Assessing Box Office Performance Using Movie Scripts: A Kernel-Based Approach

Jehoshua Eliashberg, Sam K. Hui, and Z. John Zhang

Abstract—We develop a methodology to predict box office performance of a movie at the point of green-lighting, when only its script and estimated production budget are available. We extract three levels of textual features (genre and content, semantics, and bag-of-words) from scripts using screenwriting domain knowledge, human input, and natural language processing techniques. These textual variables define a distance metric across scripts, which is then used as an input for a kernel-based approach to assess box office performance. We show that our proposed methodology predicts box office revenues more accurately (29 percent lower mean squared error (MSE)) compared to benchmark methods.

Index Terms—Entertainment industry, green-lighting, movie production, text mining, kernel approach

1 INTRODUCTION

When deciding which scripts to turn into movies (i.e., "green-lighting"), movie studios and film makers need to assess the box performance of a movie based only on its script and allocated production budget [10]. Such assessments are extremely challenging, as most post-production drivers of box office performance (e.g., actor, actress, director, MPAA rating) are unknown at the point of green-lighting, when financial commitments have to be made. Usually, movie producers rely on a "comps"-based approach to assess the box office potential of a new script. Specifically, they identify around five to ten past movies that are "similar" to the focal script, and use the box office performance of those movies as benchmarks for the revenue potential of the focal script. While most movie experts believe that a movie's story line is highly predictive of its ultimate financial performance [3], [11], [12], [14], [27], it is unclear how "similarity" between movies scripts should be measured. For instance, should one focus on the overall theme, the actual words/language used, or the structure of the scenes and dialogues?

Our goal is to answer the above questions and in turn develop a decision aid that helps studios make green-lighting decisions. Towards that end, we develop a methodology, based on text mining and the kernel approach, that identifies the "comps" of a new script based on its content and textual features, and hence assesses its revenue potential. Guided by domain knowledge from screenwriting and techniques from natural language processing [24], we extract the following textual features from a script, ordered from "higher-level" concepts to "lower-level" features: (i) genre (e.g., action, thriller, comedy), (ii) content of the story line (e.g., whether there is a surprising ending), (iii) semantic features of the script (e.g., number of scenes, length of the dialogues), and (iv) the actual set of words that are used in the script (e.g., prevalence of vulgar language). We extract the low-level features (semantics features and the actual words used) by machine, while the high-level concepts (genre and content variables) are determined by human readers.

We then define a "distance metric" between scripts based on their textual features. Because some textual features may be more important than others for the purpose of defining "comps", we allow for unequal "feature weights" for each textual variable in the distance metric. Building upon the previous literature [30], we estimate the feature weights using cross-validation in conjunction with regularization. Given the distance metric with estimated feature weights, we use a kernel-based approach [13] to assess the box office performance for new scripts.

Applying our proposed method to a database of 300 movie shooting scripts, we find that the proposed method outperforms several benchmark methods in predicting box office performance. Through a hypothetical portfolio selection scenario, we demonstrate the potential economic significance of our approach by showing that our method generates portfolio returns that are significantly higher than portfolios selected by a comps-based approach that mimics the current industrial practice. These results, albeit preliminary in nature due to several limitations of our data set (discussed in Section 2.4), suggest that the proposed methodology is promising and worthwhile of further testing.

The contribution of our research is threefold. First, to the best of our knowledge, this paper is the first that collects and analyzes *actual movie scripts* (about 120-pages each). This is a major step forward compared to previous research that studies only "spoilers" (1-page storyline summaries written by viewers after they watch the movie) [8], or other post-production information to predict box office performance [9], and thus much closer to the actual set of

- *J. Eliashberg is with Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania. E-mail: eliashberg@wharton.upenn.edu.*
- *S.K. Hui is with the Stern School of Business, New York University, 40 West 4th Street, Tisch 910, New York, NY 10016. E-mail: khui@stern.nyu.edu.*
- *Z. John Zhang is with Wharton School, University of Pennsylvania, Philadelphia, Pennsylvania. E-mail: zjzhang@wharton.upenn.edu.*

Manuscript received 1 Oct. 2013; revised 15 Jan. 2014; accepted 10 Feb. 2014. Date of publication 20 Feb. 2014; date of current version 26 Sept. 2014.
Recommended for acceptance by J. Pei.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2014.2306681

information that film producers have at the point of greenlighting. Second, we show that the kernel approach outperforms both regression and tree-based methods [6], [8] in the context of assessing box office performance. Third, the estimated "feature weights" provide some insights about which textual features require particular attention when identifying useful "comps" for a new script.

The remainder of this paper is organized as follows. Section 2 provides an overview of the script data set, describes how we extract textual information from scripts, and discusses several key limitations of our data set. Section 3 describes the kernel-based approach and discusses how we estimate the "feature weights". In Section 4, we compare our method with other benchmark methods and present a hypothetical portfolios selection scenario. Finally, Section 5 concludes with directions for future research.

2 Extracting Textual Features from Movie Scripts

Our data set is comprised of 300 movies released between 1995 and 2010, whose shooting scripts are available online. While the script has widely been viewed as a key determinant of how consumers will react to a movie and hence a key driver of box office performance [3], [11], [12], [14], [27], no prior academic research has ever collected and analyzed actual scripts. This lack of attention by academics may be due to the amount of efforts entailed in compiling an adequate data set, as discussed in Sections 2.1, 2.2, 2.3 below. For each script, we record the US box office revenue and production budget from the Internet Movie Database (IMDB) (www.imdb.com) and the Box Office Mojo database (www.boxofficemojo.com).

2.1 Genre and Content Variables

The highest level of textual information in movie content can be summarized by its genre and "content" variables. The genre of a script summarizes the overall theme of a movie and helps identify its target audience. At a more detailed level, the "content" variables measure various aspects of the story line of a script: e.g., what is the premise of the story line? Is the setting familiar to most viewers? Is the ending of the story logical/surprising?

We asked two independent readers trained in film studies to read each script and answer a questionnaire about the genre and the story line, as shown in Table 1.[1] We consider eight possible genres (note that the genre categories are not mutually exclusive). Having categorized a script into one or more genre(s), readers then answer a set of 24 "content" questions about the storyline for each script (as shown in Table 1), developed by [8] in their analysis of movie story lines. These questions are simple "yes or no" questions that have been identified by screenwriting experts as important aspects of movie scripts [3], [11], [12], [14], [27], thus providing an informative set of textual features. We average the two readers' (0/1) responses for each question.

While we acknowledge that the above procedure is necessarily subjective as it involves human input, there is no feasible alternative available as computers obviously cannot yet understand scripts. To explore the extent of subjectivity, we study the inter-rater agreement between the two readers. Across genre and content questions, both readers give the same answer (yes/no) around 83 percent of the time, suggesting reasonable degree of agreement.

2.2 Semantic Variables

The second layer of textual information captures the structure by which a movie script is written, and provides a "preview" of how the final movie will look like. As shown in Fig. 1, a script is organized into interior/exterior *scenes*, and each scene is comprised of *dialogues* spoken by the different characters. At the scene level, we obtain an estimate of the total number of scenes in the movie, and how often the characters interact in interior or exterior space [3]. At the dialogue level, the script carries information about the manner by which characters communicate with each other: whether they tend to give long prose or short dialogues, and how evenly these dialogues are distributed among the characters. Most movie experts agree dialogues affect viewers' enjoyment [3], [11], [12], [14]. To capture the aforementioned semantic information, we focus on scene variables (i and ii) and dialogue variables (iii)-(v) below.

i) Total number of scenes (NSCENE).
ii) Percentage of interior scenes (INTPREC).
iii) Total number of dialogues (NDIAG).
iv) Average length of dialogues (AVGDIAGLEN).
v) The "concentration index" of dialogues (DIAGCONC).

We use the Herfindahl-Hirschman (HH) index [16] to compute the concentration index of dialogues (DIAGCONC), defined as $DIAGCONC = \sum_i s_i^2$, where s_i denote the share of the dialogues by character i. The HH index may take value between 0 and 1, with higher HH index indicating that the dialogues are more concentrated to a few characters. In our data set, the HH index ranges from 0.04 to 0.41.

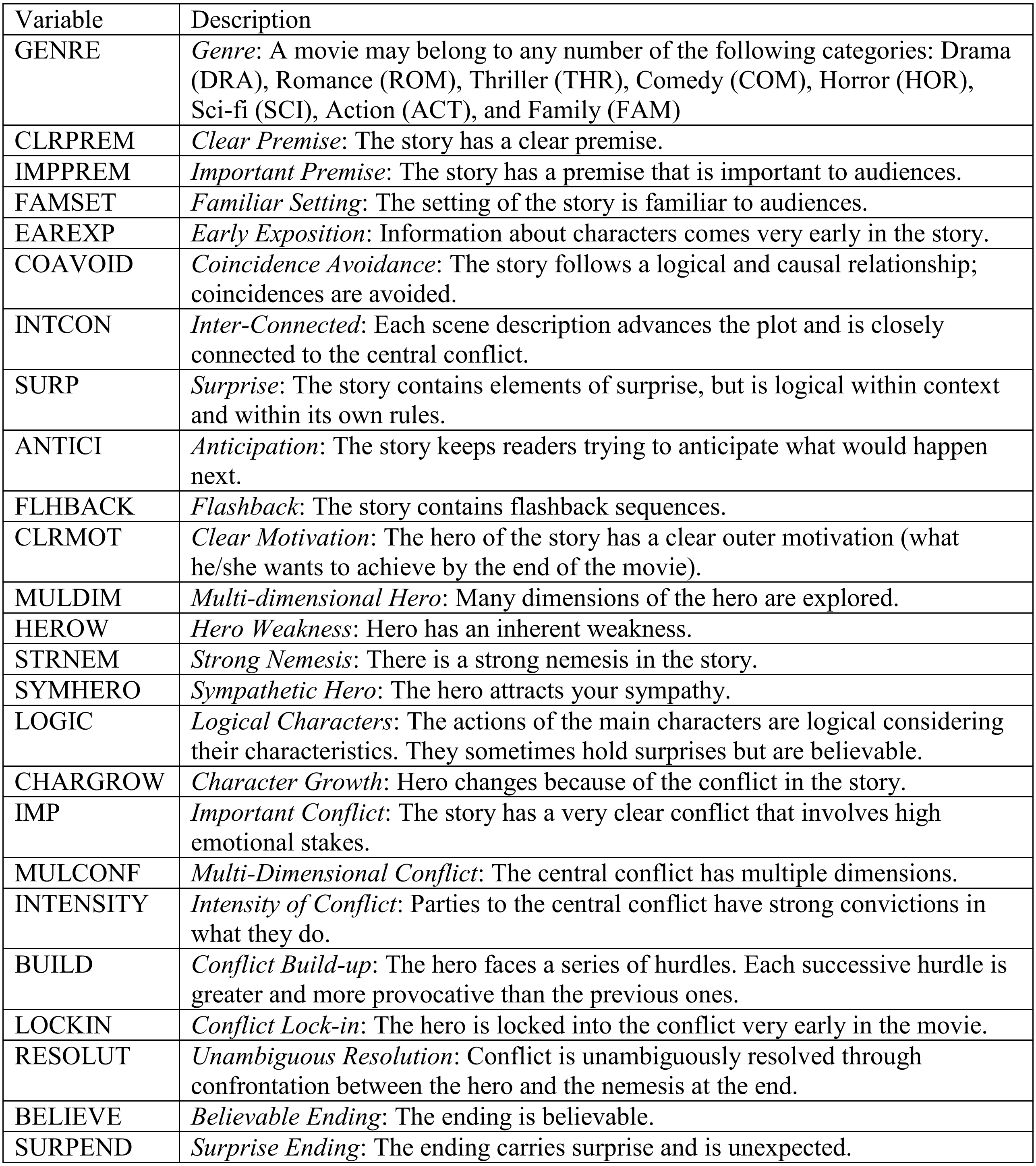
2.3 Bag-of-Words Variables

The third layer of textual information comes from the actual words used in the script, captured using a "bag-of-words" representation that is commonly used in natural language processing applications [24]. Bag-of-words representation has been applied to business applications such as document retrieval [4], text classification [23] and automated text content analysis [17].

Words used in a script and their usage frequencies are the building blocks of a story line. In particular, the frequencies of key terms (e.g., the use of vulgar words) can set the overall tone, language, theme, and sentiment of the movie, beyond that captured by genre and content variables. We extract bag-of-words information as follows. First, we eliminate all punctuations, a standard list of English names, and "stop words" (e.g., a, an, is, am, are, this, that, him, her) [24]. Next, we use a "stemming" algorithm [32] to reduce each word to its simplest form (e.g., "going" is reduced to

[1] Given more resources, studios should employ more readers to minimize potential subjectivity.

TABLE 1
Summary Description of Genre and Content Variables Extracted from Each Script



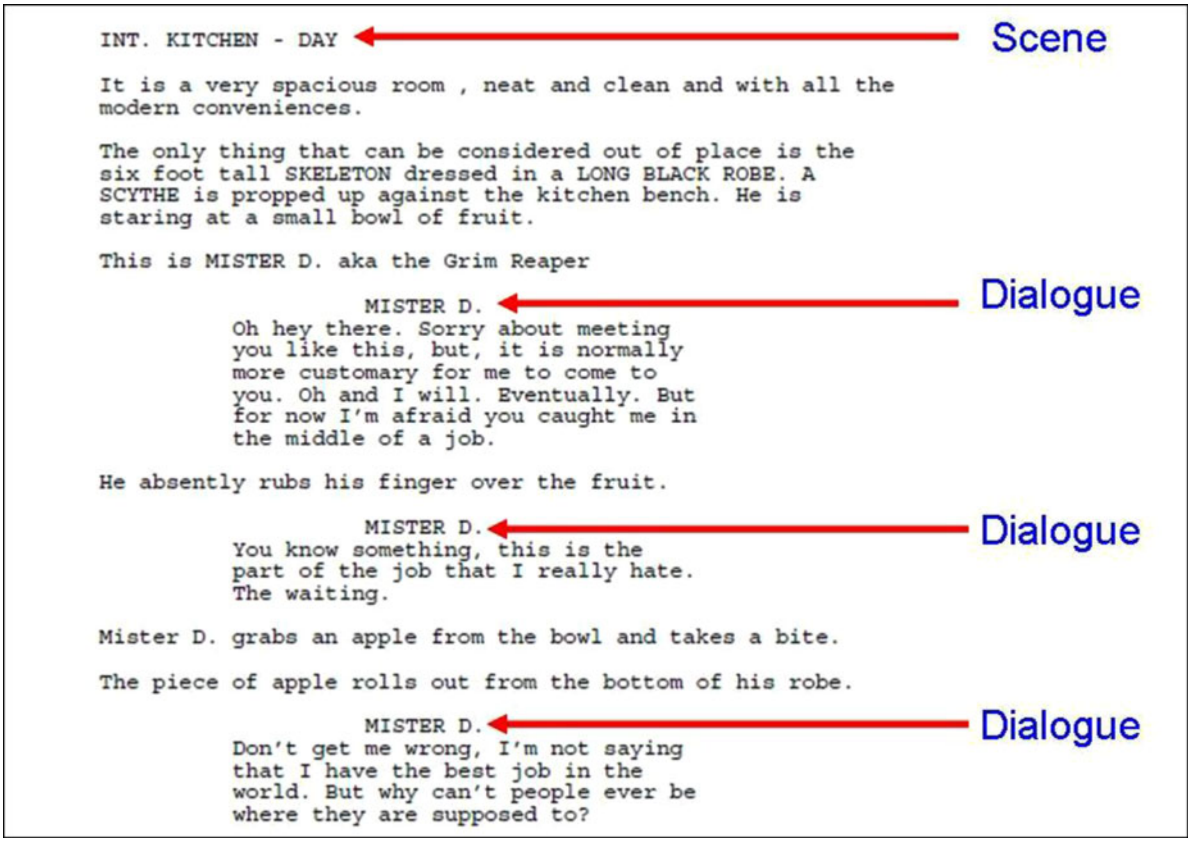
Variable	Description
GENRE	*Genre*: A movie may belong to any number of the following categories: Drama (DRA), Romance (ROM), Thriller (THR), Comedy (COM), Horror (HOR), Sci-fi (SCI), Action (ACT), and Family (FAM)
CLRPREM	*Clear Premise*: The story has a clear premise.
IMPPREM	*Important Premise*: The story has a premise that is important to audiences.
FAMSET	*Familiar Setting*: The setting of the story is familiar to audiences.
EAREXP	*Early Exposition*: Information about characters comes very early in the story.
COAVOID	*Coincidence Avoidance*: The story follows a logical and causal relationship; coincidences are avoided.
INTCON	*Inter-Connected*: Each scene description advances the plot and is closely connected to the central conflict.
SURP	*Surprise*: The story contains elements of surprise, but is logical within context and within its own rules.
ANTICI	*Anticipation*: The story keeps readers trying to anticipate what would happen next.
FLHBACK	*Flashback*: The story contains flashback sequences.
CLRMOT	*Clear Motivation*: The hero of the story has a clear outer motivation (what he/she wants to achieve by the end of the movie).
MULDIM	*Multi-dimensional Hero*: Many dimensions of the hero are explored.
HEROW	*Hero Weakness*: Hero has an inherent weakness.
STRNEM	*Strong Nemesis*: There is a strong nemesis in the story.
SYMHERO	*Sympathetic Hero*: The hero attracts your sympathy.
LOGIC	*Logical Characters*: The actions of the main characters are logical considering their characteristics. They sometimes hold surprises but are believable.
CHARGROW	*Character Growth*: Hero changes because of the conflict in the story.
IMP	*Important Conflict*: The story has a very clear conflict that involves high emotional stakes.
MULCONF	*Multi-Dimensional Conflict*: The central conflict has multiple dimensions.
INTENSITY	*Intensity of Conflict*: Parties to the central conflict have strong convictions in what they do.
BUILD	*Conflict Build-up*: The hero faces a series of hurdles. Each successive hurdle is greater and more provocative than the previous ones.
LOCKIN	*Conflict Lock-in*: The hero is locked into the conflict very early in the movie.
RESOLUT	*Unambiguous Resolution*: Conflict is unambiguously resolved through confrontation between the hero and the nemesis at the end.
BELIEVE	*Believable Ending*: The ending is believable.
SURPEND	*Surprise Ending*: The ending carries surprise and is unexpected.

"go"). We then tabulate all the unique stemmed words that occur in one or more documents to produce a word-document matrix.

Even after stemming and eliminating stop words, the resulting word-document matrix still has very high dimension, as more than 25,000 unique words have appeared in one or more scripts. Thus, following [8] we compute an "importance index" for each word, defined as follows:

$$I_i = \left(1 - \frac{d_i}{D}\right) \times N_i, \qquad (1)$$

where d_i denotes the number of scripts containing the ith word, D denotes the total number of scripts, and N_i is the total frequency of occurrence of the ith word across all scripts. The importance index defined here is similar to TF-IDF weights [24], where words that occur frequently but in



```
INT. KITCHEN - DAY

It is a very spacious room , neat and clean and with all the
modern conveniences.

The only thing that can be considered out of place is the
six foot tall SKELETON dressed in a LONG BLACK ROBE. A
SCYTHE is propped up against the kitchen bench. He is
staring at a small bowl of fruit.

This is MISTER D. aka the Grim Reaper

                    MISTER D.
          Oh hey there. Sorry about meeting
          you like this, but, it is normally
          more customary for me to come to
          you. Oh and I will. Eventually. But
          for now I'm afraid you caught me in
          the middle of a job.

He absently rubs his finger over the fruit.

                    MISTER D.
          You know something, this is the
          part of the job that I really hate.
          The waiting.

Mister D. grabs an apple from the bowl and takes a bite.

The piece of apple rolls out from the bottom of his robe.

                    MISTER D.
          Don't get me wrong, I'm not saying
          that I have the best job in the
          world. But why can't people ever be
          where they are supposed to?
```

Fig. 1. Semantic structure of a script.

fewer documents are assumed to be more important.[2] We keep only the top 100 most "important" words. [3] Finally, we perform latent semantic analysis (LSA) [7] to further reduce the dimensionality of the word-document matrix. Based on singular-value decomposition (SVD) [7], [21], LSA allows us to index each script by a set of "scores". Because the singular values shows an "elbow" at the two singular-value solution [18], we retain two latent semantic scores for each script, labeled as LS1 and LS2, and use them as textual features for further analyses.

Although researchers typically do not assign meaning to the latent dimensions identified by LSA and only use them as predictors [7], we conduct additional exploratory analyses to gain some intuitions about what LS1 and LS2 may represent, which may shed some light on how they are related to segment interest. Towards that end, we study the singular vectors corresponding to LS1 and LS2 to see which words load heavily onto each vector; the list of words are shown in Appendix I, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2014.2306681. We speculate that LS1 is related to a particular setting (with words such as "robot", "sheriff", "pilot", "platform", "sword", "chief", "evil", "vampire", "tank"), while LS2 appears to be related to the style of languages in the dialogues (e.g., "fxxx", "yeah", "shit", "dude", "gonna"). Higher value of LS2 indicates more prevalent use of vulgarity, which has obvious implications for the segment of audience that the movie attracts; for example, a movie with too much vulgarity will likely be given an "R" rating by the MPAA, which in turn excludes children and teens from the potential audience.

2.4 Summary and Potential Data Limitations

Summary statistics of each variable in our data set are shown in Table 2. All textual variables and the (log-) production budget are then standardized [13], and used as predictors in a kernel-based approach (described in Section 3) to forecast box office performance.

At this point, we would like to point out three important caveats of our data set. First, our data set is comprised of only scripts that are actually produced into movies. Depending on how our method is applied, we may face the problem of sample selection bias (Heckman 1979), if scripts that are produced are systematically different from those that are not. Due to this limitation, it is most appropriate to use the proposed methodology as a decision aid during the final stages of green-lighting, where movie makers may seek additional input to help decide whether or not to green-light a script that is perceived to be "at the margin". In that setting, the sample selection bias is likely to be minimal because, *ex ante*, the decision of whether or not to produce the script could have gone either way. In contrast, the

[2] In the TF-IDF specification, the term $\left(1-\frac{d_i}{D}\right)$ in Eq. (1) is replaced by $\log\left(\frac{D}{d_i}\right)$, both of which are decreasing functions of d_i. We use the specification in Eq. (1) to maintain consistency with the previous literature [8].

[3] The results remain substantially unchanged if we retain 500 words instead of 100 words; the MSEs for Kernel-I and Kernel-II methods change by less than 5 percent.

TABLE 2
Summary Statistics of All Variables (Before Standardization)

Variable	Mean	SD	Min	Max
Response variable				
LNBOX	3.79	1.33	-3.51	6.63
LNBUDGET	3.57	0.98	-0.74	5.52
y_i	0.21	0.97	-5.30	2.95
Textual variables				
GENRE_DRA	0.53	0.43	0.00	1.00
GENRE_ROM	0.40	0.38	0.00	1.00
GENRE_THR	0.18	0.35	0.00	1.00
GENRE_COM	0.26	0.39	0.00	1.00
GENRE_HOR	0.11	0.29	0.00	1.00
GENRE_SCI	0.19	0.35	0.00	1.00
GENRE_ACT	0.47	0.44	0.00	1.00
GENRE_FAM	0.08	0.25	0.00	1.00
CLRPREM	0.90	0.25	0.00	1.00
IMPPREM	0.64	0.33	0.00	1.00
FAMSET	0.81	0.31	0.00	1.00
EAREXP	0.86	0.27	0.00	1.00
COAVOID	0.86	0.29	0.00	1.00
INTCON	0.81	0.31	0.00	1.00
SURP	0.92	0.21	0.00	1.00
ANTICI	0.85	0.30	0.00	1.00
FLHBACK	0.42	0.47	0.00	1.00
CLRMOT	0.94	0.22	0.00	1.00
MULDIM	0.90	0.23	0.00	1.00
HEROW	0.44	0.40	0.00	1.00
STRNEM	0.58	0.46	0.00	1.00
SYMHERO	0.97	0.14	0.00	1.00
LOGIC	0.99	0.10	0.00	1.00
CHARGROW	0.57	0.42	0.00	1.00
IMP	0.92	0.22	0.00	1.00
MULCONF	0.76	0.37	0.00	1.00
INTENSITY	0.96	0.15	0.00	1.00
BUILD	0.80	0.35	0.00	1.00
LOCKIN	0.88	0.29	0.00	1.00
RESOLUT	0.57	0.42	0.00	1.00
BELIEVE	0.90	0.24	0.00	1.00
SURPEND	0.56	0.43	0.00	1.00
LS1	0.00	1.00	-2.32	2.04
LS2	0.01	1.00	-5.18	2.10
NSCENE	153.08	60.50	33.00	389.00
INTPREC	0.63	0.15	0.02	1.00
NDIAG	827.98	213.19	284.00	1915.00
AVGDIAGLEN	10.47	1.96	6.28	18.52
DIAGCONC	0.16	0.06	0.04	0.41

proposed method should not be used as an initial screening tool, as the characteristics of the scripts at the initial screening stage can be very different from those that are ultimately made into the movies, and as a result the sample selection problem is likely to be very significant.

Second, the scripts that we collected are "shooting scripts", i.e., scripts that are used in actual production of the movie. However, a script may have evolved over time from the point of green-lighting to actual production, and as a result our analysis here may not truly reflect the predictive performance of our method at the point of green-lighting. The same caveat applies to the production budget as well.

Finally, we note that our sample size of 300 scripts is smaller than the sample size typically used in machine learning studies, albeit comparable to previous research that studies movie spoilers (e.g., [8] looks at 281 spoilers). As can be expected, it takes a lot of time and effort to read every script and answer the genre and content questions (see Section 2.1); the current sample of 300 scripts have already taken more than two years to collect. Due to the small sample size, our results should be viewed as preliminary in nature.

With the aforementioned caveats in mind, we now discuss the kernel-based approach to forecast box office performance using textual features of movie scripts.

3 A Kernel-Based Approach to Forecast Box Office Performance

The kernel-based method utilizes a distance metric to determine the "similarity" between a new observation and each observation in the training database [13]. As such, the kernel-based approach is free of functional form assumptions (unlike regression or tree-based methods), thus allowing it to flexibly capture the complex relationship between the textual features of a script and box office performance. Therefore, we feel that a kernel-based approach is especially appropriate for the purpose of green-lighting movie scripts, where the "correct" relationship between textual variables in a script and box office revenue is impossible to specify a priori.

Another practical advantage of the kernel-based approach is that it directly parallels the current green-lighting approach that studios utilize, as the "comps"-based approach is a special case of a kernel-based method. Thus, our approach directly speaks to the intuitions of studio managers, who are already comfortable with the process of obtaining "comparables" before making predictions about box office revenue. As a result, it is straightforward to communicate our results to studio managers, a key advantage that has important implications for actual implementation [9], as it ensures that the output of our decision support system is "business friendly" and can be seamlessly communicated to decision makers [5].

3.1 The Kernel-Based Approach

We use the following notations. Scripts in the training sample are indexed by $i = 1, \ldots, N$. Each script is comprised of J distinct "features," (shown in Table 2) and is denoted as $\vec{x}_i = \{x_{ij}\}_{j=1,\ldots,J}$, along with a "response" variable y_i. We define the response variable (y_i) for each movie by its (transformed) return of investment (ROI). Specifically:

$$y_i = \log(\mathrm{BOX\ OFFICE_i/BUDGET_i}). \qquad (2)$$

.

Later, we use a kernel-based approach to predict y_i for a new movie, then transform it to a prediction of box office revenue $(\text{Predicted Box Office}_i = \text{Budget}_i(e^{y_i}))$.We specify (transformed) ROI as the response variable in the kernel-based method because such specification confers several statistical advantages. First, the distribution of y_i is much closer to normality than box office revenues, which has a heavy right tail [34]. Although normality of the response variable is not a formal requirement for kernel-based methods [29], researchers have found that it is often advisable to pre-process observed responses to make the empirical distribution more “bell-shaped”, and in particular, taking a log-transform to reduce a heavy right tail [6]. Second, the definition of y_i already incorporates the (linear) effect of log- production budget as an “offset” variable [25]. This allows the kernel-based approach to partial out most of the effect of production budget, and focus on capturing the remaining variations driven by the textual variables. Third, empirically we find that our specification of y_i in Eq. (2) yields better predictive results compared to using box office revenue as the response variable.

As discussed in Section 2, the features we consider here are the textual variables extracted from each script along with its production budget. The distance metric between two observations is defined, based on (weighted) Euclidean distance, as follows: [4]

$$d(\vec{x}_i, \vec{x}_l) = \sqrt{\sum_{j=1}^{J} v_j^2 (x_{ij} - x_{lj})^2}, \tag{3}$$

where $\vec{v} = \{v_j\}_{j=1,\ldots J}$ is a vector of “feature weights”. A predictor variable that has a larger value of v_j is considered more important when defining how similar two observations are.

Given Eq. (3), the kernel-based method makes prediction for a new observation as follows. Index the new observation by i^*, with a vector of features $\vec{x}_{i*}$; the goal is to predict the response variable y_{i*}. First, using Eq. (3), we compute all the pairwise distances $d(\vec{x}_{i*}, \vec{x}_l)$ between the new observation and each observation in the training database. Then, we define the “weight” of a training observation based on its distance from the new observation:

$$w_{li^*} = \frac{\exp(-\theta\, d(\vec{x}_{i*}, \vec{x}_l)}{\sum_l \exp(-\theta\, d(\vec{x}_{i*}, \vec{x}_l))}. \tag{4}$$

.

Note that the scaling parameter θ defines the extent to which the weights are related to distances, which will be calibrated in Section 3.2. The larger θ is, the more severely the observations that are more dissimilar to the new observation are down-weighted. Finally, the response variable y_{i^*} is predicted by the weighted sum of the response variables of the training observations:

$$\hat{y}_{i^*} = \sum_l w_{li^*} y_l. \tag{5}$$

.

3.2 Calibration of the Tuning Parameter θ

We calibrate the “tuning parameters” θ and $\vec{v}$ using a combination of domain knowledge and cross validation [2], [22]. Specifically, we divide our data into two sets: a training sample of 265 scripts covering movies released in 2008 or earlier, and a holdout sample of 35 scripts of movies released after January 2009. We calibrate our parameters using only the training sample, and use the holdout sample to assess our model’s predictive capability. Note that the above definition of training and holdout sample preserves “time consistency”, i.e., only movies that are released before the focal movie are used in making revenue predictions.

We calibrate θ following an argument in [30]. Because θ is the “bandwidth” parameter of the Gaussian kernel in Eq. (4), it makes sense to set θ such that it is roughly in line with the range of the distances $d(\vec{x}_i, \vec{x}_l)$. To see this, notice that if θ is set “too small” ($\theta \approx 0$), every movie in the training sample will be weighted roughly the same, thereby reducing discriminatory power. In contrast, if θ is too large compared to the range of $d(\vec{x}_i, \vec{x}_l)$, the weighted average in Eq. (5) will be dominated by the closest comparable, thus increasing the bias of our prediction. The underlying concept here is the “bias-variance tradeoff”: larger θ leads to lower bias yet higher variance, while smaller θ results in higher bias and lower variance [13].

Given the conceptual argument above, we set the value of θ by appealing to studios’ domain knowledge. As discussed earlier, studio managers typically look at no more than 10 “comps” when making a green-lighting decision. Therefore, we select θ such that any “comp” beyond the 10th will receive minimal weight; this is achieved by setting θ so that, on average, the 10th comp receives a weight that is proportional to the density of a standard normal distribution at two standard deviations from the mode, thus the 11th or further comps have weights that are negligible. Specifically, we set θ such that $\exp(-\theta(d_{(10)} - d_{(1)}) \approx \phi(2)/\phi(0)$, which results in $\theta = 17$.[5]

3.3 Calibration of the Feature Weight ($\vec{v}$)

Next, we calibrate the “feature weights” $\vec{v}$. As a starting point, a seemingly reasonable “default” choice is to put equal weights on every variable, i.e., set $v_j = 1$ for all j [13].[6] We refer to this as the Kernel-I approach; we will evaluate its predictive performance versus the Kernel-II approach that involves unequal feature weights.

We propose the following approach, based on regularization and cross-validation, to calibrate the feature weights $\vec{v}$ for the Kernel-II approach. We first define the leave-one-out mean squared-error, *LOOMSE*, a key component of our objective function described later, as follows. We let $i = 1, \ldots, n (n = 265)$ index the scripts in the training sample, and let $\hat{z}_i(\theta, k, \vec{v})$ be the predicted value of the (log-) box office revenue of the ith script, when all except the ith script are used as the training data. z_i denotes the actual log- box office revenue for the ith script. The *LOOMSE* is defined as

$$LOOMSE(\theta, k, \vec{v}) = \frac{1}{n} \sum_{i=1}^{n} (z_i - \hat{z}_i(\theta, k, \vec{v}))^2. \tag{6}$$

.

While it is tempting to directly minimize *LOOMSE* as a function of $\vec{v}$, previous research [20] has shown that such

[4] Note that other distance metrics have been proposed in [13]; we leave the selection of optimal distance metric for future research.

[5] We have conducted a robustness check by considering $d_{(9)}$ and $d_{(11)}$ instead of $d_{(10)}$; this corresponds to setting $\theta = 2.0$ and 1.5, respectively. The MSEs for Kernel-I and Kernel-II change by less than 3 percent. Details are available upon request.

[6] Because the feature variables are all standardized before entering into our model as predictors, the condition $v_j = 1 \forall j$ implies equal weights across all features.

TABLE 3
Calibrated Feature Weights in the Kernel-II Approach

Variable	v_j^2	Variable	v_j^2	Variable	v_j^2	Variable	v_j^2
EAREXP	2.23	CHARGROW	1.24	INTCONN	0.94	SYMHERO	0.79
BUDGET	1.90	LS1	1.15	LOGIC	0.94	SURPEND	0.78
STRNEM	1.79	NDIAG	1.11	LOCKIN	0.94	IMPPREM	0.73
GENRE_ROM	1.54	NSCENE	1.11	BUILD	0.94	MULDIM	0.72
GENRE_THR	1.49	FLHBACK	1.09	HEROW	0.90	SURP	0.65
LS2	1.39	GENRE_ACT	1.05	INTPERC	0.90	CLRMOT	0.62
ANTICI	1.33	MULCONF	1.04	GENRE_FAM	0.89	GENRE_DRA	0.61
GENRE_SCI	1.32	DIAGCONC	1.01	INTENSITY	0.88	CLRPREM	0.59
FAMSET	1.30	GENRE_COM	1.01	BELIEVE	0.87	AVGDIAGLEN	0.54
RESOLUT	1.26	COAVOID	0.97	IMP	0.83	GENRE_HOR	0.47

approach tends to lead to over-fitting. Thus, [20] restricts the elements of $\vec{v}$ to take only a finite set of values, thus reducing the degrees of freedom in $\vec{v}$. In our setting, however, it is difficult to a priori identify a set of appropriate values for $\vec{v}$. Instead, we propose using a "regularization" approach to avoid overfitting. Specifically, in addition to *LOOMSE*, we add a "penalty term" that penalizes the distance between $\vec{v}$ and the vector of 1 s (which correspond to the a priori assumption that all features are weighted equally). Formally,

$$\text{Objective Function} = LOOMSE + \lambda \sum_{j=1}^{J} \left(v_j^2 - 1\right)^2. \qquad (7)$$

We then calibrate λ (the extent of the complexity penalty) using a cross-validation approach (see Appendix II, available online), which results in a choice of $\lambda = 0.05$. Finally, given the choice of λ, we minimize the objective function in Eq. (7) as a function of $\vec{v}$ using the Nelder-Mead method [31], to arrive at the Kernel-II approach. The resulting estimates of $\vec{v}$ are shown in Table 3, which provides some insights about which textual variables are most important for identifying "comps" for a focal script. As can be seen, the five most important features are (i) early exposition, (ii) production budget, (iii) strong nemesis, (iv) genre—romance, and (v) genre—thriller. Beyond the obvious genre and production budget variables that studio already consider, "content" features such as "early exposition" and "strong nemesis" also appear among the most important variables. Thus, our findings are consistent with the belief by movie experts [3] that "early exposition", i.e., communicating the general theme of the movie as early as possible (preferably within the first ten minutes of the movie), is an important content feature of a script. Further, expert scriptwriters [11], [12] also believe that a strong nemesis is central for advancing the story line, as it helps set up a conflict between the protagonist and his/her nemesis.

4 Empirical Results

4.1 Holdout Prediction

Keeping in mind the caveats of our data set (discussed in Section 2.4), we apply the proposed Kernel-I/II approaches to predict the box office revenue of each movie in the holdout sample (35 movies released after January 2009). Predictive performance is measured by mean squared error on log- box office revenue (in $Million). We compare our proposed methods to six benchmark methods, which include the linear model (i), tree-based methods (ii and iii) proposed in the previous literature [6] and [8], as well as comps-based methods (iv, v, and vi) that are constructed to (loosely) reflect studios' current practice.

The implementation details for the two tree-based methods (ii and iii) are discussed in Appendix III, available online. For the comps-based methods (iv-vi), we consider three specifications. For method (iv), comps are defined based only on genre. Specifically, the feature weights are specified so that the "genre" variables receive a weight of 1.0, while all other variables receive feature weights of zero. Similarly, for method (v), comps are defined based only on production budget: the feature weight of "production budget" is set to be 1.0, while all other feature weights are equal to zero. For method (vi), both genre and production budget are considered; an equal weight is put on genre and production budget, and all other variables are given zero weights. For all three specifications, we set $\theta = 0$ (i.e., a simple average is taken) and the weights w to zero after the 10th "comp" to maximally reflect current practice.

The results of all eight methods are summarized in Table 4. As a baseline for comparison, multiple regression (method i) has a holdout MSE of 0.6571. Tree-based methods, which includes Bag-CART [8] (method ii) and Bayesian additive regression tree [6] (method iii), provide better predictive results. Across both methods, the improvement over regression is around 8.5 percent, which is consistent with the findings in [8]. The results of method (iv)-(vi) show that the performance of comps-based methods depend critically on how the comps are constructed. Comps-based approaches generated using only genre (method iv) or only production budget (method v) provide predictive performances that are no better than multiple regression, while a comps-based method that considers both genre and production budget (method vi) provides superior predictive performance. Specifically, method (vi) gives holdout MSE (0.5395) that is 17.9

TABLE 4
Holdout Prediction Performance on 35 Movies Released After 2009

Method	Holdout MSE
(i) Multiple regression	0.6571
(ii) Bag-CART	0.6082
(iii) Bayesian additive regression tree (BART)	0.6010
(iv) Comps based only on genre	0.6576
(v) Comps based only on budget	0.6781
(vi) Comps based on genre and budget only	0.5395
(vii) Kernel-I (equal feature weights)	0.4096
(viii) Kernel-II (optimized feature weights)	0.3822

percent smaller than that of multiple regression. While it is rather surprising that a simple comps-based method outperforms more sophisticated tree-based methods, it provides some validation for the current managerial practice. Taking a step forward, the Kernel-I approach, which uses all textual variables (with equal weights) improve holdout predictive performance even further, with a holdout MSE of 0.4096, a 37.7 percent improvement over multiple regression. Finally, the Kernel-II approach, which allows for unequal feature weights, has the lowest MSE across all methods (0.3822). Compared to multiple regression, this represents an improvement of almost 42 percent.

The results in Table 4 suggest that studios should think beyond genre and the production budget, and consider a richer set of textual variables when making green-lighting decisions. Indeed, part of the improvements of using Kernel-I/II approaches come from considering not only movies that are of the same genre as the focal script, but also movies of *different* genres[7] (but otherwise similar in content, semantics, and word usage). Across the holdout sample, we find that about 78 percent of the top 10 comps for each script in the holdout sample have genre that overlap with the focal script, while 22 percent of the comps are of completely different genre. This highlights the value of using a richer set of textual information to define our distance metric.

To further explore the role of the three sets of textual variables (genre/content, semantics, and bag-of-word variables) in driving predictive performance, we conduct additional analyses that exclude certain sets of predictor variables. The results of eight different specifications are shown in Table 5. As can be seen, holdout MSE increases whenever a subset of textual variables is excluded, again indicating that each subset of textual predictors adds to predictive performance. Among the three types of textual variables, it appears that genre/content is the most important (i.e., predictor performance worsens the most when genre/content is excluded), followed by the semantic variables, with the bag-of-words variables being the least important.

4.2 Portfolio Selection

To demonstrate the potential economic significance of our proposed method, we conduct a hypothetical portfolio selection exercise that compares the performance of the comps-based approach (method vi), which reflects current practice, with our proposed Kernel-I/II methods.

We consider the following portfolio selection setting. Suppose we would like to pick r scripts to form a movie portfolio. First, based on the predicted box office revenue and the given production budget, we compute the predicted ROI of each of the 35 scripts in the holdout sample. Then, scripts in the holdout sample are ranked based on predicted ROI, and the r scripts that have the highest predicted ROI are selected. We vary r from 5 to 20, and compare the ROIs of the overall portfolios which are selected by the comps-based method (method vi), the Kernel-I, and the Kernel-II method, respectively.

The results are shown in Fig. 2. While there is a lot of variability in portfolio ROIs ((total box office – budget)/budget) across all methods, portfolios selected by Kernel-I and Kernel-II approaches consistently provide higher portfolio returns compared to those selected by the comps-based method. For instance, when $r = 10$ movies scripts are selected to form a portfolio, the selections by Kernel-I and Kernel-II method yield portfolio ROIs of 130.3 percent (Box office = \$1184.7M; Budget = \$514.5M) and 134.6 percent (Box office = \$1236.3M; Budget = \$527.0M), respectively, while the selection by the comps-based method yields a ROI of 76.4 percent (Box office = \$307.8M; Budget = \$174.5M).[8] Across different values of r (from 5 to 20), the median ROI of portfolios selected by Kernel-I and Kernel-II is around 134.0 and 134.1 percent (respectively), while the median ROI of portfolios selected by comps-based method is only around 83.9 percent. Thus, it is clear that the improvement in prediction accuracy afforded by the Kernel-I/II methods is also economically significant.

5 Discussion and Conclusion

In this paper, we developed a methodology, based on the kernel-based approach, to predict the revenue potential of

[7] Recall that a script can belong to more than one genre (see Section 3.1). A comp has genre that overlap with the focal script if at least one of these genre are the same. For example, if the focal script is of genre {action, romance}, a comp that has genre {action, thriller} “overlaps” with the focal script because they share a common genre (action).

[8] The opportunity cost of capital can be a factor in this analysis, but it is insignificant enough in our case such that we can avoid this complication.

TABLE 5
Holdout Predictive Performance (in Terms of MSE) for Kernel-I and Kernel-II Methods, When Different Sets of Predictors Are Excluded

	Kernel-I	Kernel-II
Original results	0.4096	0.3822
Bag-of-word excluded	0.4300	0.4021
Semantics excluded	0.4446	0.4219
Genre/content excluded	0.4362	0.4400
Bag-of-word & semantics excluded	0.4613	0.4433
Bag-of-word & genre/content excluded	0.4238	0.4402
Semantics & genre/content excluded	0.4639	0.4665
Bag-of-word & Semantics & genre/content excluded	0.5396	0.5250

movie scripts at the point of green-lighting. We collected a database of 300 movies scripts, and extracted three layers of textual information from each script: genre/content, semantics, and bag-of-words variables, using a combination of screenwriting domain knowledge, human input, and natural language processing techniques. Holdout prediction results suggested that our proposed Kernel-I and Kernel-II approaches outperform regression, tree-based method, and the comps-based approach. Further, through a hypothetical portfolio selection exercise, we showed that such improvement in predictive accuracy is economically significant. Importantly, because of its close similarity with the "comps"-based approach, our proposed approach speaks directly to the intuitions of studio managers, leading to key communication advantages which further enhance the actionability of our approach [5].

We now discuss several promising directions for future research. In the current research, we have mainly considered content variable and a script's semantic features (scenes and dialogues). In future research, one may consider a wider set of textual variables and features that extract more information from a script. For example, [33] propose a methodology to capture the "hidden sentiment factors" in reviews. Including such features may help improve predictive performance even further.

Further, the current research focuses on point prediction of box office revenue given a script. In order to apply our method in general portfolio optimization problems, one should also consider forecasting the predictive distribution of box office revenue. That would allow us to compute usual risk estimates such as value-at-risk (VaR) for any movie portfolio [19], [26], or conduct mean-VaR optimization [1] common in financial analysis and asset allocation.

Clearly, the next step of this research is to develop our methodology into a full-blown implementation through collaboration with movie studios, which can help alleviate some of the key data limitations discussed in Section 2.4. To date, we have worked with several independent studios and investors and provided box office forecast based on script information; generally, the feedback that we obtain have been positive. Several challenges remain, including how to scale up the database and update/maintain it over time, and how to communicate our findings to studios most effectively. We will continue to work on these problems,

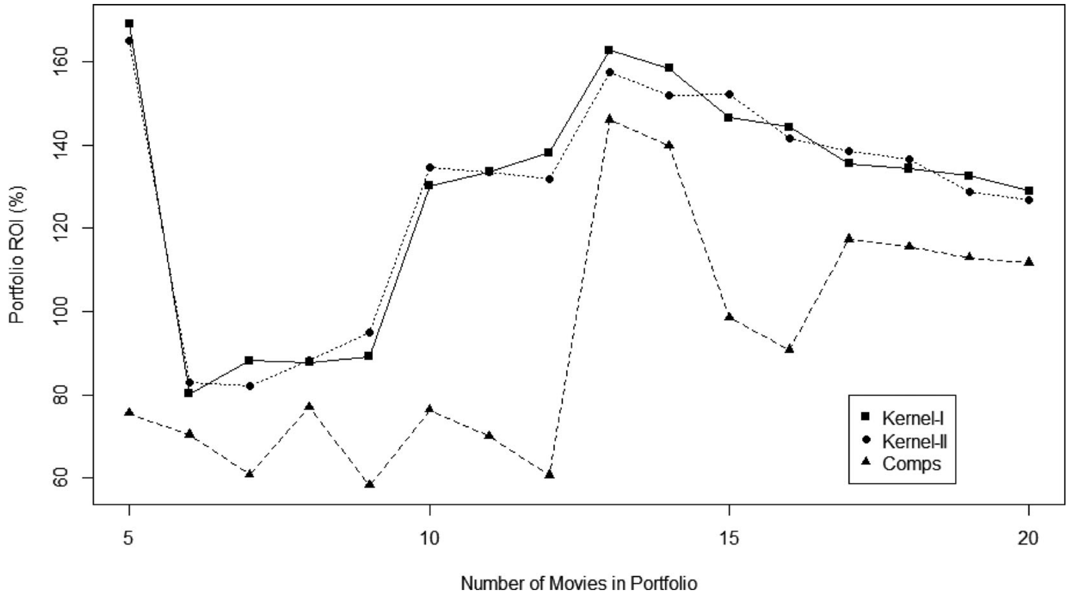


Fig. 2. Comparison of portfolio selection performance for the Kernel-I, Kernel-II, and comps-based methods.

with the goal of providing an effective decision support system for the green-lighting process that helps studios identify and produce better movies.

ACKNOWLEDGMENTS

The authors contributed equally to the paper and are listed alphabetically.

REFERENCES

[1] G.J. Alexander and M.B. Alexandre, "Economic Implications of Using a Mean-VaR Model for Portfolio Selection: A Comparison with Mean-Variance Analysis," *J. Economic Dynamics and Control*, vol. 26, no. 7/8, pp. 1159-1193, 2001.

[2] D.W.K. Andrews, "Asymptotic Optimality of Generalized CL, Cross-Validation, and Generalized Cross-Validation in Regression with Heteroskedastic Errors," *J. Econometrics*, vol. 47, no. 1991, pp. 359-377, 1991.

[3] I.R. Blacker, *The Elements of Screenwriting*. Macmilan Publishing, 1998.

[4] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research*, vol. 3, pp. 993-1022, 2003.

[5] L. Cao, "Domain-Driven Data Mining: Challenges and Prospects," *IEEE Trans. Knowledge and Data Eng.*, vol. 22, no. 6, pp. 755-767, June 2010.

[6] H. Chipman, E. Geroge, and R. McCulloch, "BART: Bayesian Additive Regresion Trees," *The Annals of Applied Statistics*, vol. 4, no. 1, pp. 266-298, 2010.

[7] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *J. Am. Soc. for Information Science*, vol. 41, no. 6, pp. 391-407, 1990.

[8] J. Eliashberg, S.K. Hui, and Z. John Zhang, "From Story Line to Box Office: A New Approach for Green-Lighting Movie Scripts," *Management Science*, vol. 53, no. 6, pp. 881-893, 2007.

[9] J. Eliashberg, C. Weinberg, and S. Hui, "Decision Models for the Movie Industry," *Handbook of Marketing Decision Models*, pp. 437-468, Springer, 2008.

[10] E.J. Epstein, *The Big Picture: The New Logic of Money and Power in Hollywood*. Random House, 2005.

[11] S. Field, *Screenplay: The Foundations of Screenwriting*. third ed., Dell Publishing, 1994.

[12] S. Field, *The Screenwriter's Problem Solver*. Dell Publishing, 1998.

[13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[14] M. Hauge, *Writing Screenplays that Sell*. HarperCollins Publishers, 1991.

[15] J.J. Heckman, "Sample Selection Bias as a Specification Error," *Econometrica*, vol. 47, no. 1, pp. 153-162, 1979.

[16] A.O. Hirschman, "The Paternity of an Index," *The Am. Economic Rev.*, vol. 54, no. 5, p. 761, 1964.

[17] D. Hopkins and G. King, "A Method of Automated Nonparametric Content Analysis for Social Science," *Am. J. Political Science*, vol. 54, no. 1, pp. 229-247, 2010.

[18] R.A. Johnson and D.W. Wichern, *Applied Multivariate Statistical Analysis*. sixth ed., Pearson, 2007.

[19] P. Jorion, *Value at Risk: The New Benchmark for Managing Financial Risk*. third ed., McGraw-Hill, 2006.

[20] R. Kohavi, P. Langley, and Y. Yun, "The Utility of Feature Weighting in Nearest-Neighbor Algorithms," *Proc. Ninth European Conf. Machine Learning*, 1997.

[21] T. Landauer, P.W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," *Discourse Processes*, vol. 25, no. 2-3, pp. 259-284, 1998.

[22] K.C. Li, "Asymptotic Optimality for C_p, C_L, Cross-Validation and Generalized Cross-Validation: Discrete Index Set," *Annals of Statistics*, vol. 15, pp. 958-975, 1987.

[23] Y.H. Li and A.K. Jain, "Classification of Text Documents," *The Computer J.*, vol. 41, no. 8, pp. 537-546, 1998.

[24] C.D. Manning and H. Schuetze, *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[25] P. McCullagh and J. Nelder, *Generalized Linear Models*. second ed., Chapman and Hall, 1989.

[26] A. McNeil, R. Frey, and P. Embrechts, *Quantitative Risk Management Concepts Techniques and Tools*. Princeton University Press, 2005.

[27] J. Monaco, *How to Read a Film*. Oxford University Press, 2000.

[28] MPAA, "Motion Picture Association of America Theatrical Market Statistics 2007," http://www.mpaa.org/2007-US-Theatrical-Market-Statistics-Report.pdf, 2013.

[29] H. Mukerjee, "Nearest Neighbor Regression with Heavy-Tailed Errors," *The Annals of Statistics*, vol. 21, no. 2, pp. 681-693, 1993.

[30] A. Navot, L. Shpigelman, N. Tishby, and E. Vaadia, "Nearest Neighbor Based Feature Selection for Regression and Its Application to Neural Activity," *Proc. Advances in Neural Information Processing Systems (NIPS)*, vol. 18, pp. 995-1002, 2005.

[31] J.A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *Computer J.*, vol. 7, pp. 308-313, 1965.

[32] M.F. Porter, "An Algorithm for Suffix Stripping," *Program*, vol. 14, no. 3, pp. 130-137, 1980.

[33] X. Yu, Y. Liu, J.X. Huang, and A. An, "Mining Online Reviews for Predicting Sales Performance: A Case Study in the Movie Domain," *IEEE Trans. Knowledge and Data Eng.*, vol. 24, no. 4, pp. 720-734, Apr. 2012.

[34] W.D. Walls, "Modelling Heavy Tails and Skewness in Film Returns," *Applied Financial Economics*, vol. 15, pp. 1181-1188, 2005.

Jehoshua Eliashberg is the Sebastian S. Kresge professor of marketing and professor of operations and information management in the Wharton School at the University of Pennsylvania. His research has focused on various issues including new product development and feasibility analysis, marketing/manufacturing/R&D interface, and competitive strategies. He has particular interest in the media and entertainment, pharmaceutical, and the hi-tech industries.

Sam K. Hui received the BS and MS degree from Stanford University, and the PhD degree in marketing from the University of Pennsylvania. He is an assistant professor of marketing at New York University. His research interests include Bayesian models, retailing, consumer tracking, and entertainment (movie, DVD, TV, casinos, and online games).

Z. John Zhang received the bachelor's degree in engineering automatic and philosophy of science from the Huazhong University of Science and Technology (China), the PhD degree in history and sociology of science from the University of Pennsylvania as well as a PhD degree in economics from the University of Michigan. He is a professor of marketing and Murrel J. Ades professor at the Wharton School of the University of Pennsylvania. His primary expertise is in revenue models and pricing strategies.