Automation in Construction 43 (2014) 23–29

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Predicting construction cost overruns using text mining, numerical data and ensemble classifiers

Trefor P. Williams *, Jie Gong [1]

Civil Engineering, Rutgers University, Department of Civil and Environmental Engineering, 96 Frelinghuysen Road, Piscataway, NJ 08854-8014, USA

ARTICLE INFO

Article history:
Received 21 August 2013
Received in revised form 31 January 2014
Accepted 22 February 2014
Available online 18 March 2014

Keywords:
Construction cost
Data mining
Text mining
Prediction

ABSTRACT

This paper discusses how text describing a construction project can be combined with numerical data to produce a prediction of the level of cost overrun using data mining classification algorithms. Modeling results found that a stacking model that combined the results from several classifiers produced the best results. The stacking ensemble model had an average accuracy of 43.72% for five model runs. The model performed best in predicting projects completed with large cost overruns and projects near the original low bid amount. It was found that a stacking model that used only numerical data produced predictions with lower precision and recall. A potential application of this research is as an aid in budgeting sufficient funds to complete a construction project. Additionally, during the planning stages of a project the research can be used to identify a project that requires increased scrutiny during construction to avoid cost overruns.

1. Introduction

Many factors affect construction cost overruns. Possibly, indicators provided in bidding text documents can identify construction projects that potentially will have large cost overruns. Text summaries of what is to be constructed for a project, and project line item text are available from project bidding data collected by state transportation agencies. Additionally, numeric data are available at the time of the bid opening including the projects' magnitude, and the number of bidders. It has now become possible to combine various text and data mining methods to project data to attempt to make predictions.

With the development of text mining algorithms that allow the extraction of information from the text, it may be possible to find indications of the projects' nature and likelihood to experience cost overruns. Text mining can be defined as the automatic discovery of previously unknown information from unstructured text data. Text mining involves extracting information of interest from text documents and then the use of data mining to discover new associations among the extracted information [1].

Text mining has been found to give excellent results for some predictive applications. For example, researchers have recently created software that detects when and where disease outbreaks will occur based on two decades of New York Times articles and other on-line sources of data. The system was successful in forecasting outbreaks of disease, violence and significant numbers of deaths between 70 and 90% of the time [2]. A construction project will typically have extensive documents produced before construction is initiated including project-scoping documents, written specifications, and plans. All of these documents can be transformed into text files useable by text mining software. In the construction industry, controlling costs is always a particular concern. Owner organizations require knowledge of a completed projects' cost to budget sufficient funds to complete a project. Additionally, knowledge of a project's likelihood to have cost overruns can identify projects that need to receive increased scrutiny during the construction process to control costs.

The goal of this research is to determine if text descriptions of a projects' characteristics can be used to develop a predictive model of a competitively bid construction projects' expected cost overrun. This paper will discuss how text can be processed, combined with numeric values and classified using data mining algorithms to produce a prediction of the level of cost overrun for a construction project.

1.1. Background

There have been several applications of text mining to construction management problems. Existing construction text mining research has focused on methods of classifying documents and extracting information from databases. A prototype system that automatically classifies construction documents according to project components using data mining techniques was proposed by Caldas et al. [3]. Soibelman and Kim [4] addressed the need for data mining in the construction industry, and the possibility to identify predictable patterns in construction data that were previously thought to be chaotic. In that study, a prototype knowledge discovery and data mining (KDD) system was developed to find the cause of activity delays from a U.S. Army Corps of Engineer's

* Corresponding author. Tel.: +1 848 445 2880.
E-mail addresses: tpw@rci.rutgers.edu (T.P. Williams), jiegong.cee@rutgers.edu (J. Gong).
[1] Tel.: +1 848 445 2881.

http://dx.doi.org/10.1016/j.autcon.2014.02.014
0926-5805/

database called the Resident Management System. Soibelman et al. [5] have addressed the need to develop additional frameworks that allow the development of data warehouses from complex construction unstructured data and to develop data modeling techniques to analyze common construction data types.

Various modeling techniques have been applied to the prediction of construction costs. They have usually focused on the use of numeric data to predict the project's outcome. Recent work has employed advanced data mining techniques to produce predictions. Son et al. [6] have developed a model using Principal Component Analysis and Support Vector Regression using 64 project definition variables to predict cost performance on building projects. Gritza and Labi [7] have applied econometric models to the analysis of highway project cost overruns. They found that, for a given project type and project duration, contracts of larger size or longer duration are generally more likely to incur cost overruns. Regression analysis and neural networks have also been applied to predicting construction costs [8–13]. Potentially, the addition of text data to various modeling techniques can enhance the predictions made by these models by covering more of the factors that can affect construction performance then can be derived from numeric data only.

2. Methods

2.1. Bidding data

Data for this analysis was collected from California Department of Transportation websites. Data from 1221 competitively bid highway projects were collected. The low bid, the completed project cost, and the numbers of bidders were collected. The percentage cost increase for each project was calculated as: Percentage Cost Increase = ((Completed Cost − Low Bid)/Low Bid) * 100.

Through experimentation, it was found that trimming large cost under run outliers from the data set produced better predictions. Therefore, 47 projects that were completed at a cost more than 25% lower than the original bid price were excluded from the analysis. The maximum cost overrun in the data set was 74% greater than the original low bid amount. Possibly, projects with very large cost under runs indicate a situation where the project scope was modified after the bid opening.

Each project was assigned to one of three cost overrun groups. Large cost overrun projects are categorized as having cost increases greater than 6%. Projects categorized as being completed near the original low bid had overruns between +6% and −3% under runs. Projects categorized as under run projects were all projects completed with an under run more than 3% less than the bid amount. The output of the model is a prediction of which of the three levels of overrun a project will have.

In addition to the numerical data, descriptive text was collected for each project. A short two to three sentence project description was obtained from a project summary included in the bid opening details. No location data was included (i.e. county or highway route). The text descriptions of the five largest project line items by dollar value were also collected. This data was added to include words in the analysis that describe the major work tasks on the project. These were the text describing standard unit price line items used by the California Department of Transportation. The text data for each project were saved as a continuous block of text. The project data collected varied widely in cost magnitude and type of construction. Some projects were maintenance projects while others were major rehabilitations or new construction. Table 1 shows an example of the input data.

2.2. Modeling software

The Rapid Miner software is a widely used data and text mining system [14]. The software incorporates powerful tools for data manipulation, data mining, and text mining. The Rapid Miner software allows experimentation with different types of data mining algorithms. In addition, the Rapid Miner software has the ability to manipulate and transform text into a format useable by data mining algorithms. All of the algorithms used in this research were originally developed for the Weka data mining software [15], and have been ported to the Rapid Miner system.

2.3. The modeling process

Various models were constructed that combined the text and numerical data to predict the level of cost overrun (or under run). Several different data mining algorithms were employed in the models with varying levels of success.

Fig. 1 illustrates the training process and testing processes. For each model type and model run 60% of the data were used for training (644 projects) and 40% was used for testing (430 projects). The first step in training is to split the data. The text data and numerical data are separated and processed separately. In the training process, the text data is submitted to text-mining algorithms to transform the text into a useable format and to provide data about the words and word pairs that are indicative of certain levels of cost overrun. The text for each project must be transformed into a numerical vector that is suitable for use with a data-mining algorithm. There are several steps that are necessary to transform the unstructured text for each project into a standardized numeric vector. These steps are tokenization, stopping, stemming, normalization and vector generation [16].

The output from the text processing is a very large sparse matrix with all of the words and word pairs as columns and projects as rows. Singular value decomposition (SVD) was used to reduce the text matrix to a smaller matrix of numerical data representing each project. This was necessary because the data mining models used can only accept numeric inputs and this transformation allows for faster computer processing.

The numerical variables are combined with the text data after the text data is processed and transformed into numeric variables. This combined data was submitted to a classification model. The models studied included Ripple Down Rules (Ridor), K-Star, Radial Basis Function (RBF) Neural Networks, and the Ensemble Stacking Method.

Table 1
Model input data.

Low bid	Number of bidders	Cost overrun level	Text
887859	3	3	INSTALL TRAFFIC SIGNAL SIGNAL AND LIGHTING ASPHALT CONCRETE (TYPE A) CLASS 2 AGGREGATE BASE 600 MM REINFORCED CONCRETE PIPE
42576814.6	4	3	BRIDGE CONSTRUCTION STRUCTURAL CONCRETE, BRIDGE TIME-RELATED OVERHEAD MOBILIZATION STRUCTURAL CONCRETE, BRIDGE FOOTING
1910418.36	4	3	RESURFACE ASPHALT CONCRETE RUBBERIZED ASPHALT CONCRETE (TYPE G) TRAFFIC CONTROL SYSTEM MOBILIZATION COLD PLANE ASPHALT CONCRETE PAVEMENT
1256988.37	4	3	UPGRADE PLANTING AND IRRIGATION PLANT (GROUP A) PLANT ESTABLISHMENT (LOCATION #1) (3 YEAR) PLANT ESTABLISHMENT (LOCATION #2) (1 YEAR) CLASS 2 AGGREGATE BASE
1996587.63	6	3	RESURFACE ROADWAY RUBBERIZED ASPHALT CONCRETE (TYPE G) REPLACE ASPHALT CONCRETE SURFACING COLD PLANE ASPHALT CONCRETE PAVEMENT TRAFFIC CONTROL SYSTEM

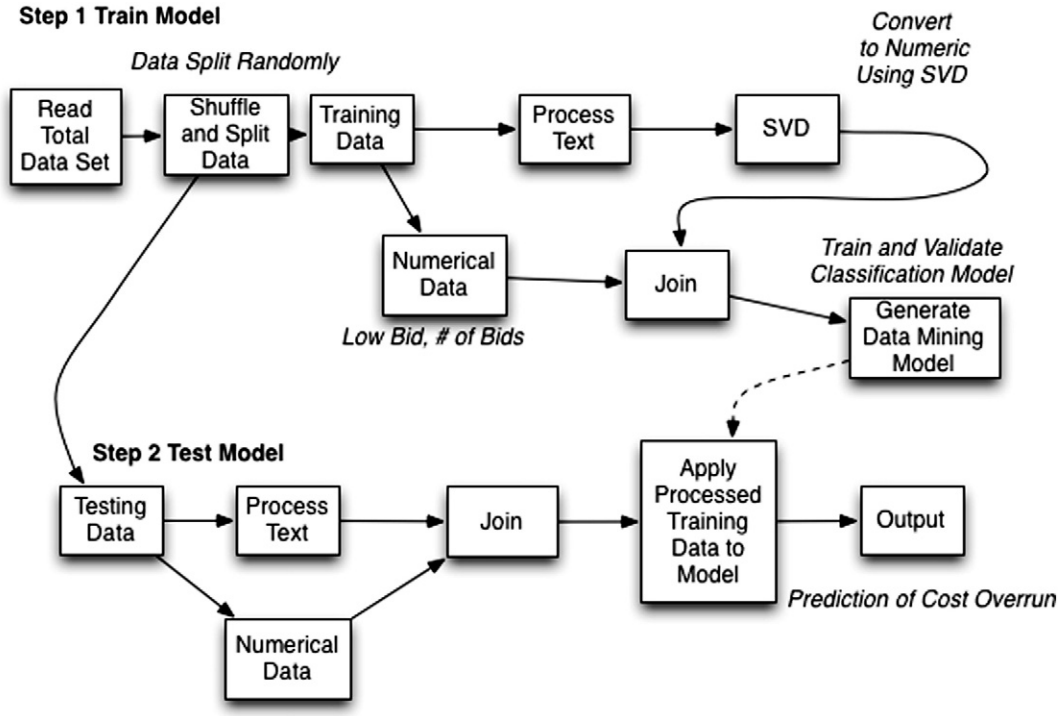

Fig. 1. The modeling process.

Additional details about these models are provided below in the Theory section.

The data were grouped into three levels of cost overrun for use with the classification algorithms. Projects with a large cost overrun are projects completed at a cost greater than 6% of the original low bid. Projects categorized as being completed near the original low bid had overruns less than 6% and projects with under runs as low as −3%. Projects categorized as under runs were all projects with under runs less than −3.

The arrangement of three levels of cost overrun was used because it produced better results than using a larger number of data groupings. Possibility with more data it would be possible to differentiate between a larger number of project groupings. It was also noted that to obtain reasonable predictions it was necessary to have a reasonable balance between the number of cases at each level of cost overrun. Through experimentation, it was found that designating projects with greater than 6% as high overruns provided enough high overrun cases for the classifiers to produce reasonable predictions that can differentiate between a high overrun project and a project completed near the original bid amount.

A bootstrapping technique where 70% of the training data were randomly selected to be training cases, and 30% of the training data were randomly selected to be test cases was used to improve the performance of the classification model. This procedure was repeated 10 times. The bootstrapping method is used when relatively small amounts of data are available to improve model performance. A separate testing set of data is then submitted to the trained model for classification.

The bottom portion of Fig. 1 illustrates step 2 of the modeling effort, the training process. The independent training set is submitted to the classification model trained in step one of the modeling process. The training set text is processed and then transformed into a numerical matrix using SVD. Once processed and combined with the numerical variables the data are submitted to the trained models and predictions are produced.

For each model type and model run 60% of the data were used for training and 40% was used for testing. Five runs using a different mixture of testing and training projects were made for each model using different random number seeds to shuffle the data. The same stream of random number seeds was used for each model type, so the

Table 2
Word stems associated with projects with large cost overruns.

Word	In documents	Total	High overrun project	Near low bid project	Under run
excav_asphalt	18	18	14	3	1
drainag	19	19	14	3	2
mobil_time	14	14	11	2	1
replac_bridg	14	14	10	4	0
cast_steel	7	10	8	2	0
steel_concret	7	10	8	2	0
overhead_structur	8	8	6	2	0
realign_roadwai	8	8	6	2	0
roadwai_bridg	8	8	6	2	0
temporari_signal	8	8	6	2	0
epoxi_coat	5	5	5	0	0
geosynthet	5	5	5	0	0
geosynthet_reinforc	5	5	5	0	0
reinforc_embank	4	5	5	0	0
temporari_fenc	4	5	5	0	0

Table 3
Word stems associated with projects completed near the low bid.

Word	In documents	Total	High overrun project	Near low bid project	Under run
barrier	100	194	48	122	24
sign_structur	37	50	11	37	2
sign_illumin	17	17	3	14	0
barrier_steel	15	19	3	15	1
steel_post	24	29	3	21	5
thrie_beam	24	38	9	26	3
median_barrier	43	43	6	30	7
sign_illumin	17	17	3	14	0
barrier_steel	15	19	3	15	1

predictions produced by the different classification models are directly comparable. An additional model using only numeric data as inputs was also tested. Because all of the models were trained and tested using the same data, the results are directly comparable.

3. Theory

3.1. Text mining

The purpose of text mining is to transform text into numeric attributes that can then be used in data mining algorithms. Text mining is often defined in the context of discovering previously unknown information that is implicit in the text but not immediately obvious [17]. In this case, we are attempting to extract information about the frequency of a word or word pair and the relationship of that frequency to cost overruns.

The following steps required are required to prepare text for further processing:

1. Uppercase letters were removed from the text.
2. The unstructured text is transformed into a sequence of tokens. Tokens can take on different forms. However, in this model the tokens were equivalent to single words.
3. Stemming. In this data transformation, related word tokens are normalized into a single form. For example, “walking” would be transformed to “walk” [17]. The Porter method of stemming was used. It is not required to use stemming when processing the text, but for this data stemming was found to increase the accuracy of the classifications.
4. Common words like “and” and “but” are removed.
5. Words that are less than four and longer then 35 characters long were removed from the word list. This is a user controlled setting in Rapid Miner. The lower bound is to keep common words that won't add to the prediction accuracy from appearing in the word list.
6. The “Generate n-gram terms” feature was used to allow for frequently occurring word pairs to be considered. In this processes, Rapid Miner has been set to allow two word terms to be entered in the term-document matrix that is generated for the text. In other words, word pairs can be combined to produce additional tokens. It is also possible to have higher order terms like word triplets. It was found that prediction results were not enhanced using the higher-level word groupings.

After these transformations are completed, a matrix with each token as a column is generated. Each row in the matrix represents a project. Each cell in the matrix contains the number of times a token occurs in a project's text description. For all algorithms except K-Star it was found that recording the term frequency, that is the number of times a token occurs in the text for a particular project, in the matrix produced the best results. However, for the K-Star algorithm it was found that a technique to modify the count by the perceived importance of the words provided more accurate results. The *tf–idf* formulation was used with the K-Star algorithm. The net result of this process is a positive score that replaces the simple frequency in the cell of the output matrix. The larger the score, the more important is the tokens expected value to the learning method [16].

3.2. Classification algorithms

Several different classification models were tested in this research. The following is a description of the classification methods used by the different models to classify project cost overruns:

- Ridor Rules. This algorithm automatically generates a set of classification rules from the input data. The Ridor algorithm is a machine learning technique that automatically generates rules from a data set [18]. The Ridor algorithm learns rules with exceptions by generating a default rule. The default or top-level rule is the class of the output that occurs most frequently. Then the algorithm uses incremental reduced-error pruning to find exceptions with the smallest error rate, finding the best exceptions for each exception and iterating [19].
- K-Star. This is a type of algorithm called a “lazy learner.” The K-Star algorithm is an instance based learning scheme developed by Cleary et al. [19]. Witten and Frank [20] describe the K-Star algorithm as a lazy classifier where the training instances are stored and are not employed until the classification time. They also state that K-Star is the nearest neighbor method with a generalized distance function. It was found through experimentation that the use of Support Vector Machines (SVMs) to weight the importance of the word tokens produced improved the K-Star predictions. SVM is a group of supervised learning methods that can be applied to classification or regression problems [21].
- Radial Basis Functions. A neural network algorithm that implements RBF as the learning mechanism. It uses the k-means clustering algorithm to provide the basis functions and learns a logistic regression.

Table 4
Word stems associated with under run projects.

Word	In documents	Total	High overrun project	Near low bid project	Under run
Binder	40	40	4	16	20
sand	9	10	2	0	8
asphalt_rubber	12	13	1	3	9
screen_medium	15	15	1	3	11
seal_coat	16	20	1	7	12
chip	21	21	2	6	13
emuls_polym	25	25	2	9	14

Table 5
Summary of model predictions.

Model	Accuracy	Class precision			Class recall		
		High overrun	Near	Under run	High overrun	Near	Run
Ridor	39.66%	42.95%	41.91%	9.74%	42.00%	51.10%	9.00%
K-Star	40.47%	40.18%	47.67%	20.72%	32.82%	55.37%	18.32%
RBF neural network	44.47%	46.71%	45.05%	2.22%	19.24%	82.35%	0.49%
Stacking	43.75%	44.32%	48.95%	18.33%	43.22%	56.52%	11.96%
Stacking (numeric only)	44.15%	39.29%	45.56%	0%	22.18%	80.78%	0%

Symmetric multivariate Gaussians are fit to the data from each cluster [22]. RBFs have been found to be useful for problems involving many variables, problems that are defined by large amounts of data, and the data are "scattered" in the problem domain [22]. It was found that an RBF model using two clusters produced more accurate predictions than an RBF model using three clusters.

- Stacking. The stacking technique creates an ensemble of classifiers, whose outputs are used as inputs to a second level meta-classifier to learn the mapping between the ensemble outputs and the actual correct classes [23]. An ensemble stacking model that combined the Ridor, K-Star, and RBF neural network models to produce enhanced results was applied to the California construction data. The JRip Rapid Miner function [19] was used as a meta-learner to generate rules to select from the output of the three classification models.

3.3. Support vector decomposition

SVD, a dimensionality reduction method, was used to transform the matrix of projects and terms into a smaller matrix of numeric values for each project. Saha et al. [24] state that SVD is often used for text mining applications because it transforms the existing feature space to a lower dimension. For this research, several thousand words and word pairs were identified. Therefore, a very large matrix is generated as the output of the text processing. SVD allowed the matrix to be transformed to a one or two column matrix.

4. Results

4.1. Word list analysis

After text processing occurs, it is possible to produce a list of words and word pairs with counts of their total frequency of occurrence and counts of when these word pairs occurred for each level of cost overrun. Table 2 shows a listing of words frequently associated with large cost overruns. It can be noted that the word stems "replac_bridg" "excav_asphalt", and "drainag" are frequently associated with cost overruns. Possibly these words are indicators of activities that are difficult to estimate or perform.

Table 3 shows a list of word stems that are usually associated with projects completed near the bid amount. Word stems like "barrier", "plant" and word pair stems like "steel_post" and "resurface_exist" indicate projects that are likely to be completed near the original bid amount. Table 4 shows a list of words that indicate projects that will have under runs. There are fewer clear examples of words indicating under runs than there are for projects with high cost overruns and those that are completed near the original bid amount.

4.2. Data mining output

Table 5 shows a summary of the predictions produced by the models. Each of the models was run five times. Each model run used a different mixture of training and testing cases. This was accomplished by shuffling the total data set using different random number generator seeds and then the data were split into training and testing sets. This insured that the training and testing sets used in each run were unique. The precision of the prediction represents the percentage of time a prediction made by the model is correct. The prediction recall represents the percentage a projects' actual level of cost overrun is correctly predicted. The average accuracy of the models ranged from 39.66% to 44.75%. However, overall prediction accuracy was reduced by the poor performance in predicting projects with significant under runs.

The results indicate that the stacking ensemble method employing a combination of text and numeric data yielded some promising prediction results for identifying projects that have high cost overruns and projects that will be completed near the original low bid. The stacking ensemble model had an average accuracy of 43.72%. The model performed best predicting large cost overruns and projects completed near the low bid amount. Prediction accuracy was low for the under run projects. The stacking model had high levels of class recall for high cost overruns and projects completed near the low bid. Table 6 shows the stacking model results for each of the five testing runs made. The class recall for "large cost overrun" projects varied from 37.93% to 53.07%, for the five testing runs. The precisions for the "large cost overrun" class varied from 39.18% to 51.68%. The "near" class had recalls varying from 50.72% to 68.60%. Precisions for the "near" class varied from 44.35% to 52.75%. Table 7 shows the confusion matrix for the most accurate stacking run.

The RBF model had the highest prediction accuracy of 44.47%. However the RBF model is unable to predict under runs and does a poor job predicting large cost overruns. This model gave highly accurate prediction for projects completed near the low bid amount. The K-Star model provided the most accurate predictions of projects with under runs, yet the accuracy of the prediction of under runs is still low.

Table 6
Detailed stacking model results.

Run	Accuracy	Class precision			Class recall		
		High cost overrun	Near	Under run	High cost overrun	Near	Under run
Run 1	47.97%	51.68%	49.82%	17.14%	41.18%	68.60%	8.00%
Run 2	44.56%	45.21%	49.43%	22.58%	37.93%	61.14%	16.67%
Run 3	45.42%	46.34%	52.74%	19.05%	53.07%	50.72%	14.81%
Run 4	40.30%	39.20%	48.42%	18.06%	39.20%	51.20%	15.48%
Run 5	40.51%	39.18%	44.35%	14.81%	44.71%	50.93%	4.82%
Average	43.752	44.322	48.952	18.328	43.218	56.518	11.956

Table 7
Confusion matrix for the best stacking model run.

	Actual high cost overrun	Actual near original bid	Actual under run	Class prediction
High cost overrun prediction	77	40	23	51.68%
Near original bid prediction	97	142	46	49.82%
Under run prediction	13	16	6	17.14%
Class recall	41.18%	68.60%	8.00%	

An additional model using only numeric inputs was tested. The result for this model is also shown in Table 5. The numeric data, consisting of the low bid amount and the number of bidders, was input to the same combination of classifiers using stacking. Although the model that was produced had a similar overall accuracy to the combined text and data-stacking model, it produced poorer recall and precision results for each level of cost overrun. This indicates that there is useful information contained in the text that can contribute to improved predictions of cost overrun.

5. Discussion

The text mining analysis indicated that there are words and word pairs that can be associated with different levels of cost overruns, particularly for projects completed near the low bid amount and for projects with large cost overruns. The model predictions indicate that text descriptions about a project combined with numerical data describing the project contain useful information that can be used to produce predictions about a competitively bid projects' cost performance with some success. The ensemble stacking models developed are best able to predict projects that were completed near the original low bid and projects that were completed with large cost overruns. Some prediction model runs were able to predict large cost overruns with precision and recall greater than 50%. The use of ensemble learning, where the results of several classification algorithms are combined, improved the accuracy and recall of the predictions. This indicates that ensemble learners are useful when applied in the area of construction cost predictions. Collection of additional data would also probably improve the predictions because it was noted that there were some words that occurred in the testing data that did not occur in the training data.

A model using stacking with only numerical inputs had a similar overall accuracy to the combined data and text stacking model, but it was found that it had significantly lower levels of precision and recall. This indicates that the inclusion of the text data provides additional information about projects' expected level of overrun.

An additional consideration is the amount of text that is available to describe the project. The text used in this analysis included only a one or two sentence description of the project plus the line item text. Perhaps, more detailed project descriptions that better identify projects' unique aspects could yield even higher percentages of correct predictions. The data used in this analysis did not have any geographical location data. Perhaps including location data that allow differentiation of rural and urban projects can increase the prediction accuracy.

It is widely recognized that many factors other than a description of the work to be performed affect the completed project cost. Various internal and external factors affect project costs such as poor execution by the contractor design errors, scope change, and unforeseen site conditions [25]. These types of problems are not directly reflected in the text describing a construction projects' scope and work tasks. However, it is possible that a project that is more difficult to construct as conveyed in the text description would be a project prone to cost increases if the design is incomplete, or the contractor does not manage the project effectively. Potentially the accuracy of the models can be improved by adding information about the contractor conducting the project. For example, the name of the contractor can be added to the text input, and contractor performance metrics can be added as numeric input. Data of this type is becoming increasingly available as project owners like DOTs make more data available online.

6. Conclusions

Construction project text descriptions can be usefully combined with numerical data to improve prediction of cost overruns. Construction projects with under runs were not accurately predicted by the data mining models. Using the ensemble data-mining technique of stacking prediction accuracies of greater than 50% were noted for both projects with large cost overruns and projects completed near the low bid amount. Future research in this area should include the addition of location and complexity information, and measures of the prime contractor performance. Additionally, it will be interesting to study if the data collected for California is generalizable to other jurisdictions or if each government agency requires a model to be built using its specific terminology.

References

[1] M.A. Hearst, Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, Untangling Text Data Mining, 1999, pp. 3–10.
[2] T. Simonite, MIT Tech. Rev, http://www.technologyreview.com/news/510191/software-predicts-tomorrows-news-by-analyzing-todays-and-yesterdays/ June 30 2013.
[3] C.H. Caldas, L. Soibelman, Automating hierarchical document classification for construction management information systems, Autom. Constr. 12 (4) (2003) 395–406.
[4] L. Soibelman, H. Kim, Data preparation process for construction knowledge generation through knowledge discovery in databases, J. Comput. Civil Eng. 16 (1) (2002) 39–48.
[5] L. Soibelman, J. Wu, C. Caldas, I. Brilakis, K.-Y. Lin, Management and analysis of unstructured construction data types, Adv. Eng. Inform. 22 (1) (2008) 15–27.
[6] J. Zhang, N.M. El-Gohary, Information transformation and automated reasoning for automated compliance checking in construction, Proceedings of the ASCE International Workshop on Computing in Civil Engineering, Los Angles, CA, 2013, pp. 701–708.
[7] H. Son, C. Kim, C. Kim, Hybrid principal component analysis and support vector machine model for predicting the cost performance of commercial building projects using pre-project planning variables, Autom. Constr. 27 (2012) 60–66.
[8] C. Gkritska, S.S. Labi, Estimating cost discrepancies in highway contracts: multistep econometric approach, J. Constr. Eng. 134 (12) (2008) 953–962.
[9] S.M. Trost, G.D. Oberlender, Predicting accuracy of early cost estimates using factor analysis and multivariate regression, J. Constr. Eng. 129 (2) (2003) 198–204.
[10] K.M. Nassar, W.M. Nassar, M.Y. Hegab, Evaluating cost overruns of asphalt paving project using statistical process control methods, J. Constr. Eng. 131 (11) (2005) 1173–1178.
[11] K. Petroutsatou, E. Georgopoulos, S. Lambropoulos, J.P. Pantouvakis, Early cost estimating of road tunnel construction using neural networks, J. Constr. Eng. 138 (6) (2011) 679–687.
[12] T.P. Williams, Bidding ratios to predict highway project costs, Eng. Constr. Archit. Manag. 12 (1) (2005) 38–51.
[13] C.G. Wilmot, G. Cheng, Estimating future highway construction costs, J. Constr. Eng. 129 (3) (2003) 272–279.
[14] I. Mierswa, et al., Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Rapid prototyping for complex data mining tasks, Yale, 2006. pp. 935–40.
[15] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.
[16] S.M. Weiss, N. Indurkhya, T. Zhang, Fundamentals of Predictive Text Mining, Springer-Verlag, New York., 2010
[17] G. Miner, J. Elder, T. Hill, D. Delen, A. Fast, Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications, Academic Press, Waltham, MA, 2012.
[18] B.R. Gaines, P. Compton, 1995, Induction of ripple-down rules applied to modeling large databases, J. Intell. Inf. Syst. 5 (3) (1995) 211–228.

[19] J.G. Cleary, L.E. Trigg K*, An instance-based learner using an entropic distance measure. In Proceedings of 12th International Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, 1995.
[20] 1995. I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, San Francisco, 2005.
[21] O. Ivanciuc, Applications of support vector machines in chemistry, Rev. Comput. Chem. 2007 (23) (2007) 291.
[22] M.D. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge University Press, Cambridge, 2003.
[23] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45.
[24] S.K. Saha, P. Mitra, S. Sarkar, A comparative study on feature reduction approaches in Hindi and Bengali named entity recognition, Knowl. Based Syst. 27 (2012) 322–332.
[25] J.S. Shane, K.R. Molenaar, S. Anderson, C. Schexnayder, Construction project cost escalation factors, J. Manag. Eng. 25 (4) (2009) 221–229