] addressed the box-office prediction problem using a multilayer back-propagation neural network. Similar to the work by Sharda et al. [1], Zhang et al. [4] identified 11 input variables to the neural network based on market survey. The weights for the neural network model are selected using statistical methods to maximize the accuracy of the model. Even though the accuracy of the proposed model is better than the accuracy of the model proposed by Sharda et al. [1], the dataset used in this work is rather small, consisting of only 241 movies classified according to 6 classes.

Recent research in link-based classification has focused on ways in which inherent dependency relations between instances can be used to improve results

of traditional learning algorithms, which assume instances to be independent. Many techniques have been proposed in the past to exploit the graph structure of particular problems. Getoor et al. [10] have proposed a framework to capture correlations among links using link distributions. They used an iterative classification algorithm, which combines both link information and information intrinsic to instances (e.g., web-page content) for classification. The authors have applied this approach to web and citation collections and reported that using link distribution improved accuracy in both cases. The work by Parimi et al. [12] combined content features, obtained by modeling user interests using LDA, with graph features to predict friendship links in *LiveJournal* social network. Zhou et al. [8] proposed a framework that uses the adjacency matrix of the graph for propagating labels in the graph. However, this technique like other techniques outlined above, cannot be used to propagate labels in a directed graph.

Zhou et al. [6], [7] have studied techniques, which aim at exploiting the structure of the graph globally rather than locally, thus ensuring that the classification is consistent across the whole graph. The approach proposed in [6] is motivated by the framework of *hubs* and *authorities*, while the work proposed in [7] is inspired by the *PageRank* algorithm used by Google search engine. Although the techniques proposed in [6] and [7] are applicable for classification in directed graphs, they rely solely on link structure and ignore content information (e.g., content of a web-page). To address this limitation, Zhu et al. [5] proposed an algorithm to jointly factorize the content information and link information (represented as weighted edges in a directed graph) in a supervised setting, and showed that this joint factorization improves the classification accuracy compared to just using link information.

We plan to take advantage of the approach proposed in [5] to factorize the link information from the movie graph that we construct, and hypothesize that the factors obtained as a result of this model capture the similarity between movies better than measures that use only the content. In addition, the dimensionality of the problem is reduced when representing each instances by its factors.

3 Data

The dataset that we used in this work consists of 977 movies released as 'wide releases' between the years 2006 and 2011. There are approximately 150 movies in each year and for each movie we collected features such as actor and director profiles (to compute star value), genre, release date, sequel information, budget, runtime, number of theaters and MPAA rating. All these features, from movie information to actor and director profiles are collected from a well-known site: Box Office Mojo. We selected these features based on the work by Sharda et al. [1], and grouped them into two categories based on how we use them in our proposed approach. Specifically, we use the budget, runtime, number of theaters and MPAA rating as features intrinsic to a movie (i.e., content features), while actors, directors, genre, release date and sequel information features are used to construct dependency relations between movies, or more precisely weighted edges in the movie graph (i.e., link features).

Intuitively, the *budget, runtime, number of theaters* and *MPAA rating* features represent information particular to a movie and directly contribute to the revenue of the movie, independent of other movies. These features can be seen as content features and are directly used when training the model (without additional processing or transformation). As opposed to these features, link features capture dependencies between features and are used to form a directed graph (precisely, weighted edges) between movies. They influence the movie gross indirectly, by the means of graph features extracted from the graph. More intuition behind using the dependency features is provided in what follow.

The *actors* and *directors* of a movie, in general, have a popularity index associated with them. The popularity that an actor or a director achieves through the success of their movies creates a positive sentiment for their up-coming movies, affecting the revenue of that movie. This is precisely what we want to capture when using actors and directors to construct dependencies between movies, i.e. weighted edges in the movie graph. Other dependency features like *genre* and *sequel* are meant to capture the percentage of audience that are loyal to that genre or franchise, respectively, when used in the graph network. For example, a person who is mostly inclined towards movies with genre 'thriller', would most likely watch the awaited thriller movies rather than dramas or comedies. Similarly, a person who likes movies like *Sherlock Homles-1* and *Sherlock Holmes-2* would most likely watch the upcoming movie *Sherlock Holmes-3*. The feature *release date* captures the competition that a movie might have to face when it is released. It is said by the industry experts that revenue, in general, is likely to be divided among the movies released at the same time. We should note that the feature *MPAA rating* can also be categorized as a dependency feature, but the presence of only 4 different kinds of ratings (G, PG, PG-13, R) will result in too many links in the graph network. Also, we believe that, unlike genre, MPAA rating will not be useful in capturing the interest of audience towards a particular rating. Hence, it is just used as a movie content feature.

4 Approach

As mentioned above, our objective is to design an approach that can take advantage of link features (capturing dependencies between movies), in addition to content features, for the task of predicting box-office revenues of movies before their theatrical release. In this section, we will describe in detail the construction of a dependency graph between movies, several weighting schemes used for computing weights on the edges of the graph and an algorithm to capture the structure of the graph. We will also present the baseline model against which our approach will be compared.

4.1 Relational Setting

Graph Construction and Weighting Schemes: In Section 3, we categorized features as link features (constructed based on the dependency graph) and

independent features. Here, we explain how the movie graph is constructed using link features. Two movies in the dataset are connected if they:

1. have a common actor
2. have a common director
3. have the same genre
4. are sequels
5. are released around the same time (e.g., two weeks apart)

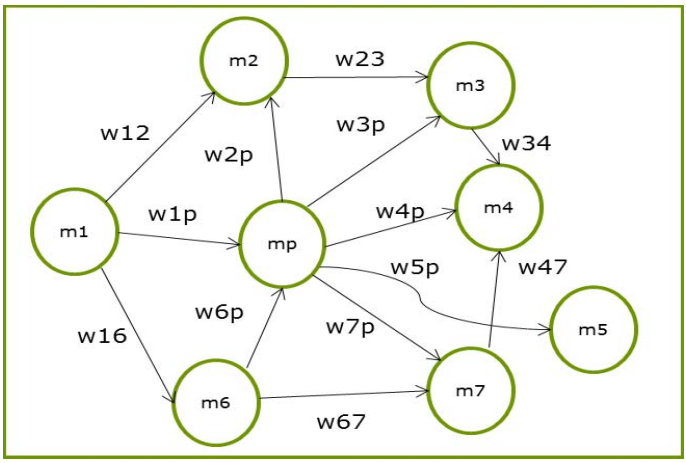

Fig. 1. Example of a movie graph created using dependency features

Figure 1 depicts an example movie graph that can be constructed using the dependency features. As seen in Figure 1, each link has a weight term which can be seen as the similarity between the two connected movies. In this work, we experiment with three different weighting schemes to see which one represents our data better. We only use dependency features in weighting schemes 2 and 3. The three schemes are described as follows:

1. **A Constant Weight '1'**: In this weighting scheme each link in the graph is assigned a constant weight value '1' without considering how similar the nodes connected by the edge are. This is a simple weighting scheme but has the disadvantage of not capturing the similarity between the nodes.
2. **Radial Basis Function (RBF) Kernel**: The weight corresponding to an edge in the graph is given by the following kernel function:

$$k(x, x') = exp(-\|x - x'\|^2 / 2\sigma^2) \quad (1)$$

where x and x' are the feature representations of the nodes connected by the edge and the features are computed in a way similar to those in the independent setting (Section 4.2). Equation 2 captures the similarity between the two nodes.

3. **Custom Weighting Scheme**: Even though weights from RBF kernel capture the similarity between the nodes, they cannot be used to capture the negative effect from the feature 'Competition' or the positive effect of 'Sequel'. We shall consider two examples to understand the disadvantage of using an RBF kernel. The headers '1', '2', '3' and '4' in Tables 1 and 2 correspond to the features: 'Star Value', 'Genre', 'Sequel' and 'Competition', respectively. In the example in Table 1, the two movies are clearly connected as they are sequels and hence are very similar to each other. However, we will have a weight value less than 1 for the edge connecting these two movies because of the difference in the values for feature 'sequel'. Consider the example in Table 2 and assume that the two movies are connected based on release date. The edge connecting these two movies will have a weight value of '1' if RBF kernel is used because of identical values for all the features. Even though the movies in Table 2 are less similar than those in Table 1, the weight values from RBF kernel suggest otherwise. Because of this disadvantage, we designed a custom weighting scheme for the movie domain.

   In this weighting scheme, the weight for a link between two movies is determined by linearly combining the dependency features with coefficients as shown in Equation 2. The intuition of weights on the graph in weighting schemes 1 and 2 is that they represent similarity between the two nodes connected by the edge. However, in this weighting scheme, the intuition of weight has changed from similarity between the nodes to the positive/negative effect that can be transferred from one node to other node. The feature *Release-Date* in Equation 2 takes a value '1' if the two movies are connected based on release date and '0' if not. Other features in Equation 2 are computed in a way similar to those in independent setting (Section 4.2). Optimal values for the coefficients can be obtained either by using a validation set or by trial and error based on domain knowledge. Because of small number of movies in our dataset, we determined the coefficients by trial and error.

Table 1. Example 1, depicting the disadvantage of RBF Kernel

Title	1	2	3	4
Batman Begins	High	Action	0	A
The Dark Knight	High	Action	1	A

Table 2. Example 2, depicting the disadvantage of RBF Kernel

Title	1	2	3	4
IronMan II	High	Action	1	A
The Dark Knight	High	Action	1	A

$$w12 = a1 * (ActorValue) + a2 * (DirectorValue) + a3 * (Genre) + a4 * (Sequel) - a5 * (ReleaseDate) \quad (2)$$

Classification in Directed Graphs: Given a network graph with nodes and weights on edges, the objective is to find the class label for the nodes which do not have any class label. For example, in Figure 1, if we know the revenues for

the movies *m1* through *m7*, we would want to find out the revenue for the movie *mp*. As explained in Section 2, there has been some work in the past to solve problems that match the above criterion. In this work, we use the algorithm proposed by Zhu et al. [5] because of its ability to combine content information into the model, and at the same time, reduce the dimensionality of the problem. This algorithm falls into the category of transductive algorithms in machine learning. Transductive algorithms are algorithms which are trained on specific training instances to reason on specific test instances. Suppose we add a new node to the graph, we would have to run the algorithm again to predict the revenue for the newly added node. The algorithm that we use in this work uses the matrix factorization technique to factorize the adjacency matrix in a graph network. The factors generated as a result of this factorization can be seen as link features and can be further used as features for classification. The factorization is given by the following equation:

$$\min_{Z,U} \left\| A - ZUZ^\top \right\|_F^2 + \gamma \left\| U \right\|_F^2 \tag{3}$$

where,

- 'A' is the adjacency matrix represented by weights
- 'Z' is an $n \times l$ feature matrix, n = #instances, l = #features
- 'U' is an $l \times l$ matrix
- $\|.\|_F$ is the Frobenius norm.

Equation 3 can be solved using optimization techniques such as *gradient descent*. The intuition behind Equation 3 is to approximate the 'A' matrix using the product $ZUZ^\top$ and as a result, we obtain link features in the 'Z' matrix which can be used to represent each instance (movie in our case).

The pseudo-code to obtain features using the above technique is as follows:

1. Construct the graph using the link features.
2. Use one of the weighting schemes to fill the adjacency matrix with weights.
3. Run the above matrix factorization algorithm to get 'Z' matrix.
4. Use the 'Z' matrix to represent each movie for classification.

Movie Representation: As explained earlier, each movie in the dataset is represented using the features obtained by factorizing the adjacency matrix. These features are henceforth referred to as link or graph features. The number of link features used to represent a movie (i.e., the number of factors) is decided using a validation set. Other features such as movie independent features can also be appended to the link features. Once all the movies in the dataset are represented either using just link features or link + movie independent features, we build predictive models to test our hypotheses. An example representation for the movie 'The Dark Knight' is shown in Equation 4. The movie independent features are appended to the graph features in this example.

$$TheDarkKnight = f_1 \cdots f_l, Budget, \#Theaters, Time, MPAA, Class \tag{4}$$

4.2 Baseline: Independent Setting

Many researchers in the past have assumed that movies are independent when addressing the gross prediction problem. In this setting, we use this independence assumption between the movies, construct features and build predictive models to classify movies into gross ranges. This approach is identical to the one described in Sharda et al. [1] and thus serves as a baseline for our approach.

Feature Construction: The features that are used to build predictive models in the independent setting are the following:

1. **Star Value**: The star value for a movie is contributed by the actors and directors of that movie. We collected information about top 5 actors and all the directors for a movie and crawled the actor/director profile from the site Box Office Mojo. The value an actor or a director contributes is determined by averaging the gross for all the movies (released) the actor or director took part in. The overall value from the actors/directors for a movie is the average of values for the actors/directors. The star value for a movie is obtained by taking a weighted average of the values contributed by all the actors and the directors (depicted in Equation 5). The coefficients are set as 0.7 for actor value and 0.3 for director value based on past research which indicates that the director value for a movie is not as significant as the actor value for that movie. We used three independent binary variables to represent the degree of star value in our model: 'High', 'Medium', and 'Low' values, by discretizing the star value computed. Based on our calculations, a movie is assigned a star value 'High' if the value from Equation 5 is greater than 65 million, 'Medium' if the value is between 25 million and 65 million and 'Low' if the value is less than 25 million.

$$StarValue(m) = 0.7 * ((a1 + a2 + a3)/3) + 0.3 * ((d1 + d2)/2) \quad (5)$$

2. **Sequel**: Similar to other prior studies, we used a binary variable to determine whether a movie is a sequel or not. Our intuition is that the sequels are positively correlated with the success of a movie as they are filmed, because of the success of the previous versions of that movie. The feature takes the value '1' if the movie is a sequel and '0' if the movie is not.
3. **Competition**: We used this feature to capture the competition that a movie faces from other movies that are released around the same time. In a study by Moon et al. [3], it is reported that the pool of entertainment dollars is shared between the movies that are released around the same time. Many studies in the past have found release date to be an important contributor to a movie's box-office success and it can be used to capture the level of competition. The feature competition is expected to negatively influence the success of a movie. We represented competition using the following values: 'High', 'Medium' and 'Low'. A value 'High' indicates high competition, a value 'Medium' indicates medium and a value 'Low' indicates low competition for a movie. Based on

the release dates of the movies in our dataset, we assign 'High' to the movies released in 'January', 'August', 'September' or 'October'; 'Medium' to the movies released in 'February', 'March', 'April', 'November', or 'December' and 'Low' to the movies released in 'May', 'June' or 'July'.

4. **Genre**: Most of the past work has identified genre as a content category determiner and used it in their work even though it is rarely found to be significant. We followed the convention and used it as a feature. Eighteen different genres are used to tag a movie and we allowed a movie to be tagged with more than one genre value. The tags we used are shown in Table 3.
5. **Budget**: In the recent past, budget for a movie seems to be one of the features highlighted in the promotions for a movie with the objective of attracting more people to watch the movie. We believe that this feature contributes to the success of a movie and is positively correlated with it. We used a positive integer to represent this feature.
6. **Run Time**: This feature is represented as a continuous variable and captures the length of the movie.
7. **Number of Theaters**: Previous work for solving this problem showed correlations between a movies' financial success and the number of screens it is released in. We represent this feature as a continuous variable indicating the number of screens a movie is scheduled to be shown at its opening.
8. **MPAA Rating**: A commonly used variable in predicting the gross for a movie, which takes the values 'G', 'PG', 'PG-13' and 'R'.

Table 3 summarizes all the features used in the independent setting and the possible values that each feature may take. An example of how the movie 'The Dark Knight' is represented in the independent setting is shown in Table 4. Table 5 depicts the gross ranges for each of the classes used in this work. This is the same for both independent setting and dependent setting.

Table 3. Summary of features and the values they take in the independent setting

Competition	StarValue	Sequel	Genre	RunTime	Budget	Theaters	MPAA
High Medium Low	High Medium Low	1 0	Period, Crime, Action, Romance, Thriller, Family, Historical, Sci-Fi, Horror, Drama, Comedy, Sports Fantasy, Music, War, Animation Documentary Adventure	RunTime for the movie in minutes	Budget for the movie in dollars	Positive integer	G PG PG-13 R

Table 4. Representation of **'The Dark Knight'** in independent setting

Competition	StarValue	Sequel	Genre	RunTime	Budget	# Theaters	MPAA	Class
C	High	1	Action	150	185000000	4366	PG-13	9

Table 5. Discretization of the movie gross into 9 classes

Class No	1	2	3	4	5	6	7	8	9
Range (in Millions)	< 10 (flop)	> 10 < 20	> 20 < 30	> 30 < 45	> 45 < 70	> 70 < 100	> 100 < 150	> 150 < 225	> 200 (blockbuster)

5 Experimental Design

As explained in Section 3, the dataset that we used in our experiments consists of 977 movies released as 'wide releases' between the years 2006 and 2011. Movies released between the years 2006 and 2010 are considered as training instances and movies released in the year 2011 are considered as test instances. This will ensure that we use information from the past to predict the gross for future movies. Features are constructed for the movies in the training set and test set for the relational setting and independent setting, as described in Section 4.1 and Section 4.2, respectively. Model parameters and number of graph features, in case of the relational setting are tuned using a validation set constructed using movies between the years 2006 and 2010.

5.1 Research Questions

Our experiments have been designed to address two main research questions:

- Which weighting schemes is better in terms of prediction accuracy?
- How does the relational setting compare with the independent setting?

5.2 Experiments

To answer the above questions, we have designed the following experiments:

1. **Experiment 1**: In this experiment, we test the performance of predictive models trained on features constructed using the independent setting (Section 4.2). This experiment will be referred to as exp_1 henceforth.
2. **Experiment 2**: We ran two variants of this experiment in which we test predictive models trained on graph features constructed using weighting scheme 1 (Section 4.1). In the first variant which will be called exp_2_0 henceforth, we use just the graph features to build the models. In the second variant which will be called exp_2_1, we add the movie independent features (*Budget, MPAA Rating, No. Theaters* and *Runtime*) to the features used in exp_2_0.

3. **Experiment 3**: In this experiment, we build models using graph features constructed using weighting scheme 2. Similar to experiment 2, we have exp_3_0 and exp_3_1.
4. **Experiment 4**: The graph features in this experiment are constructed using the weighting scheme 3. Similar to the above two experiments, we have exp_4_0 which uses just graph features and exp_4_1 which uses graph features along with movie independent features. The values for the coefficients in Equation 2 are: a1=0.55, a2=0.25, a3=0.2, a4=1, a5=1.

For all the experiments, we used Weka implementations of the Logistic Regression and Random Forest algorithms.

5.3 Evaluation Criteria:

To evaluate the results of the experiments outlined above, we have used the following metrics:

1. **Bingo Accuracy**: Also known as BINGO or simply accuracy, it is defined as the ratio between the number of instances correctly classified and the total number of instances.
2. **AUC**: Area under the ROC curve, or AUC, is one of the popular metrics used to evaluate prediction models. For each experiment, we report the weighted average AUC value, as output by Weka.

6 Results and Discussion

6.1 Comparison between Different Weighting Schemes

As mentioned earlier, experiments have been conducted to test which of the weighting schemes better capture the information from the features used in the dependency setting. As expected, weights generated using the custom weighting scheme produced better results for the evaluation metrics in most of the cases. This can be seen from Tables 6 and 7. Custom weighting scheme has better AUC and accuracy values compared to the AUC and accuracy values from the other two weighting schemes for the random forest classifier when using just the graph features (Table 6) or when the movie independent features are appended to the graph features (Table 7). For the logistic regression classifier, both AUC and accuracy values are better compared to the AUC and accuracy values from the other two weighting schemes when we append the graph features to the movie independent features. Hence, it is evident from Tables 6 and 7 that the classification accuracies of the predictive models built using the custom weighting scheme are better than those built using a constant weighting scheme or weights generated using an RBF kernel.

Table 6. AUC and accuracy values for logistic regression and random forest classifiers trained on graph features constructed using different weighting schemes. Best AUC and accuracy values, across all the experiments, for each classifier are highlighted.

Exp	Metrics	Logistic Regression	Random Forest
exp_2_0	AUC	0.538	0.536
	ACC	15.07	15.75
exp_3_0	AUC	**0.596**	0.559
	ACC	21.23	16.44
exp_4_0	AUC	0.585	**0.598**
	ACC	**24.66**	**21.92**

Table 7. Similar to Table 6, AUC and accuracy values for classifiers trained on movie independent features appended to the graph features. Best AUC and accuracy value across all the experiments, for each classifier are highlighted.

Exp	Metrics	Logistic Regression	Random Forest
exp_2_1	AUC	0.702	0.667
	ACC	26.03	20.55
exp_3_1	AUC	0.673	0.729
	ACC	21.92	27.4
exp_4_1	AUC	**0.735**	**0.732**
	ACC	**33.56**	**32.88**

6.2 Comparison between Independent and Relational Settings

Table 8 depicts the comparison between the results for the independent setting and the dependent setting for logistic regression and random forest classifiers. The proposed approach of using a dependency relation between the movies was able to improve the accuracy of the predictive models and is very close in-terms of the AUC metric compared to the independent setting. The reason for a lower AUC value in relational setting for the predictive models compared to those in independent setting might be because we have optimized the accuracy metric during the validation experiments to get the number of graph features to use and to tune the model parameters, ridge in case of logistic regression and number of trees for the random forest classifier.

Table 8. AUC and accuracy values for logistic regression and random forest classifiers in the independent and relational settings, using the three weighting schemes. Best AUC and accuracy values across all the experiments, for each classifier, are highlighted.

Exp	Metrics	Logistic Regression	Random Forest
exp_1	AUC	**0.748**	**0.745**
	ACC	29.45	26.71
exp_2_1	AUC	0.702	0.667
	ACC	26.03	20.55
exp_3_1	AUC	0.673	0.729
	ACC	21.92	27.4
exp_4_1	AUC	0.735	0.732
	ACC	**33.56**	**32.88**

As expected, better results for the accuracy are achieved using the custom weighting scheme and we hypothesize that tuning the coefficients a1, a2, a3, a4 and a5 from Equation 2 might improve the results further. It is also evident that the results shown in Table 8 are much better that the results from a random classifier which will have an accuracy of 11.11% (1/9).

7 Conclusions and Future Work

We analyzed the problem of predicting the revenue of a movie before its theatrical release and identified several factors influencing the same. We designed an approach to construct a dependency network for the movies in the dataset and worked on the application of a transductive algorithm to predict the missing labels for the nodes in the graph. We have designed three different weighting schemes to represent the network using the adjacency matrix and experiments are conducted using all the weighting schemes to determine which is effective. It is evident from Tables 6 and 7 that the custom way of generating weights produced better results compared to the other two weighting schemes. Our hypothesis that considering a dependency relation between movies helps improve the prediction accuracy is confirmed by the results in Table 8. The results also show that the AUC and accuracy values of the predictive models are improved when the movie independent features are appended to the graph features.

As an extension to the work described in this paper, we would like to study the influence of social media on a movie's success. Our intention is to capture the word-of-mouth effect or the demand for the movie using social media. There has been some work in the recent past along the lines of using social media data to predict movie ratings and box-office gains. Asur et al. [13] used 'Twitter' social media to collect tweets related to movies and analyzed their influence on a movie's revenue. They concluded that the rate of tweets on a movie has a positive influence on the movie's success. Moreover, sentiments extracted from tweets further improved the predictions. Wong et.al. [14] also used 'Twitter' data to predict the rating as well as box-office gains for a movie, by doing sentiment analysis on the tweets. Encouraged by the results published in [13] and [14], we plan to test how informative will the data from 'Twitter' be, for the movies in our dataset. To accomplish this, we developed a framework to query and retrieve tweets about movies from a popular search engine for *Twitter* called Topsy. The number of unique users and total tweets can be used to capture the percentage of audience interested in the gossip about the movie and the sentiment of a tweet can be used to know if a user likes or dislikes the movie. We hypothesize that the features constructed from the social media data will further improve the classification accuracy and be useful in answering the question: 'How predictive are the features from different data sources?'

Acknowledgements. We would like to thank Dr. Shenghuo Zhu for sharing the code of his algorithm.

References

1. Sharda, R., Delen, D.: Predicting box-office success of motion pictures with neural networks. Expert Systems with Applications 30, 243–254 (2006)
2. Zhang, L., Luo, J., Yang, S.: Forecasting box office revenue of movies with BP neural network. Expert Systems with Applications 36, 6580–6587 (2009)
3. Moon, S., Bergey, K.P., Lacobucci, D.: Dynamic effects among movie ratings, movie revenues, and viewer satisfaction. American Marketing Association (2010)
4. Zhang, W., Skiena, S.: Improving movie gross prediction through news analysis. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (2009)
5. Zhu, S., Yu, K., Chi, Y., Gong, Y.: Combining content and link for classification using matrix factorization. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2007)
6. Zhou, D., Schölkopf, B., Hofmann, T.: Semi-supervised learning on directed graphs. In: Proceedings of Neural Information Processing Systems (2005)
7. Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005 (2005)
8. Zhou, D., Bousquet, O., Navin, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Proceedings of Advances in Neural Information Processing Systems, vol. 16 (2004)
9. Shanklin, W.: What businesses can learn from the movies. Business Horizons 45(1), 23–28 (2002)
10. Geetor, L., Lu, Q.: Link-based Classification. In: Twelth International Conference on Machine Learning, ICML 2003, Washington DC (2003)
11. Neville, J., Jensen, D., Gallagher, B.: Simple Estimators for Relational Bayesian Classifiers. In: Proceedings of the Third IEEE International Conference on Data Mining, ICDM 2003 (2003)
12. Parimi, R., Caragea, D.: Predicting friendship links in social networks using a topic modeling approach. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) PAKDD 2011, Part II. LNCS, vol. 6635, pp. 75–86. Springer, Heidelberg (2011)
13. Asur, S., Huberman, B.A.: Predicting the future with social media. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (2010)
14. Wong, F.M.F., Sen, S., Chiang, M.: Why watching movie tweets won't tell the whole story? In: Proceedings of the 2012 ACM Workshop on Online Social Networks, WOSN 2012, pp. 61–66. ACM, New York (2012)