Predicting the Effectiveness of Bike Classifieds

Lawrence Xing
Stanford University
lxing@stanford.edu

Abstract

We apply text classification techniques to the problem of predicting the eventual sale of classified bike advertisements. A combination of Naïve Bayes, support vector machines, and support vector regression performs reasonably well on bag-of-words and tf-idf feature representations of the data. The use of information gain thresholds to prune out less relevant term features yields a significant improvement in accuracy for the two vector approaches. Finally, we demonstrate that support vector regression is a novel and effective classification method for text categorization.

1. Introduction

Online advertising is a major source of monetization on the internet. It is the primary source of revenue for Google, and numerous other websites host ads alongside their content. With increasing amounts of money being devoted to advertisement, the topic is a prime candidate for optimization.

One common line of research is personalized advertising, which matches of specific products to the customers most likely to buy them [1]. While this can certainly be effective, it does not reveal what makes an ad inherently appealing or unappealing. In this paper, we investigate methods for predicting the appeal of an advertisement broadcast to a set audience.

To narrow the scope of investigation, we focused on a specific set of advertisements: classified bike listings from the website craigslist.org. We chose this data set for multiple reasons. First, it was very simple to identify the sale of an advertisement by continuously polling the listing until it was taken down. Secondly, we hoped to extract a rich feature set from bike classifieds based on price, devaluation, number of images, componentry, etc. Finally, the author enjoys cycling.

1.1. Background Work

There has been little research into machine learning based on the rich feature sets described above. Part of this lies in the specificity of the problem domain, and another part stems from the absurdity of trying to optimize an ad with respect to its price and devaluation from MSRP.

However, there is a substantial foundation for approaching advertisement success prediction as a text classification problem. Sahami et al. describe a Bayesian approach to document classification using basic probabilistic methods [2]. Additionally, the field of information retrieval (IR) has developed techniques for featurization of documents, including term frequency, document frequency, and term frequency-inverse document frequency [3]. Using the last of these feature representations, it has been shown that support vector machines (SVMs) can be used to effectively categorize text [4].

Smola and Vapnik also separately describe a regression technique whose derivation is similar to that of SVMs called support vector regression (SVR) [5] [6]. We attempt to apply this to text classification in what appears to be a novel approach.

Finally, to combat the high dimensionality of traditional IR document representations, we adopt a thresholding technique based on the information gain metric as recommended by Joachims [4] and described by Yang and Pedersen [7].

2. Methodology and Results

2.1. Dataset

Craigslist offers an enormous variety of listings, ranging from appliances to housing. For this investigation, we limited the range of posting data to road bikes with a list price of over $500 in the San Francisco Bay area. Section 1 describes our rationale for this subset. Additionally, the price bound was chosen to observe a relatively expensive market that would warrant buyer scrutiny.

The conventional barrier between an online ad and the corresponding purchase is spatial; the two are hosted at different locations, requiring server-side knowledge to associate the two. On Craigslist, the separation is temporal. It is possible to identify the sale of a bike listing when the listing is taken down from the website. This in turn requires tracking of the posting throughout its lifetime, from initial submission to removal.

To collect advertisement data, we polled the first two pages of Craigslist's bike listing section for new postings

at a frequency of once every two days. Once identified, we accessed each posting's individual listing page to extract the title, body text, price, posting date, and images. In addition to these initialization scrapes, we also polled stored ads with the same frequency to check of they had been removed (and sold). Because we could not recover the exact sale time associated with a removed ad, we assigned each a coarse sale time bucketed within two days of when it was first discovered missing.

During ad initialization, we excluded advertisements older than one week at the time of scraping. The rationale for this was that these ads were inherently biased by virtue of their age. Other, already sold listings within the same time cohort would have been erroneously excluded from data collection because they would have been taken down at the time of scraping.

One complication was that Craigslist does not have a public API for its data, and in fact actively discourages high-volume scraping of its website due to bandwidth considerations [1]. Offenders can be IP-blocked from the website. To combat this, we conservatively inserted a delay in between the fetching and polling of each individual posting. Additionally, we limited the total number of queries per day. This limited the overall size of the data set; we collected 600 classified ads.

In practice, we observed that a posting that does not sell within four weeks will not sell at all because it becomes buried from view by newer postings. Thus, we assigned a binary class label to our dataset based on whether or not it sold within one month.

2.2. Data Processing and Feature Selection

Initially, we intended to construct a rich feature set for each advertisement based on the price, the presence of images in the ad, the devaluation from MSRP, post time, and any mention of specialized componentry. Initial experiments on this featurization yielded poor results, so we discarded it in favor of an IR-grounded approach.

For each ad title and body, we performed the following transformations:

- Strip HTML tags using the web framework Ruby on Rails' ActionView module
- Remove stop words using the NLTK corpus
- Porter stem remaining words
- Tag and replace numeric tokens

We stripped HTML data to sanitize useless metadata (although ad formatting could be a potential avenue of exploration for another time). Stop words were removed to prune non-informative words like "the" and "a", and the remaining words were Porter stemmed to regularize variations of certain root forms. Numeric tokens were rewritten for the same reason, to regularize information like "54cm" and "52cm" into the common root "<num>cm". For the last preprocessing step, we used a custom regular-expression based tagger for numeric expressions only. We deliberately eschewed the NLTK tagger due to its large set of tags, which we felt were too specific for the task at hand.

Finally, we computed the tf-idf vector representation of each document. Suppose the vocabulary of the entire corpus is V. Then each document can be represented based on term frequencies in a $|V|$-dimensional vector, where the i^{th} element is the number of occurrences of the i^{th} term in the document. The resulting vector contains a bag-of-words representation for the document. However, this approach tends to overweight common terms which do not provide much discriminatory information between documents. Thus it is desirable to weight each term element inversely to the number of documents in which the term appears (the document frequency). The final term frequency-inverse document frequency (tf-idf) for term *t* in document *d* from a dataset of documents *D* is then

$$tfidf_{t,d} = \mathrm{f}(\mathrm{t},\mathrm{d}) * \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

Once the tf-idf scores were computed, we additionally normalized each document vector to unit length to decrease the weight of longer documents.

2.3. Naïve Bayes

One of the simplest approaches to text classification is Naïve Bayes. This technique generatively models the probability of a document conditioned on a class, then selects the class which produces the highest *maximum a posteriori* (MAP) estimate. The generative model is augmented by an assumption of conditional independence between individual terms in the document. If an advertisement is represented as a series of terms $(t_1, t_2, \cdots, t_m)$, then the model's approximation for the likelihood of a classification is

$$p(sold|t_1, t_2, \cdots) \propto \left[\prod_{i=1}^{m} p(t_i|sold)\right] p(sold)$$

The parameters $p(sold)$ and $p(t_i|sold)$ are maximum likelihood estimates based on the training data. The estimate we used above is a slight variation of the original Naïve Bayes model called the multinomial event model. This model considers the likelihood of each term in the document, as opposed to the likelihood of each distinct term.

In addition to implementing the Naïve Bayes model, we

[1] http://stackoverflow.com/questions/237124/how-do-craigslist-mashups-get-data

also applied Laplace smoothing with a smoothing parameter of 1 to place a prior on unseen terms.

To test Naïve Bayes, we used leave-one-out cross validation. We ran it separately on both advertisement titles and body texts.

2.4. SVM

Using the tf-idf vector representation of each advertisement, the problem reduces to binary categorization of at high-dimensional data set. SVMs are appropriate for this, and it has been demonstrated that they work on tf-idf features [4].

We trained an SVM to classify both advertisement titles and body texts. We found that L1-regularized SVM with a linear kernel performed best. The Gaussian radial basis function, poly-2, and poly-3 kernels were tried did not perform as well. The loss parameter *C* was experimentally determined by iterating over the logspace from 0.001 to 1000, and reached an optimal value at 100.

For computational reasons, we only performed 10-fold cross validation to test SVM.

2.5. Support Vector Regression

A popular approach to text classification is logistic regression, commonly used with a Bayesian prior on the class distributions. We were interested in experimenting with an alternative method called support vector regression (SVR), which has not to our knowledge been used before for text categorization.

As a variant of regression, SVR attempts to fit a function that directly maps input vectors to output labels. With a linear kernel, this function *f(x)* takes the form

$$f(x) = \langle w, x \rangle + b$$

where w and b are the parameters of the SVR model. Instead of optimizing an objective function of total mean-squared error like logistic regression, SVR attempts to confine all error within the smallest possible threshold. In this sense it is the opposite of SVM, which tries to confine functional margins outside of the largest possible threshold in order to separate data.

SVR is typically solved by posing the equivalent problem of minimizing w while holding the error threshold constant at some ϵ. With regularization to allow for soft margins, the problem is to minimize

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\gamma_i$$

subject to

$$|y_i - \langle w, x_i \rangle - b| \leq \varepsilon + \gamma_i$$
$$\gamma_i \geq 0$$

This problem statement is remarkably similar to that of SVM. Indeed, the resulting solution can be represented as the inner product of the input with a subset of the training data; hence the term "support vector" regression.

We trained an L1-regularized SVR on both advertisement titles and body texts using only a linear kernel. As with SVM, we experimentally set the loss parameter *C* to be 100. To test the data, we performed leave-one-out cross validation.

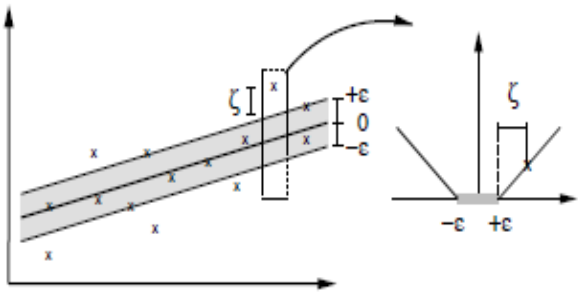

Figure 1. Graphical representation of SVR. The regression bounds all points within the smallest possible margin, penalizing those that fall outside [5].

2.6. Results

We adopted the IR metrics of precision and recall to quantify the accuracy of each technique. Precision is defined as the number of true positives over total positives, and measures how much of the retrieved data is appropriate. Recall is defined as the number of true positives over true positives over false negatives, and measures how much of the data that should have been retrieved was actually positively categorized. In addition, the F-score of a technique is the harmonic mean of the precision and recall.

In this case, we chose sold ads to be positives and unsold ads to be negatives. However, there was no particular reason to fixate on identifying sold ads, so we used overall prediction accuracy as the final metric, with precision and recall as additional descriptors.

	Precision	Recall	F-score	Accuracy
Random	0.390	0.542	0.454	0.487
NB Title	0.500	0.381	0.433	0.607
NB Body	0.462	0.458	0.460	0.577
SVM Title	0.487	0.475	0.481	0.600
SVM Body	0.467	0.364	0.410	0.587
SVR Title	0.520	0.542	0.531	0.623
SVR Body	0.518	0.364	0.428	0.617

Table 1. Test accuracy for baseline, Naïve Bayes, SVM, and SVR on both advertisement titles and body texts.

Figure 2. Training and test accuracy of all three methods with respect to fraction of data set used.

The test accuracy of all three methods was around 60%, a reasonable improvement over a baseline of random guessing. Of the three techniques, SVR performed the best in all four categories of precision, recall, F-score, and accuracy. This appears to validate the use of support vector regression in text classification. Training and prediction using ad titles consistently outperformed learning using advertisement body text. One explanation for this is that the title feature vectors were considerably shorter due to a smaller vocabulary – 591 terms as opposed to 4290.

While the models performed acceptably on the test data, they all exhibited an exceptionally high accuracy rate on the training data. To determine if the test and training errors would converge over time, we repeated the classification experiments using reduced data sets consisting of 40%, 60%, and 80% of the full data set. This performed no better (Figure 2); training error consistently hovered near zero, indicating a severe overfit.

This phenomenon was obvious in hindsight. Our data set size was only 600 advertisements, while the bag-of-words feature representations for ad titles and bodies used 597 and 4290 dimensions respectively. Given the low amount of training data relative to the features, we should have expected a high amount of variance.

To address this problem, we chose to reduce the number of features in the tf-idf space. The alternative of gathering more data was risky due to Craigslist's IP-blocking of high-volume scrapers. Unlike rich feature sets where each feature has a semantic interpretation, there is no obvious method for discarding elements from tf-idf vectors. Instead we turned to a mathematical approach.

2.7. Information Gain for Feature Pruning

The entropy of a distribution P over class labels c_1, $c_2, \cdots, c_m$ is defined as

$$H\big(P(c)\big) = -\sum_{i=1}^{m} P(c_i) * \log P(c_i)$$

Intuitively, a dataset with balanced classes will produce the highest entropy because both terms of the summation will be moderate, while a skewed dataset will have low entropy because one term will be close to zero. Suppose a binary classification is made based on some term *t*. Then define $H\big(P(c|t)\big)$ to be average entropy of the two sets classified by *t*, weighted by the size of each set. If classification based on *t* alone separates classes well, then the resulting entropy $H\big(P(c|t)\big)$ should be lower than the original $H\big(P(c)\big)$. The information gain of *t* is thus defined as $H\big(P(c)\big) - H\big(P(c|t)\big)$.

We used this metric to prune the training feature sets by only retaining elements of the tf-idf vectors with the highest information gain; that is, those terms that would separate sold and unsold classifieds the best. Joachims recommends using a fixed threshold [4], but with no baseline of information gain we instead varied the number of retained features from 1 to the original feature set size.

With experimentation, we determined the optimal feature set size to be 45/45 out of 597 original terms for title analysis, and 591/3020 out of 4290 for body text analysis for SVR/SVM respectively. We used these pruned feature sets to perform SVM and SVR again with the same parameters.

2.8. Results Using Pruned Feature Sets

	Precision	Recall	F-score	Accuracy
SVM Title	0.846	0.466	0.601	0.757
SVM Body	0.776	0.703	0.738	0.803
SVR Title	0.878	0.551	0.677	0.793
SVR Body	0.854	0.695	0.766	0.833

Table 2. Test accuracy for SVM, and SVR on both advertisement titles and body texts using pruned tf-idf vectors.

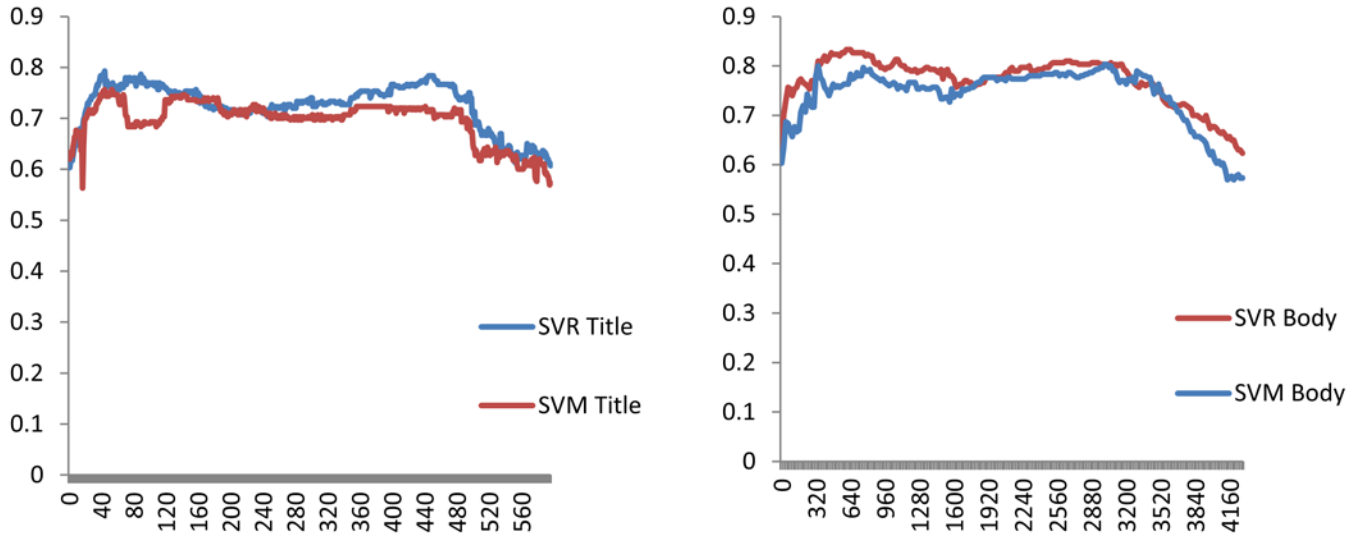

Figure 3. Test accuracy of SVR and SVM on title and body tf-idf representations, with respect to number of features included in the pruned vector.

The results of SVM and SVR with pruned feature sets were significantly improved over use of the full feature sets. Notably, both methods performed better operating on the body text than the title text. These two observations are consistent with the hypothesis that overfitting was caused by a high dimensionality. In the latter case, we would expect that as the body tf-idf vector was reduced to a similar size as the original title tf-idf vector, the comparative advantage of title analysis' low dimensionality would be removed.

We also observed that feature pruning seemed to follow a general trend of reduced accuracy with an extremely low or high number of features, and the highest accuracy somewhere in the middle. This observation held true for all four combinations of SVM/SVR and title/body (Figure 3). This illustrates the traditional bias-variance tradeoff in model feature selection: with very few features, our classifiers tended to underfit, while the full feature set induced an overfit.

Information gain weighting allowed us to recover the most entropy-reducing terms in the tf-idf weightings. Out of interest, we present them here:

- Title terms: sst redlin almost japan clean
- Body terms: bent item bicycl benicia durabl

3. Conclusion

Our results demonstrate that the appeal of classified advertisements can largely be predicted by their raw content in addition to their presentation. We also show that with the best classifiers, the body text is generally a stronger predictor of success than the title.

Joachims' use of information gain thresholding for feature selection proved to be an effective method for reducing dimensionality in a conservative fashion, and we effectively used it to compensate for a high dimensionality-to-dataset ratio. Finally, we have shown that support vector regression is a valid method for text classification and can in some cases outperform SVM and multinomial Naïve Bayes.

We would like to acknowledge Andrew Ng and the CS229 course staff for their teaching, and LIBSVM, LIBLINEAR, Ruby on Rails, and the NLTK as invaluable tools for this study. We wish to disacknowledge Craigslist for their anti-data mining practices.

References

[1] J. Shanahan. Digital Advertising: Going from broadcast to personalized advertising. *NIPS 2010 Workshop: Machine Learning in Online Advertising*, 2010.

[2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian aproach to filtering junk e-mail. *Learning for Text Categorization – Papers from the AAAI Workshop*, 1998.

[3] C. Manning. Introduction to Information Retrieval, 2008.

[4] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. *Proc. Of the European Conference on Machine Learning (ECML)*, 1998.

[5] A. Smola. A tutorial on support vector regression. *NeuroCOLT2 Technical Report Series, NC2-TR-1998-030*, 1993.

[6] H. Drucker, C. Burges, L. Kaufman, A.Smola, and V. Vapnik. Support Vector Regression Machines. *Advances in Neural Information Processing Systems 9, NIPS*, 1996.

[7] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. *International Conference on Machine Learning (ICML)*, 1997.