Data Mining for Customized Critical Fractile Solutions

Shawn O'Neil
soneil@cse.nd.edu

ABSTRACT

We apply data mining techniques to aid in the prediction of future demands for short life-cycle products with the goal of maximizing profits, commonly known as the newsvendor problem. In particular, we apply cost sensitive text classification techniques to a web based analogue of the problem's namesake: rather than predicting the sales demand for newspapers, we attempt to predict the number of comments a story summary will engender on the social news aggregation site `slashdot.org`.

1. INTRODUCTION

In the newsvendor problem, we face the problem of predicting what the demand will be for some product, so that we can order/manufacture the appropriate amount beforehand. If we order too few items, we lose profits on unmet demand. If we order too many, we lose profit on wasted inventory. The items in question are presumed to be perishable: items ordered for one selling period cannot be sold in a later period. If c is the cost per item to order, and r is the selling price of each item, ordering x units gives a net profit of $r \cdot \min\{x, demand\} - x \cdot c$.

The newsvendor problem is interesting when applied to situations where 1) demand is highly uncertain, and 2) the amount to be ordered needs to be decided well in advance of the selling season, making estimations based on initial demand observations impossible. Many products (often called *short life-cycle products*) have these properties, such as electronics, fashion items, some vaccines, and of course newspapers [8, 3, 2].

Traditionally, solutions to the problem make a stochastic assumption about the demands in some way. For example, it is often assumed that the demands will be drawn from some known probability distribution. In this case, an order quantity is decided upon which maximizes the expected profit [5]. While these approaches are frequently too optimistic in their assumptions that the estimated distribution will be correct [10], in other cases they can perform suitably well.

More recently, researchers have used worst case approaches to give guarantees on profit or other measures of success under less strict assumptions. For example, one can assume only a lower and upper bound on the demand range and derive a solution which minimizes the maximum regret (defined as the difference between the optimal profit and the actual profit made) [10]. These "mini-max" solutions are often too pessimistic to be useful in practice, however.

In this paper, we consider the problem of predicting demands in a data mining context. In theory, there should be some correlation between properties of the products being sold and the demand for them. Specifically, we show that if enough historical data is available, this correlation can be exploited to increase profits over the traditional stochastic solutions.

While newspapers are a prototypical short life-cycle product, data about daily newspaper sales is difficult to obtain and associate with the contents of the paper itself. In the absence of any actual industry newsvendor data set, we instead focus on an online equivalent. Summaries of news articles which appear on the front page of news aggregation site `slashdot.org` will serve as our papers, and the number of comments a summary engenders will serve as the demand for that summary. The product "features" are the words present in the summary, represented in bag of words format.

While these "products" are not strictly short life-cycle products, they have some of the requisite properties, such as a relatively large variance in demand (see Section 4) and a short "selling season." (Few comments are posted to a summary after the first day, when it leaves the front page.) In this setting, we consider the order cost per item c as \$1 per item, and the resale value r as \$4 per item.

2. RELATED WORK

Because the features of our products in this case are bag of words representations of news article summaries, the data mining problem we are considering is very similar to that of the well known text classification problem. In text classification, one or more labels are associated with each document in the corpus, and the goal is to predict which labels would be assigned to future test documents. Just as in this setting, online news articles are often the focus of such classification

[6].

Dimensionality reduction is an important topic in text mining applications; because the presence or absence (or even frequency count) of every term in the corpus represents a feature, the number of features can easily grow into the hundreds of thousands. Yang and Pedersen compared various feature selection methods for text classification, including document frequency thresholding, information gain, mutual information, χ^2 criterion, and term strength [13]. When tested with the k-Nearest Neighbor algorithm and Linear Least Squares Fit mapping (both very strong text classification approaches [12]), information gain and the χ^2 criterion proved most effective.

The origins of the newsvendor problem can be traced as far back as Edgeworth's 1888 paper [4], in which the author considers how much money a bank should keep on reserve to satisfy customer withdrawal demands, with high probability. If the demand distribution and the first two moments are assumed known (normal, log-normal, and Poisson are common), then it can be shown that expected profit is maximized by ordering x such that $\phi(x) = (r - c)/r$. ϕ is the cumulative density function of the distribution. This is the well known *critical fractile* solution. The book by Porteus gives a useful overview [7].

When only the mean and standard deviation are known (but not the distribution type), Scarf's results give the optimal order quantity which maximizes the expected profit assuming the worst case distribution with those two moments [9]. However, for this paper we'll compare the profits our models achieve to those achieved by critical fractile, since as we'll see in Section 4 the demand distribution for this data set is fit well by a log-normal distribution.

3. NEWSVENDOR AND THE CRITICAL FRACTILE SOLUTION

As we've just seen, the critical fractile solution (which performs very well when the demands actually follow the stochastic assumptions assumed) prescribes ordering x units such that $\phi(x) = (r-c)/r$. For a visual representation, see Figure 1.

It is interesting to note that the critical fractile solution does *not* prescribe ordering the average demand, or even the median (unless $r = 2c$), because of the unbalanced cost of over ordering versus under ordering.

While the critical fractile solution orders using only the overall demand distribution, it may be the case that features of each product in question can indicate a different distribution each time. In particular, suppose that we discretize the output space into equal width bins. If we then apply any classification algorithm which results in a probability vector over the class space (such as Naive Bayes or k-Nearest Neighbors), we can interpret this probability vector as a "customized" distribution for this product and apply critical fractile.

For an example, consider the situation represented in Figure 2. In this case, our training data suggests that the maximum demand we are likely to see is 200 units. We've accordingly discretized the demand into 10 bins. After training a classifier on the training instances (in which the output class is also discretized), the test instance is given and the model predicts bin 1 with probability 0.05, bin 2 with probability 0.1, and so on. Rather than simply order an amount corresponding to the most likely bin, we order the $(r-c)/r^{th}$ percentile of this non-continuous density function. Of course, we must remember to normalize so that the total area under the curve is 1.

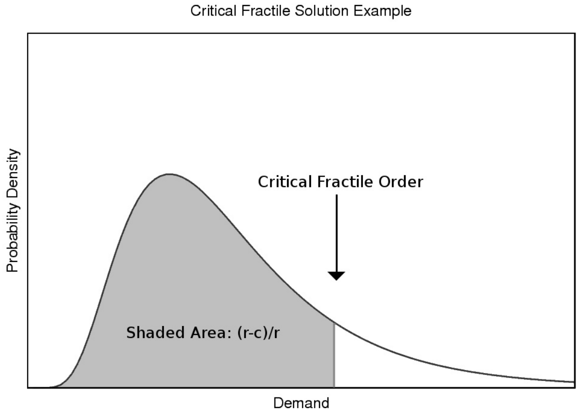

Figure 1: Representation of the critical fractile solution when demands are drawn from a log-normal distribution, where per item cost is c and per item profit is r.

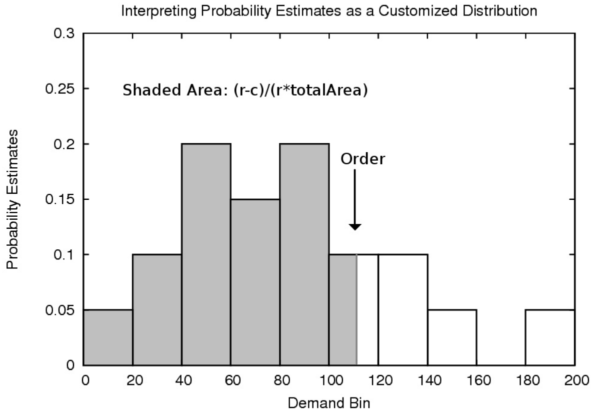

Figure 2: After discretizing the output space, we interpret a probability vector over the classes as a "customized" distribution for this test instance, and apply critical fractile.

4. DATA, PREPROCESSING, AND PROFIT BASELINES

As an analogue to real-world newspapers, we've collected summaries of news articles which appeared on the front page of `slashdot.org`, as well as the number of comments each summary received. We collected six years worth of data, from the first day of 2002 to the last day of 2007. Story summaries also included the title, the name of the editor who posted the story, and the category the story was posted under.

The data was separated into three categories: years 2002-2005 for training data (27,429 instances), 2006 for validation data (7,185 instances), and 2007 for testing data (7210 instances). During data collection, all words were lowercased and any non-alphanumeric characters were dropped. Over all, this corpus represents 116,478 unique words. Stemming these terms using the Libstemmer library [1] resulted in 95,168 root terms. 83,804 of these were present in the 2002-2006 data.

To reduce the dimensionality of this data set, we ranked the stemmed terms which appeared in the 2002-2006 data according to the generalized information gain formula given by Yang and Pedersen [13]:

Definition 4.1. *(Due to Yang and Pedersen) Let $\{c_i\}_{i=1}^m$ denote the set of categories in the target space, and $Pr(t)$ be the probability that term t appears at least once in any document. The information gain of t is defined to be*

$$\begin{aligned} G(t) = &-\sum_{i=1}^{m} Pr(c_i)\log(Pr(c_i)) \\ &+ Pr(t)\sum_{i=1}^{m} Pr(c_i|t)\log(Pr(c_i|t)) \\ &+ Pr(\bar{t})\sum_{i=1}^{m} Pr(c_i|\bar{t})\log(Pr(c_i|\bar{t})) \ . \end{aligned}$$

In order to compute the information gain of terms in this manner, it was necessary to discretize the output space. First, the top 0.5% of demands (from 1,454 to 5,687 comments) were placed in their own class. Next, the remaining demand range (0 to 1,453 comments) was discretized into 19 equal width bins, for a total of 20 classes for information gain computation. For some interesting statistics regarding the relative information gain of terms appearing in Slashdot summaries, see Appendix A.

Figure 3 shows the distribution of demand for the 2002-2006 data, plotted along with a fitted log-normal curve. The log-normal distribution is fitted using maximum likelihood estimation. The 95% confidence intervals for μ and σ (the mean of the standard deviation of the associated normal) are [5.625,5.638] and [0.644,0.654], respectively.

Using these parameter estimates, the critical fractile solution for $r = 4$ and $c = 1$ is to order 432.3 units every period. This could be considered a "realistic" baseline order quantity for the 2007 test data. On the other hand, we can also consider a better performing "perfect" critical fractile as a baseline for comparison, wherein the order quantity for 2007 is found by fitting a curve to the 2007 data itself. Figure 4 shows the critical fractile orders for different sets of year ranges. The first two rows correspond to "realistic" orders for 2006 and 2007, respectively, and the third and fourth rows "perfect" orders.

Using these order quantities, Figure 4 shows the profit of critical fractile on the validation (2006) and test (2007) years, under realistic and perfect assumptions. For the validation and test phases of our model design, we used these profits as benchmarks for comparison.

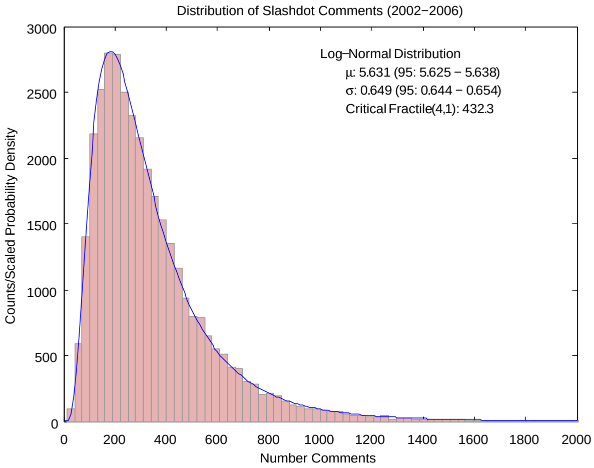

Figure 3: Distribution of Slashdot summary comments for 2002-2006 data. The 95% confidence intervals for μ and σ (first two moments of the associated normal) are [5.625,5.638] and [0.644,0.654], respectively. Critical fractile prescribes ordering 432.3 units when $r = 4$ and $c = 1$.

Order Type	Years	Critical Fractile Order
"Realistic" 2006	2002-2005	451.5
"Realistic" 2007	2002-2006	432.3
"Perfect" 2006	2006	354.7
"Perfect" 2007	2007	340.2

Figure 4: Order values for critical fractile after fitting a log-normal distribution to the data in the given year range.

Order Type	2006 Profit	2007 Profit
"Realistic" Cr. Fr.	\$3938952	\$3805189
"Perfect" Cr. Fr.	\$4083363	\$3935214

Figure 5: Profits of critical fractile for 2006 and 2007 years, using realistic and perfect assumptions.

5. MODEL DESIGN AND VALIDATION

We primarily considered two approaches which are capable of returning probability vectors over a discrete space: Naive Bayes and k-Nearest Neighbors, both of which performed better when coupled with cost sensitive training methods (described below). However, we quickly found that Naive Bayes wasn't up to the task. In order for Naive Bayes to be accurate enough in the discretized space, we needed to use a fairly coarse-grained discretization of the demand. This caused problems when converting probability vectors to order quantities using the techniques of Section 3: too much information is lost in such coarse discretization to be useful.

In retrospect, it isn't surprising that k-Nearest Neighbors using cost sensitive evaluation performed best of the approaches we tried. KNN is a classic algorithm for text mining applications, and the newsvendor problem's asymmetric cost function (if $r \neq 2c$) would suggest a cost sensitive approach.

For cost sensitive operation, we used the technique of reweighting the training data according to assignment costs as opposed to predicting the minimum expected cost class, which resulted in "collapsing" the probability vector we desired into a simple prediction vector with a weight of 1 in the most likely class. It is worth noting here that experimentally, the techniques of Section 3 proved vital; when predicting the most likely bin or the median of the customized distribution, profits suffered significantly.

The misclassification cost matrix was an NxN matrix, where N was the number of equal width bins used in discretizing the output space. Entry c_{ij} in the matrix was defined to be the loss in profit if the demand was predicted to be the center of the i^{th} bin, but was actually the center of the j^{th} bin:

$$\begin{aligned} c_{ij} &= \mathit{actual} \cdot (r - c) \\ &\quad - [\min\{\mathit{actual}, \mathit{predicted}\} \cdot r - \mathit{predicted} \cdot c] \end{aligned}$$

where $\mathit{actual} = j(\mathit{maxDemand}/N) + \mathit{maxDemand}/2N$, and $\mathit{predicted} = i(\mathit{maxDemand}/N) + \mathit{maxDemand}/2N$.

In summary, the final methodology chosen was this: first, the data was reduced to only the top X stemmed terms as previously ranked by information gain. Using Weka [11], we converted the reduced summaries to bag of words representations using TF transformation (wherein word frequency count is converted to $\log(1 + \mathit{frequency})$). Further, we discretize the demand space into $N = 250$ equal width bins. (Tests indicated a value of this size to perform better than smaller values, though full exploration of this parameter was not done.)

Applying the cost sensitive meta classifier with the cost matrix described earlier, we applied k-Nearest Neighbors as a base classifier with $1/\mathit{distance}$ distance weighting. Finally, for each validation instance, we interpreted the returned probability vector as a customized distribution as described in Section 3, and computed the final profit.

Using the 2002-2005 data as training instances, Figure 5 shows the total profit for 2006 using this method, while varying the number of terms used X and the value of k for KNN.

	K = 200	100	50	20
X = 1000	\$4135301	\$4142325	\$4134584	\$4107730
X = 500	\$4148215	\$4159586	\$4154879	\$4124844
X = 300	\$4149253	\$4150081	\$4150761	\$4131604
X = 200	\$4162993	\$4176796	\$4171004	\$4154230
X = 100	\$4157303	\$4161411	\$4154230	\$4123017

Figure 6: Total profits of the KNN approach for 2006 validation set, varying number of stemmed terms chosen X and number of neighbors used k.

Amongst these values for X and k, we see a global maximum in profit at $k = 100$ and $X = 200$, for a total profit of \$4176796. Referring back to Figure 4, this represents a gain of about 6% over the "realistic" critical fractile profit for the validation data, and a 2.2% profit increase over a "perfect" critical fractile approach.

6. TEST SET RESULTS

Having settled on suitable values of $k = 100$ for the Nearest Neighbors algorithm and $X = 200$ of the top information gaining terms to use, we applied the method to the 2007 test data using years 2002-2006 as training examples. The total profit gained was \$3965615. This represents a 4.21% increase over "realistic" critical fractile for the test data, and a 0.77% increase over the profit of the "perfect" critical fractile solution.

These increases are definitely modest. However, because of the large sample size (7,210 test instances), a one tailed Student's t-test shows that both increases are significant at a 99% confidence level.

7. CONCLUSION AND FUTURE WORK

With these results, it seems that data mining techniques such as the k-Nearest Neighbors classifier and cost sensitive methods could be viable solutions for the newsvendor demand prediction problem. Of course, this is only the case when enough historical data, including demand *and* product feature data, is available.

Given the amount of data available in this application, it seems odd that the profit gains were as modest as they were. On the other hand, it isn't clear how much correlation actually exists between story summaries and the number of comments gained. Because Slashdot comments are threaded and discussion based, a story with many comments is more likely to gain more. These sorts of "popularity based" feedback systems would be very sensitive to initial conditions which might not be represented in the data. This observation further supports the idea that the Slashdot data we've gathered is newsvendor-like; in the marketplace, some short life-cycle products (particularly consumer electronics) are either "flops" or "runaway hits," fueled in large part by initial popularity.

In terms of improving the results shown here, a couple of ideas spring to mind. First, a method of ranking the stemmed terms which respects the continuous nature of the target space, such as variance reduction, may be more appropriate than information gain. Second, while the simple approach of discretizing the demand space and applying Naive Bayes didn't work, we suspect a more sophisticated approach tailored to the operation of critical fractile may be possible. Basically, while critical fractile assumes a distribution over the demand d, and orders q such that

$$Pr(d \leq q) = \frac{r - c}{r} \ ,$$

we can use Bayes' rule and conditional independence assumptions to estimate a solution for q such that

$$Pr(d \leq q | F_1, F_2, \ldots, F_X) = \frac{r - c}{r} \ .$$

If the training data is processed and organized appropriately, we should be able to accomplish this using a binary search on q.

Finally, it is possible that older instances may be less relevant to predicting demand than newer instances. In other words, it may be that performance will improve (both for our model and for "realistic" critical fractile) if we train not on all available historical data, but only the most recent year or two. Determining how much historical data to take into account, however, is a complex problem in itself. Thus, we've here assumed that any realistic approach will utilize as much historical data as is available.

8. REFERENCES

[1] C version of the libstemmer library. Available as `http://snowball.tartarus.org/download.php`.

[2] E. Barnes, J. Dai, S. Deng, D. Down, M. Goh, H. C. Lau, and M. Sharafali. Electronics manufacturing service industry. The Logistics Institute–Asia Pacific, Georgia Tech and The National University of Singapore, Singapore, 2000.

[3] S. Chick, H. Mamani, and D. Simchi-Levi. Supply chain coordination and the influenza vaccination. In *Manufacturing and Service Operations Management*. Institute for Operations Research and the Management Sciences, 2006.

[4] F. Y. Edgeworth. The mathematical theory of banking. *Journal of the Royal Statistical Society*, 1888.

[5] G. Gallego. Ieor 4000: Production management lecture notes, 2007. Available as http://www.columbia.edu/ gmg2/4000/pdf/lect_07.pdf.

[6] K. Lang. Newsweeder: learning to filter netnews. pages 331–339. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1995.

[7] E. L. Porteus. *Foundations of Stochastic Inventory Theory*. Stanford University Press, Stanford, CA, 2002.

[8] A. Raman and M. Fisher. Reducing the cost of demand uncertainty through accurate response to early sales. *Operations Research*, 44(4):87–99, January 1996.

[9] H. E. Scarf. A min-max solution of an inventory problem. In *Stanford University Press*, 1958.

[10] C. L. Vairaktarakis. Robust multi-item newsboy models with a budget constraint. *International Journal of Production Economics*, pages 213–226, 2000.

[11] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[12] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90, Apr. 1999.

[13] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. pages 412–420. Morgan Kaufmann Publishers Inc., 1997.

APPENDIX

A. RELATIVE INFORMATION GAIN OF SLASHDOT TERMS

Those familiar with the social news aggregation site `slashdot.org` know it as a popular news source and discussion board for a range of topics, usually related to science and technology. On Slashdot, users find news or other interesting articles on the internet, write up short summaries about them, (usually around 100 words long), and submit them to designated editors. The editors then sort through the entries and post the ones they find interesting to the front page at the rate of about 19 stories per day (for years 2002-2007). The editors also associate each summary with a category, such as "Your Rights Online," "Ask Slashdot," and "Linux." Users are then free to comment on and discuss the story and/or summary (a common joke insists that very few users read the story itself).

For this paper, we needed to rank terms (stemmed using the Libstemmer library [1]) according to some measure of "importance" with respect to the number of comments a summary would receive. To do this, we discretized the space of comments into 20 classes, and computed the information gain for each term. Section 4 gives the details of this process.

Looking at the ranking produced by this process gives some insight into the nature of the data, and is also interesting for anyone who enjoys reading Slashdot. Figure A shows the top 12 stemmed terms (out of 83,805) ranked according to information gain. Most notably, this figure indicates that our data contains some highly dependent terms. For instance, "ask" and "slashdot" are found together in articles in the "Ask Slashdot" category. The term "slashdot" appears on it's own as well: only about 37% of articles containing the term "slashdot" also contain "ask." The term "cliff" is actually the name of an editor; about 80% of the summaries posted by Cliff have the category "Ask Slashdot." Finally, the term "i" is often seen in editorial footnotes, but is also seen in 72% of "Ask Slashdot" articles.

In fact, while we treated category terms just as any other words, they appear to have significant importance in terms of information gain. While all of these are also commonly present in story summaries, "hardwar," "right," "apple," "review," and "mobile" are also words appearing in categories.

Stemmed Term	Ranking
ask	1
cliff	2
hardwar	3
microsoft	4
review	5
window	6
wireless	7
slashdot	8
mobil	9
right	10
i	11
appl	12

Figure 7: The top 12 stemmed terms appearing in the Slashdot corpus, as ranked by information gain.

Figure A shows the ranked information gain position of editor names, as well as the number of stories posted by that editor for 2002 to 2006. It appears that there isn't a significant correlation between the number of stories posted by an editor and the relative predictive capability of their posting.

Stemmed Term	Ranking	Stories Posted
cliff	2	920
scuttlemonkey	19	2326
zonk	28	4975
kdawson	68	625
cmdrtaco	91	4499
cowboyn	1073	2409
samzenpus	2450	1056

Figure 8: Ranked position of selected Slashdot editors, as ranked by information gain.

Figure A shows the ranking of a few more selected stemmed terms, along with the number of summaries those terms appeared in for the 2002 to 2006 data.

Stemmed Term	Ranking	Summaries Appeared In
bush	30	171
roland	34	316
riaa	37	367
sco	44	405
iraq	135	80
kerri	149	27
googl	320	1413
evil	535	116
photoshop	1296	50
vorbi	1616	64
mp3	2014	541
option	3092	472
acid2	4346	10
gimp	8495	54
att	62476	7

Figure 9: Ranked position of selected terms of interest, as ranked by information gain.