Coping with Demand Shocks: A Distribution-Free Algorithm for Solving Newsvendor Problems with Limited Demand Information

Shawn O'Neil[*], Xuying Zhao[†], Daewon Sun[‡],
Amitabh Chaudhary[§], Jerry C. Wei[¶]

{soneil, xzhao1, dsun, Amitabh.Chaudhary.1, jwei}@nd.edu

University of Notre Dame

Notre Dame, IN 46556

November 12, 2008

[*]Mailing Address: Shawn O'Neil, Cushing 206, Department of Computer Science and Engineering, Notre Dame, IN 46556.

[†]Mailing Address: Xuying Zhao, 361 MCOB, Department of Management, Notre Dame, IN 46556; Telephone: 574-631-9061; Fax: 574-631-5127.

[‡]Mailing Address: Daewon Sun, 359 MCOB, Department of Management, Notre Dame, IN 46556; Telephone: 574-631-0982; Fax: 574-631-5127.

[§]Mailing Address: Amitabh Chaudhary, 352 Fitzpatrick, Department of Computer Science and Engineering, Notre Dame, IN 46556; Telephone: 574-631-8322.

[¶]Mailing Address: Jerry C. Wei, 355 MCOB, Department of Management, Notre Dame, IN 46556; Telephone: 574-631-5460; Fax: 574-631-5127.

Coping with Demand Shocks: A Distribution-Free Algorithm for Solving Newsvendor Problems with Limited Demand Information

Abstract

We present a new, robust, and effective algorithm for the multiple-period newsvendor problem when there is little demand information available. In today's competitive market, demand volume and even distribution can change quickly. The algorithm needs only a rough estimate of the lower and upper bounds of demand range; no other knowledge such as the demand mean, variance, or distribution type is necessary. Through simulations we show that our algorithm performs well compared to four other standard newsvendor problem solutions in a variety of situations, except when salvage values are high.

Keywords: *Newsvendor Problem; Demand Forecasting; Demand Shocks; Machine Learning Algorithm; Inventory Management; Stochastic Demands.*

1 Introduction and Motivation

In the newsvendor problem (NVP), a firm faces stochastic demand and most likely has only one opportunity to supply the product. Overage of inventory must be salvaged or discarded, while underage costs a firm in lost sales and sometimes in lost customer loyalty. Therefore, it is critical for a firm facing the NVP to choose optimal order quantities. The NVP is commonly observed in managing goods with short life cycles and fast-changing demand, such as fashion apparel, seasonal toys, high-tech consumer gadgets, and perishable fresh food (Silver, Pyke, and Peterson 1998). For example, Ziegler (1994) and Ziegler (1995) reported that in one year IBM salvaged inventory worth $700 million, and in another year IBM understocked some popular products, incurring an estimated $100 million in lost sales. In supermarkets, ever increasing product variety and assortment makes it more difficult to match supply with demand. Further, overstocking perishable products substantially increases the product spoilage rate, which can exceed 40% in some supermarkets (Ketzenberg and Ferguson 2008).

The NVP is easier to solve when the firm can forecast the full demand distribution (including type, mean, and variance), in which case it can apply the well-known "critical ratio" solution (Porteus 2002). However, it is very hard to forecast the precise demand distribution in reality. In many highly competitive markets, practitioners confront fast-changing demand and can face frequent "demand shocks" which indicate changes taking place in the mean or shape of the distribution. Such shocks can be caused by a competitor unexpectedly entering the market, surprisingly high popularity for the new version of a product, or macro economic disturbance like the 2008 credit crisis. A forecasting method which uses few parameters to describe the demand distribution cannot possibly capture such changes (Murty 2006). Today, even with the advent of information technology that collects point-of-sales data and analyzes it in real time using data mining techniques, accurately forecasting a demand distribution remains a challenging task.

Furthermore, the performance of the critical ratio solution decreases when demand variance grows, and in practice, demand variance is usually quite high. Working with retailers, Nahmias and Smith (1994) found that even for basic items in department stores demand can vary dramatically even after adjusting for known causal factors such as promotions. They observed that, in a major retail chain, the variance to mean ratios of weekly demand can range from 3 to 500 for items that can be replenished frequently.

Graves (1999) recognizes that one major theme in the continuing development of inventory theory is to incorporate more realistic assumptions about product demand into inventory models. He observes that in most industrial contexts, demands are uncertain and hard to predict, behaving like random walks that evolve over time with frequent changes in their directions and rates of growth or decline. In this paper, we study the newsvendor problem when there is no information about demand mean, variance, or distribution type. We present a new machine learning algorithm, called Weighted Majority Newsvendor Shifting (WMNS). The algorithm is unique as it only requires rough estimates of the demand range (and thus is robust to forecast errors) and quickly adapts to demand variation and demand shocks. We design simulations to compare WMNS against four other traditional "distribution-free" methods such as the Scarf's algorithm, moving average, and exponential smoothing. The results show that WMNS outperforms all the traditional methods (in terms of relative regret) in stationary demand situations, i.e., when the demand distribution is unknown but is stationary along time. As demand variation increases, the performance remains steady while the performances of other methods worsen. Most of the results we present focus on the performance of WMNS and the other approaches in the face of demand shocks (sudden changes in the demand distribution mean). In these tests, the WMNS algorithm again outperforms others whether the shocks reflect a drastic increase or drastic decrease of the mean demand. In our simulations, the performance of WMNS is also consistent across three demand distributions: normal, lognormal, and uniform. The only instance in which WMNS is not the best among the five methodologies is when the salvage value is high.

2 Literature Review

We are interested in the newsvendor problem in the absence of information on the mean, variance, or distribution of the demand. This problem is practically significant and has attracted extensive attention in the literature, which we review in Section 2.1. The algorithm we propose is based on machine learning. To add context to our algorithm, we summarize the relevant known machine learning algorithms in Section 2.2.

2.1 Newsvendor Problem with Incomplete Demand Information

The newsvendor problem is applicable to a wide variety of products in industry. For example, fashion goods are typically short lived (Raman and Fisher 1996), consumer electronics have a short selling season due to their continuously evolving nature (e.g., cellular phones can have a lifecycle as short as six months) (Barnes, Dai, Deng, Down, Goh, Lau, and Sharafali 2000), and some vaccines such as those for influenza are only useful for a single season (Chick, Mamani, and Simchi-Levi 2008).

When full demand information is available, newsvendor problems are easily solvable using a stochastic model for the demands. However, such approaches are inadequate in many real-world applications because full demand information is hard to get. Newsvendor problems without full demand information have inspired multiple research papers. For example, given only the demand mean and variance, Scarf (1958) and Gallego and Moon (1993) adopt the *maximin* approach to derive the order quantity that maximizes the worst case profits. Gallego (1998) and Gallego and Moon (1993) use the maximin approach to solve the multiple-period newsvendor problem with limited demand information. Moon and Choi (1995) apply the maximin approach to the newsvendor problem modified to include customer balking.

However, as noted in Perakis and Roels (2008), the maximin approach is risk averse. In some situations, it leads to an order quantity much lower than the optimal one. "For instance, when only the mean is known and the lost sales cost is zero, the maximin order quantity is zero, even if the profit margin is high."(Perakis and Roels 2008)

Perakis and Roels (2008) also show that a slightly less conservative approach is to optimize according to the *minimax regret*, which aims to minimize the maximum loss from not being able to make optimal decisions because of limited demand information. For example, Chamberlain (2000) applies a minimax regret approach to the newsvendor problem when only the distribution type is known, and not the mean or variance. Further, Bergemann and Schlag (2005) and Lim and Shanthikumar (2007) relax the assumption of known distribution type; instead, they assume the distribution type can be any one of a specified group. Perakis and Roels (2008) analyze the newsvendor problem with a minimax regret approach when only partial demand information (e.g., mean, variance, symmetry, unimodality) is known.

Vairaktarakis (2000) describes solutions for several performance criteria including minimax regret in the setting of multiple item types per period and a budget constraint, where the only restriction is a lower and upper bound on demand. Like others mentioned, this solution is risk averse because of the strict minimax criterion and the loose restrictions placed on the demand. In this paper, we also adopt a variation on the minimax regret strategy and also only require an estimate of the demand range. However, unlike the solutions presented in Vairaktarakis (2000), we do not attempt to minimize the maximum regret in each period. We instead attempt to minimize the maximum regret from the best single repeated order quantity over many periods. Thus, even under such loose restrictions on the demand, our approach does not suffer from being particularly risk averse.

2.2 Machine Learning and Expert Advice Problems

An important topic in machine learning is *learning from expert advice* (Blum 1996, Kalai and Vempala 2005). In an expert advice problem we need to make a series of decisions. For each decision we have access to the recommendations of a set of n experts. After each decision the correct decision is revealed to us. The objective is to minimize the error, usually to an amount comparable to the error incurred by the *best expert in hindsight*.

In the variant of interest to us, in every period j, we make a $prediction_j$ in the range $[0, 1]$. The true value, revealed after each prediction, is $label_j \in [0, 1]$. The error is measured by the total regret $(\sum_j |prediction_j - label_j|)$. For this problem, Littlestone and Warmuth (1994) present an algorithm called Weighted Majority Continuous, or WMC, which achieves the following bound

$$\mathsf{WMC}_{TotalRegret} \leq (1 + \epsilon)BestExpert_{TotalRegret} + \frac{\ln(n)}{\epsilon} \ .$$

Whereas $|prediction - label|$ is a rather simple regret function, the regret function for the newsvendor problem (which is the profit of ordering the demand minus the profit of ordering the prediction) has a larger range and is more complex, requiring a careful analysis.

In the "shifting target" variant of the above problem the objective is to develop a stronger regret bound. Instead of comparing with the single best expert in hindsight, the sequence of decisions is partitioned into some k subsequences, and for each subsequence the best expert in hindsight for that subsequence is considered. Littlestone and Warmuth (1994) also describe a "shifting target" algorithm, which has a similar bound, except now the comparison is with the sum of the individual regrets of the best expert for each of the k subsequences—when the best possible partitioning into subsequences is also chosen in hindsight.

While Littlestone and Warmuth (1994) mention that a continuous version of the algorithm can be created, for this variant they assume labels and predictions are discrete in the set

$\{0, 1\}$. As such, our algorithm and its analysis for the continuous newsvendor regret function differs significantly from the original formulation. We focus on adapting this shifting target approach because a sequence of demands which experiences a shock can be seen as two separate sequences, each of which has its own best critical fractile solution.

Weighted Majority and variations thereof have been applied to a wide variety of areas including online portfolio selection (Cover and Ordentlich 1996a, Cover and Ordentlich 1996b), robust option pricing (Demarzo, Kremer, and Mansour 2006), and predicting user actions on the world wide web (Armstrong, Freitag, Joachims, and Mitchell 1995).

3 Algorithm Description and Theoretical Bounds

In studying the newsvendor problem, we suppose there are t time periods. At the beginning of each period, the firm needs to decide an order quantity to stock. At the end of each period, the demand realizes and unsold products have to be salvaged. The purchase cost, selling price, and salvage price per item are c, r, and s, respectively. Following the newsvendor literature, we also assume an additional cost c_u per unit of demand not met in case of stockout.

Suppose we had the benefit of expert advice in solving the newsvendor problem. At the beginning of each period the experts give their recommendations and we use those to choose an order quantity γ. At the end of the period the true *demand* is revealed. The regret function is then the following: $demand \cdot (r - c) - r \cdot \min\{demand, \gamma\} + c \cdot \gamma + c_u \cdot \max\{0, demand - \gamma\} - s \cdot \max\{0, \gamma - demand\}$. As noted, this function is not simple, even if we ignore salvage values and stockout cost. A second difficulty is that expert advice is usually unavailable in newsvendor situations.

The algorithm given below, called Weighted Majority Newsvendor Shifting (WMNS), is a non-trivial adaptation of the shifting target version of WMC (O'Neil and Chaudhary 2008) that incorporates the newsvendor regret function. In Section 3.1 we show how WMNS's regret can be bounded in terms of the regret of the best expert in hindsight for a subsequence. Secondly, we show in Section 3.2 how to adapt WMNS when expert advice is not available, by simulating pretend experts. For this adaptation, we bound WMNS's regret in terms of the regret of a hypothetical algorithm which knows the entire demand sequence but is not allowed to change its order quantity too often (see SSTOPT in Section 3.3).

WMNS, essentially, maintains weights on the experts based on past performance. It works as follows: Each expert makes a prediction on what the upcoming demand will be. These predictions must be in the range $[m, M]$, where m and M are estimates of the smallest and largest demands to be seen, respectively. Each expert is also assigned an initial weight of 1.0. Every period, the amount of product ordered by the algorithm is the normalized weighted average prediction of a subset of the experts. At the end of every period, experts' weights are adjusted downward corresponding to (i) a factor dependent on the amount of regret that expert would have suffered using his/her prediction, and (ii) a β factor in the range $(0, 1]$, which controls the overall rate of weight reduction. The detailed description follows next and includes a parameter $\delta \in (0, 1]$ (discussed in Section 3.1).

Algorithm WMNS

Step 1. Initialize experts, weights, and constants. Initialize the maximum single period regret $\mathbf{C} = \max\{(M - m)(r - c + c_u), (M - m)(c - s)\}$. For each expert $i \in \{1, \ldots, n\}$, set i's initial weight $weight_i$ to 1. Let i's prediction in order period j be $pred_i^{(j)}$. (If real experts are unavailable, define $pred_i^{(j)}$ according to Equation 2 of Section 3.2.)

Repeat steps 2 and 3 for each newsvendor period $j \in \{1, \ldots, t\}$:

Step 2. Determine order quantity. Let the weight threshold be δ times the average expert weight: $\delta\left(\sum_{i=1}^{n} weight_i/n\right)$. Let $\mathcal{U}$ be the set of experts whose weights are strictly larger than the threshold. Order the quantity $\left(\sum_{i\in\mathcal{U}} pred_i^{(j)} \cdot weight_i\right) / \left(\sum_{i\in\mathcal{U}} weight_i\right)$, which is the weighted average prediction of those experts whose weights are above the threshold.

Step 3. Discover actual demand and update weights. Let the hypothetical profit of each expert i, $prof_i$, be $r \cdot \min\{demand, pred_i^{(j)}\} - c \cdot pred_i^{(j)} - c_u \cdot \max\{0, demand - pred_i^{(j)}\} + s \cdot \max\{0, pred_i^{(j)} - demand\}$. The adjusted hypothetical loss of expert i is i's regret scaled to a 0–1 range: $adjustedLoss_i = \left(demand(r-c) - prof_i\right)/\mathbf{C}$. (If this value is greater than 1, make it 1. This can happen only if the demand is outside the range $[m, M]$.) Finally, for each expert i in $\mathcal{U}$, update $weight_i$ by multiplying it by $1-(1-\beta)adjustedLoss_i$.

3.1 Shifting Target/Demand Shocks

Notice that the algorithm only "uses" experts whose weights are larger than δ times the average expert weight. If δ is set to 0, the amount ordered will be the weighted average prediction of all the experts; this will work well in many situations. However, consider a situation in which many demands are drawn from a very small sub-range of $[m, M]$, and then the underlying process changes suddenly, such that demands are drawn from another area of the possible range. If weights are allowed to become arbitrarily small the algorithm will be inordinately slow to track this change. This is because the weights of experts who we should now be listening to are extremely small compared to the others, and it will take time to adjust the weights accordingly.

Thus, WMNS compensates for this by using a "weight limiting" parameter $\delta \in (0, 1]$. Based on this parameter, WMNS ensures that no expert's weight becomes too small compared to others. The algorithm will only reduce an expert's $weight_i$ (using the weight reduction parameter β) if $weight_i$ is greater than δ times the average of all experts' weights. Keeping the weights from becoming too disparate in this manner allows WMNS to make adjustments in the face of dramatic shifts (such as unpredicted shocks to the demand distribution caused by economic factors) rapidly.

It is partly due to the use of this δ parameter that we can give the following theorem. Proofs of all theorems and lemmas can be found in the appendices.

Theorem 1 *Let* $\mathsf{OPT}^{(j)}$ *be the optimal newsvendor profit in period* j*,* $\mathsf{WMNS}^{(j)}$ *be* WMNS*'s profit in period* j*, and* $\mathsf{EX}_i^{(j)}$ *be the profit expert* i *would make in period* j *using their prediction. For any subsequence of newsvendor periods indexed from* init *to* fin *and for all experts* i*,* WMNS*'s regret satisfies*

$$\sum_{j=init}^{fin} (\mathsf{OPT}^{(j)} - \mathsf{WMNS}^{(j)}) \leq \frac{\mathbf{C} \ln\left(\frac{n}{\beta\delta}\right)}{(1-\beta)(1-\delta)} + \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=init}^{fin} (\mathsf{OPT}^{(j)} - \mathsf{EX}_i^{(j)})}{(1-\beta)(1-\delta)} \qquad (1)$$

if the salvage value $s = 0$ *and all demands are in the range* $[m, M]$*.*

This theorem implies that if there is any subsequence of the demand sequence for which a particular expert does very well (suffers low regret), then WMNS will do nearly as well. Further, if a different expert begins doing well, WMNS will track this change quickly.

The restriction on the demand range is a clear necessity for the theorem; the restriction that salvage value is 0 is due to the proof technique used and the non-linearity of the newsvendor profit function. Importantly, as we'll see in Section 4, the algorithm still operates (and usually performs very well) even if these assumptions are not met.

3.2 Simulating Experts

In general, our algorithm is able to utilize predictions from a panel of actual human experts, and we've seen that the regret suffered will be close to the regret of the best expert on any subsequence. For when such a panel of experts may not be available, we give an alternate approach wherein we simulate some *static* experts. A static expert predicts the same thing every period. For example, a simulated expert A might always predict a demand of ten units, expert B might always predict a demand of 20 units, and so on. Clearly, these simulated experts would be less powerful in their predictive abilities than human experts. However, if we create the simulated experts carefully good results can be obtained.

When simulating such static experts, we'd like to ensure that wherever the "static optimal" solution happens to be (such as what might be ordered by a clairvoyant critical fractile solution) we have an expert predicting nearby. We can achieve this by dividing the range $[m, M]$ into n "buckets" and having a simulated expert predict the minimax regret point (which Vairaktarakis (2000) calls the "deviation robust order quantity") of each bucket.

There are $n+1$ bucket endpoints, $\{q_0, q_1, \ldots, q_n\}$. For a given bucket i (with endpoints q_{i-1} and q_i) the minimax regret order quantity is $\big(q_i(r-c+c_u)+cq_{i-1}\big)/(r+c_u)$, which results in a maximum regret of $\big(c(q_i-q_{i-1})(r-c+c_u)\big)/(r+c_u)$ when the demand is at either endpoint (Vairaktarakis 2000).

To achieve our "many buckets, same regret" goal, we simply need to choose the endpoints according to:

$$q_i = \frac{i(M-m)}{n} + m \ .$$

Defining $pred_i^{(j)}$ as the demand prediction made by simulated expert i in period j, we let each expert i always predict the optimal order quantity for the i^{th} bucket:

$$pred_i^{(j)} = \frac{q_i(r-c+c_u)+cq_{i-1}}{r+c_u} = \frac{i(M-m)}{n} - \frac{c(M-m)}{n(r+c_u)} + m, \ \forall j \ . \tag{2}$$

Lemma 1 *For h periods of a newsvendor game, there exists a simulated expert i such that* *the difference in i's profit and any given static algorithm is at most*

$$\frac{c(M-m)(r-c+c_u)h}{n(r+c_u)} .$$

Lemma 1 formalizes the intuitive notion that by creating our n simulated experts according to Equation (2), the difference in profit of the best expert from any other static choice (including the optimal static choice) can be bounded from above. Further, this bound can be reduced simply by using a larger n.

The primary difficulty in implementing WMNS is clearly in estimating the range of possible demands, $[m, M]$. On the other hand, in practice a conservative estimate can be made when necessary, and n can be increased to correspondingly increase the granularity of the experts. In contrast, approaches such as distribution estimation for critical fractile can be sensitive to forecast errors. Again we stress that the algorithm still operates if the $[m, M]$ range assumption is violated, with the implication that precise information on demand bounds is not strictly necessary.

3.3 Simulated Expert Regret Bounds

Because of the way simulated experts are placed, the final regret bound below is given not merely in terms of the best expert's regret, but rather in terms of the regret of a particular type of optimal strategy which we call SSTOPT, for "Semi-Static Optimal." This restricted clairvoyant approach is allowed to split the input sequence up into k subsequences at its discretion, but is then required to choose a single order quantity to use for each subsequence (which is done in an optimal way).

Definition 1 *For a demand sequence $d_1, d_2, \ldots, d_t$, algorithm* SSTOPT *chooses orders $q_1, q_2, \ldots, q_k$ and subsequence lengths $l_1, l_2, \ldots, l_k$. For the first l_1 demands, q_1 is ordered. For the next l_2 demands, q_2 is ordered, and so on.* SSTOPT *chooses the q_i's and l_i's optimally (subject to*

$\sum_{i=1}^{k} l_i = t$ *and* $l_i \in \mathbb{N}$*) to maximize the profit:*

$$\max_{q_i,l_i} \left\{ \sum_{i=1}^{l_1} Profit(d_i, q_1) + \sum_{i=l_1+1}^{l_1+l_2} Profit(d_i, q_2) + \cdots + \sum_{i=l_1+\cdots+l_{k-1}+1}^{l_1+\cdots+l_k} Profit(d_i, q_k) \right\}$$

$Profit(d, q)$ *is the newsvendor profit function when* d *is the demand and* q *is the order.*

For the purposes of proofs in this paper, it isn't necessary to know how SSTOPT might reach its decisions, though such a study would be interesting in a theoretical sense. An extension to the work of Bertsimas and Thiele (2005) shows that when $k = 1$, SSTOPT orders the $\lceil (r - c + c_u)/(r - s + c_u) \cdot t \rceil^{th}$ order statistic of the demand sequence for all periods.

Notice the similarity between SSTOPT and a "perfect" critical fractile approach. If demands are drawn from a distribution which experiences discrete changes from time to time, a perfect critical fractile user will use the correct critical fractile solution for each subsequence where a single distribution is used. Thus, any bound given in terms of SSTOPT is also valid when expressed in terms of such a perfect critical fractile solution when the distribution changes $k - 1$ times, since SSTOPT by definition outperforms such a perfect critical fractile solution.

Theorem 2 *The total regret experienced by* WMNS *for a* t *period newsvendor game with per item cost* c*, per item revenue* r*, understock cost* c_u*, no salvage values, and all demands within* $[m, M]$ *satisfies*

$$\mathsf{WMNS}_{TotalRegret} \leq \frac{k\mathbf{C} \ln\left(\frac{n}{\beta\delta}\right)}{(1-\beta)(1-\delta)} + \frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c+c_u)t}{n(r+c_u)(1-\beta)(1-\delta)} + \frac{\ln\left(\frac{1}{\beta}\right) \mathsf{SSTOPT}_{TotalRegret}}{(1-\beta)(1-\delta)}$$

where $\mathbf{C} = \max\{(M - m)(r - c + c_u), (M - m)c\}$ *is the maximum possible single period regret,* n *is the number of experts used by* WMNS*,* $\beta \in (0, 1]$ *is the update parameter used,* *and $\delta \in (0,1]$ is the weight limiting parameter used.* SSTOPT *is a clairvoyant approach allowed to change its order quantity $k-1$ times.*

We note that the first term in the bound depends on the maximum single period regret, and is independent of the number of periods played, t. This results in the following corollary:

Corollary 1 *If k used by* SSTOPT *is constant, and n is set to equal to $\big((1-\beta)(1-\delta)(r+c_u)\big)/\big(\ln(1/\beta)c(M-m)(r-c+c_u)\big)$, the average per period regret of* WMNS *approaches*

$$1+\frac{\ln\left(\frac{1}{\beta}\right)}{(1-\beta)(1-\delta)}\mathsf{SSTOPT}_{AveragePerPeriodRegret}$$

as the number of periods t becomes arbitrarily large.

The implication of this is that while β and δ can be used to control the responsiveness of the algorithm to demand shocks, they can also be used to adjust the performance over long sequences of steady demand in a theoretical sense. However, just as with other approaches, an algorithm which is very responsive to demand shocks is also sensitive to noise.

4 Experiment Design and Results

4.1 Experiment Design

In order to evaluate the effectiveness of the WMNS algorithm, we simulate demand sequences describing a variety of conditions. We then compare the results to more traditional approaches which forecast demand using either a moving window or exponential smoothing.

We give the performance of an algorithm ALG in terms of relative regret from the "perfect critical fractile solution" $\mathsf{PERFECT}$:

$$Relative\ Regret(\mathsf{ALG})=\frac{Profit(\mathsf{PERFECT})-Profit(\mathsf{ALG})}{Profit(\mathsf{PERFECT})}*100\%,$$

where PERFECT is the algorithm which knows the full distribution information at every order point (including shape and parameters) and orders the critical fractile solution accordingly. Relative regret is similar to regret, which WMNS is designed to minimize; however, it is easier to compare performance results across experiments because it is scaled with respect to PERFECT's profit.

All experimental results are an average of 200 trials. Figure 1 shows how the average stabilizes as we include from 1 to 200 trials for various tests; 200 or even 150 trials would be sufficient to draw meaningful conclusions.

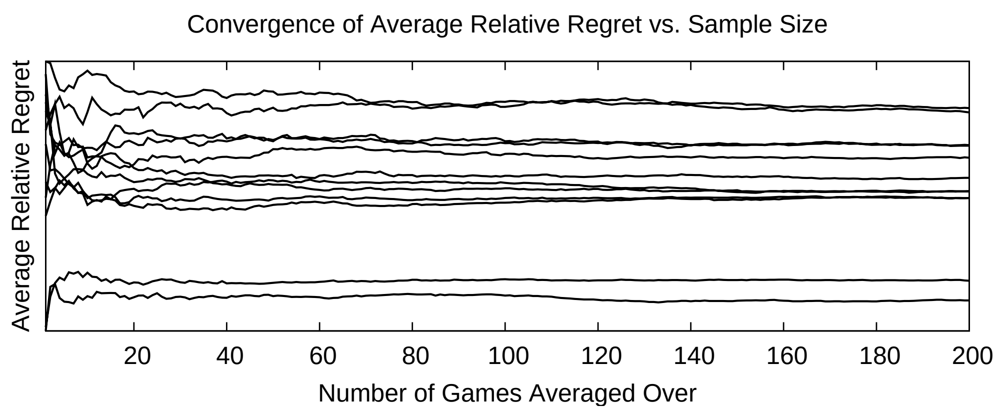

Figure 1: For all experiments, we evaluate the performance of algorithms via the average relative regret from PERFECT over 200 trials. Here we see the average stabilize as the number of trails is increased to 200, for a variety of experiments/algorithms.

The demand sequences used for testing are constructed as follows: for each trial, 100 demands are drawn from an initial distribution. We then change the distribution and draw 100 more demands, thus simulating an abrupt demand shock.

As a basic scenario for all experiments, the first distribution is normal with mean 900 and standard deviation 150. The second distribution is also normal with an identical standard deviation of 150, but has a mean of 600. The per item cost, selling price, and salvage values are \$20, \$40, and \$11, respectively. The understock cost c_u is 0 for all tests. In Section 4.3, we study changes in performance while varying shock type (increasing or decreasing demand mean), demand variance, salvage value, and distribution type.

4.2 Algorithms Tested

We test the average relative regret from PERFECT for the following five forecasting/ordering approaches:

Exponential Smoothing: EXP

This algorithm orders the current estimate of the mean in each order period. Initially, the mean is assumed to be 750 units. The mean estimate is updated according to an $\alpha \in [0, 1]$ parameter at the end of each period t after the new demand d_t is seen:

$$\mu_{t+1} = \alpha(d_t) + (1 - \alpha)\mu_t \ .$$

Larger values of α will place more emphasis on recent demands, but may be more susceptible to noise. Conversely, small values of α produce more stable predictions, but are slow to track change. We look to the advice of Brown (1959), as summarized by Niland (1970) in choosing α:

> "Brown, who is perhaps the leading authority on exponential smoothing applications, suggests that a value of .1 for α, the smoothing constant, is probably appropriate for most circumstances, but that in cases where substantial change in the demand rate is predicted, he comments that a higher value, such as .3 (or even .5) may be appropriate."

We thus use $\alpha = 0.2$, given that the input demands will be drawn from distributions with one expected demand shock.

Mean of Moving Window: MEAN

In a moving window approach, the most recent N demands are used to forecast the next demand. MEAN uses the mean of these demands as the order quantity. The mean is initially assumed to be 750 units; when the number of demands seen is between one and N, the mean is computed as the mean of whatever demand history is available. After this "warmup period," the mean of the most recent N demands is used as the next order quantity.

This approach approximates the exponential smoothing approach when $\alpha = 2/(N+1)$, so we choose $N = 9$ satisfying the equality when $\alpha = 0.2$ (Niland 1970, Nahmias 2001). A smaller window size would result in more emphasis being placed on recent demands, and the possibility of erratic forecasts. A larger window size would mean more stability, but would also track changes in the underlying demand distribution more slowly.

Critical Fractile and Moving Window: FRACT

A critical fractile solution can also be used in conjunction with a moving window forecasting method. Here, the previous N demands are used to estimate both the mean and the standard deviation of the governing distribution, and the standard critical fractile solution is then applied (Porteus 2002):

$$order = \Phi^{-1}\left(\frac{r - c + c_u}{r - s + c_u}\right) \ .$$

Here, Φ is the cumulative distribution function of the demand. We initialize this algorithm with a mean estimate of 750 and the actual standard deviation which will be used in drawing the first demand. We always allow the algorithm to use the correct distribution type (normal, lognormal, or uniform) when computing the critical fractile.

Realistically, FRACT may not know the correct shape of the demand distribution in advance. By giving this information, we hope to make this approach the most difficult benchmark for comparison. We use $N = 9$ for the window size.

Scarf's Rule and Moving Window: SCARF

We also apply Scarf's ordering rule in lieu of the critical fractile solution, using the same estimate of the mean μ and standard deviation σ as used by FRACT. Scarf's rule prescribes ordering the quantity which maximizes the worst case profit over all distributions with those parameters (Scarf 1958). Alfares and Elmorra (2005) extend Scarf's solution to the case of understock costs if $c_u < c^2$:

$$\text{If } \left(\frac{(r-c)\mu}{c\sigma}\right)^2 > \frac{(c-s)(r-c+c_u)}{c^2},\ order = \mu + \frac{\sigma}{2}\left(\sqrt{\frac{r-c+c_u}{c-s}} - \sqrt{\frac{c-s}{r-c+c_u}}\right).$$

An order of 0 is used if the condition is not met. We again use $N = 9$ for the window size.

Weighted Majority Newsvendor Shifting: WMNS

WMNS utilizes a number of decision variables: a weight update factor $\beta \in (0, 1]$, weight limiting factor $\delta \in (0, 1]$, the number of simulated experts to create n, and the "estimated demand range" $[m, M]$.

To facilitate relatively large expert weight adjustments we use $\beta = 0.1$ for all tests. (Large weight adjustments are appropriate when demands are stochastic. If the demands are, instead, adversarial a larger value of β would minimize the bound of Theorem 2.) Because the weight limiting factor δ is most crucial for determining how quickly the algorithm will adjust to demand shock, we choose $\delta = 0.5$ as a naïve balanced choice. We also (rather arbitrarily) let the number of simulated experts n be 64.

As a decently conservative estimate of the demand range $[m, M]$, we use $[300, 1200]$. Note, however, that in one experiment where the standard deviation is increased to 300 units, this assumption is frequently violated with approximately 16% of demands falling outside the range. Nevertheless, we find that WMNS still performs well, and we discuss the intuition behind this important result in the "Effect of Demand Standard Deviation" section below.

4.3 Experimental Results

Table 1 gives the relative regret (in percent) from PERFECT for the five approaches we test, in the default scenario: a normal distribution with standard deviation 150 experiencing a demand shock such that the mean reduces from 900 to 600.

Algorithm	Relative Regret
EXP	2.42
MEAN	2.37
SCARF	1.15
FRACT	1.11
WMNS	1.05

Table 1: Relative Regrets from PERFECT (in percent) in the default scenario.

These results confirm that MEAN does approximate EXP with the parameters chosen, as their relative regrets are very similar at around 2.4%. The other three algorithms perform much better, with WMNS doing best of all at 1.05% relative regret. To put these numbers in perspective, in a 10 million dollar market (that is, PERFECT's profit would be $10,000,000), EXP would have a profit of $9,758,000 whereas WMNS's profit would be $9,895,000.

Effect of Shock Type

Here, we study the effect on relative regrets as we change the shock type the demand sequence experiences. The default scenario exhibits a shock in mean demand from 900 to 600, such as might be encountered by a competitor opening a retail location nearby (or a large online retailer unexpectedly moving into a new product category). Table 2 compares this situation to one in which the shock in mean demand is from 600 to 900 units, such as might be caused by an unexpected surge in the product's popularity.

Although all traditional approaches perform better for a downward shock, an upward shock in demand mean is a more difficult situation for EXP and MEAN as compared to the others. In contrast, WMNS actually improves slightly in terms of relative regret, even though

Algorithm	μ: 900 to 600	μ: 600 to 900	μ: 900 to 900	μ: 600 to 600
EXP	2.42	2.74	1.97	2.72
MEAN	2.37	2.94	1.87	2.89
SCARF	1.15	1.29	0.70	1.10
FRACT	1.11	1.17	0.63	1.00
WMNS	1.05	0.92	0.33	0.51

Table 2: Relative Regrets from PERFECT (in percent) considering different shock types. The last two columns represent "no shock" scenarios.

it already had the lowest average relative regret of the five approaches.

The last two columns of Table 2 represent "no shock" scenarios, in which the mean stayed constant for all 200 order periods. In examining these two columns, recall that α is 0.2 and N (the window size) is 9, values consistent with the expectation of a demand shock. However, the results of the other algorithms do not surpass WMNS even when the recommended values of $\alpha = 0.1$ and $N = 19$ are used for these easier cases. For example, the best improvement was for FRACT, which when using the more appropriate α=0.1 in the no-shock μ:900 to 900 scenario reduced its relative regret to 0.35%.

Effect of Demand Standard Deviation

Varying the standard deviation of the basic scenario from 150 to 300 gives some idea of how each algorithm copes with different levels of uncertainty in the demand distribution. When the mean is 600, this represents variance/mean ratios of 37.5 to 150; when the mean is 900 it represents variance/mean ratios of 25 to 100. These ratios agree with those found in practice by Nahmias and Smith (1994).

A larger demand standard deviation represents a significantly more difficult situation for the traditional approaches, while WMNS's relative regret only increases slightly. Also, because the distributions are normal, when the standard deviation is 300 approximately 16% of the demands fall outside of WMNS's assumed demand range of $[300, 1200]$.

The intuition behind this result is that while violating the $[m, M]$ range assumption

Algorithm	σ: 150	σ: 300
EXP	2.42	4.53
MEAN	2.37	4.55
SCARF	1.15	1.86
FRACT	1.11	1.76
WMNS	1.05	1.11

Table 3: Relative Regrets from PERFECT (in percent) with different distribution standard deviations.

invalidates the proofs theoretically, in practice this range also has the important effect of defining the range of possible orders WMNS can make. Thus, if the best static order decision (the critical fractile solution) falls between m and M, it is almost always possible for WMNS to converge to it. However, if demands are seen outside of the range, there are no theoretical guarantees on how quickly this convergence will happen.

Effect of Salvage Value

Next, we look at the effect of changing the salvage value s. We test per item salvage values \$3.5, \$6.0, \$8.5, \$11.0, \$13.5, \$16.0, and \$18.5.

Algorithm	s: 3.5	s: 6.0	s: 8.5	s: 11.0	s: 13.5	s: 16.0	s: 18.5
EXP	1.51	1.60	1.84	2.42	3.19	4.45	6.37
MEAN	1.51	1.62	1.82	2.37	3.10	4.33	6.23
SCARF	1.46	1.37	1.18	1.15	0.99	0.80	0.48
FRACT	1.47	1.37	1.16	1.11	0.93	0.76	0.52
WMNS	1.25	1.15	1.09	1.05	1.00	0.99	0.94

Table 4: Relative Regrets from PERFECT (in percent) with various salvage values.

Looking at the results in Table 4, we see that increasing the salvage value has a dramatic effect on the performance of EXP and MEAN. This is expected as these approaches do not take salvage value into account. WMNS performs the best of all algorithms for salvage values lower than our default value of \$11. As the salvage value is increased, SCARF and FRACT are able to reduce their relative regret significantly. While WMNS is also able to improve, it does so at a slower rate, such that for salvage values above $11 WMNS is no longer the best.

Effect of Distribution Type

Finally, we also test the algorithms on lognormal and uniform distributions. These distributions are commonly found in the literature (Perakis and Roels 2008, Godfrey and Powell 2001). Doing so allows us to test the "distribution free" aspect of WMNS.

Just as with the default scenario, in each case the first distribution (used to draw demands before the shock) has a mean of 900, and the second distribution has a mean of 600. However, we use the larger value of 300 units for the standard deviation in these tests because the normal and lognormal distributions are very similar with a standard deviation of 150. For the case of the uniform distribution, the first distribution is uniform with range $900 \pm 300 \cdot \sqrt{3.0}$, and the second distribution is uniform with range $600 \pm 300 \cdot \sqrt{3.0}$.

Again, the larger standard deviation means that more demands will fall outside of WMNS's assumed demand range of $[300, 1200]$ on the normal and lognormal tests.

Algorithm	Normal	Lognormal	Uniform
EXP	4.53	2.91	5.78
MEAN	4.55	2.98	5.75
SCARF	1.86	2.39	2.22
FRACT	1.76	2.04	1.64
WMNS	1.11	1.44	1.42

Table 5: Relative Regrets from PERFECT (in percent) with various distribution types.

The performance of the first two algorithms in Table 5 improves when they are tested on a lognormal distribution; this is because the mean of the distribution is closer to the critical fractile solution in this case. For the uniform distribution, these algorithms suffer as the opposite is true.

The other solutions appear to be able to cope with the different distribution types effectively, with FRACT's performance approaching the performance of WMNS on the uniform distribution. Note that FRACT is given the distribution type for use in computing the critical fractile solution. Had it always assumed a normal distribution, which is commonly the case in practice, it would have suffered on the other distributions. WMNS, in contrast, makes no assumptions about the distribution type.

4.4 Experimental Summary

In most of the experimental situations the machine learning approach performed better than the other approaches with the parameters chosen. One notable feature WMNS exhibits is stability of performance across a wide range of scenarios. Over all the experiments presented in this section, WMNS had an average relative regret of 1.01%, with a standard deviation of only 0.25%. The next best algorithm, FRACT, had an overall relative regret of 1.14%, but with a standard deviation of 0.41%.

WMNS performs well in cases of both upward and downward demand shock because of the use of the weight limiting factor δ. The algorithm does fail to outperform the other approaches in the case of high salvage value. One possible explanation might be that whereas the other approaches attempt to model the demand distribution itself, WMNS attempts to model the regret function over the range $[m, M]$ (since weights are adjusted based on newsvendor regret). When the profit margins increase, the variance in the regrets suffered increases even though the distribution has stayed the same, making convergence to the correct order more difficult.

The initial order for all approaches tested was 750 units. As such, each approach has a "warmup period," wherein moving windows are filled up, exponential smoothing puts weight on recent demands, and WMNS adjusts weights so they aren't uniformly equal. We ran similar experiments to the ones shown, but didn't count any profit during the first 50 periods in order to normalize out any warmup effects. However, the results were virtually identical. This is because the initial warmup period can be seen as a demand shock which must be adjusted to.

5 Concluding Remarks

In this paper, we proposed a new machine learning method to solve a newsvendor problem with incomplete demand information. The method, called Weighted Majority Newsvendor Shifting (WMNS), requires very little information about the demand distribution but performs quite well in various situations. We tested WMNS's performance in two interesting situations: (i) stationary demand distribution and (ii) demand shocks caused by demand distribution changes. Through numerical experiments, we found that the method outperforms all benchmarks traditionally used when demand is stationary. Next, we also found that the proposed method performs best in most of the experiment settings which include demand shocks. The only environment in which WMNS does not perform best is when the salvage value is very high. However, even in this case the performance of the algorithm is stable and predictable.

As retailing channels become more competitive and complex, an efficient and effective method to handle uncertain demand in an automated fashion becomes more important. This is evident given the increasing research attention paid to newsvendor problems with incomplete demand information. Furthermore, fierce competition in retailing as well as unpredictable demand changes cause demand shocks frequently. When full demand distribution information is not available, practitioners have few choices, including Scarf's rule and forecasting techniques, many of which still require estimating the demand mean and variance. As demonstrated earlier, WMNS's performance was quite outstanding, as compared to the traditional approaches, even in the presence of demand shocks. Therefore, we believe WMNS can be successfully applied to handle the complicated problems faced by many practitioners today.

There are several ways to extend our study. First, in this paper we only considered one demand shock for each experiment setting. Although theory indicates the method will perform well, further experiments and investigation for multiple demand shocks would be useful to demonstrate its robustness compared to traditional approaches. Second, we tested WMNS's performance with three popular distributions: normal, lognormal, and uniform. Testing its performance with a wider variety of distributions and detailed sensitivity analysis may reveal more insights into whether the method's performance is sensitive to certain types of demand functions, despite its distribution-free nature. Third, in the current study we demonstrated WMNS's performance through worst case analysis and numerical experiments. Deriving an expectational analysis would be very important future research. Finally, we plan to extend our research by investigating the performance of the proposed machine learning algorithm and its variants under other stochastic environments without full demand information.

References

Alfares, H. K., and H. H. Elmorra (2005): "The distribution-free newsboy problem: Extensions to the shortage penalty case," *International Journal of Production Economics*, 93(1), 465–477.

Armstrong, R., D. Freitag, T. Joachims, and T. Mitchell (1995): "Webwatcher: A learning apprentice for the world wide web," in *AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments*.

Barnes, E., J. Dai, S. Deng, D. Down, M. Goh, H. C. Lau, and M. Sharafali (2000): "Electronics manufacturing service industry," TLI-AP white paper, The Logistics Institute–Asia Pacific, Georgia Tech and The National University of Singapore, Singapore.

Bergemann, D., and K. Schlag (2005): "Robust monopoly pricing: The case of regret," EUI Working Paper ECO No. 2005 /10, European University Institute.

Bertsimas, D., and A. Thiele (2005): "A data driven approach to newsvendor problems," Discussion paper, Massechusetts Institute of Technology, Cambridge, MA.

Blum, A. (1996): "On-line Algorithms in Machine Learning," in *Proceedings of the workshop on on-line algorithms, Dagstuhl, Germany*, pp. 306–325.

Brown, R. (1959): *Statistical Forecasting for Inventory Control*. McGraw-Hill Book Company, New York, NY.

Chamberlain, G. (2000): "Econometrics and decision theory," *Journal of Econometrics*, 95(2), 255–283.

Chick, S., H. Mamani, and D. Simchi-Levi (2008): "Supply chain coordination and the influenza vaccination," *Operations Research*, Forthcoming.

Cover, T. M., and E. Ordentlich (1996a): "On-line portfolio selection," in *Proceedings of the ninth annual conference on Computational Learning Theory*, pp. 310–313.

——— (1996b): "Universal portfolios with side information," *IEEE Transactions on Information Theory*, 42(2), 348–363.

Demarzo, P., I. Kremer, and Y. Mansour (2006): "Online trading algorithms and robust option pricing," in *Proceedings of the thirty-eighth annual AMC Symposium on Theory of Computing*, pp. 477–486.

Gallego, G. (1998): "New bounds and heuristics for (Q,R) policies," *Management Science*, 44(2), 219–233.

Gallego, G., and I. Moon (1993): "The distribution free newsboy problem: Review and extensions," *Journal of the Operational Research Society*, 44(8), 825–834.

Godfrey, G. A., and W. B. Powell (2001): "An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution," *Management Science*, 47(8), 1101–1112.

Graves, S. (1999): "A single-item inventory model for a nonstationary demand process," *Manufacturing and Service Operations Management*, 1(1), 50–61.

Kalai, A., and S. Vempala (2005): "Efficient algorithms for online decision problems," *Journal of Computer and System Sciences*, 71(3), 291–307.

Ketzenberg, M. E., and M. Ferguson (2008): "Managing slow moving perishables in the grocery industry," *Production and Operations Management*, 17(4), 513–521.

Lim, A. E. B., and J. G. Shanthikumar (2007): "Relative entropy, exponential utility, and robust dynamic pricing," *Operations Research*, 55(2), 798–214.

Littlestone, N., and M. K. Warmuth (1994): "The weighted majority algorithm," *Information and Computation*, pp. 212–261.

Moon, I., and S. Choi (1995): "The distribution free newsboy problem with balking," *Journal of the Operational Research Society*, 46(4), 537–542.

Murty, K. (2006): "Forecasting for supply chain and portfolio management," Working Paper, University of Michigan.

Nahmias, S. (2001): *Production and Operations Analysis*. McGraw-Hill Irwin, New York.

Nahmias, S., and S. A. Smith (1994): "Optimizing inventory levels in a two-echelon retailer system with partial lost sales," *Management Science*, 40(5), 582–596.

Niland, P. (1970): *Production Planning, Scheduling, and Inventory Control: A Text and Cases*. Collier-MacMillan Canada, Ltd., Toronto, Ontario.

O'Neil, S., and A. Chaudhary (2008): "Comparing online learning algorithms to stochastic approaches for the multi-period newsvendor problem," *Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments*.

Perakis, G., and G. Roels (2008): "Regret in the newsvendor model with partial information," *Operations Research*, 56(1), 188–203.

Porteus, E. L. (2002): *Foundations of Stochastic Inventory Theory*. Stanford University Press, Stanford, CA.

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

Scarf, H. E. (1958): "A min-max solution of an inventory problem," in *Studies in the mathematical theory of inventory and production*, ed. by K. Arrow, S. Karlin, and H. Scarf, pp. 201–209. Stanford University Press, Stanford, CA.

Silver, E., D. Pyke, and R. Peterson (1998): *Inventory Management and Production Planning and Scheduling*. Wiley & Sons, NY, NY.

Vairaktarakis, G. L. (2000): "Robust multi-item newsboy models with a budget constraint," *International Journal of Production Economics*, 66(3), 213–226.

Ziegler, B. (1994): "IBM sells out new Aptiva pc; Shortage may cost millions in potential revenue," Wall Street Journal, October 7.

——— (1995): "IBM's O'Malley resigns post at pc division," Wall Street Journal, February 7.

A Proof of Theorem 1

We define the variables $r, c, c_u, n, m, M, \beta, \delta$ and $\mathbf{C}$ as they are defined and used in Algorithm WMNS. We also define two subsets of the n experts: $\mathcal{U}$, those which are "updatable", and $!\mathcal{U}$, those which are not. For any period j, $\mathcal{U}$ is defined as those experts whose weights in period j, $w_i^{(j)}$, satisfy $w_i^{(j)} > (\delta \sum_{i=1}^{n} w_i^{(j)})/n$. $!\mathcal{U}$ contains all other experts.

In period j, each expert i predicts a demand of $x_i^{(j)}$ (for ease of presentation we prefer this shorter notation here instead of the corresponding $pred_i^{(j)}$ in WMNS's description) and the true demand is revealed to be $d^{(j)}$ at the end of the period. The experts are allowed to change their predictions any way they wish between periods. The only restriction is that $x_i^{(j)}$ and $d^{(j)}$ are within the interval $[m, M]$ for all i and j. WMNS will aggregate the predictions of the experts, and order an amount $\gamma^{(j)}$ which is the weighted average prediction of the *updatable* experts in period j: $\gamma^{(j)} = \sum_{i \in \mathcal{U}} w_i^{(j)} x_i^{(j)} / \sum_{i \in \mathcal{U}} w_i^{(j)}$.

Clearly, the optimal choice for period j would be $d^{(j)}$, so the true optimal profit for period j, which we'll denote as $\mathsf{OPT}^{(j)}$, is $d^{(j)}(r - c)$. We denote WMNS's profit in period j as $\mathsf{WMNS}^{(j)}$, and the profit each expert i would have made as $\mathsf{EX}_i^{(j)}$.

In every period, after the actual demand $d^{(j)}$ is revealed, we update *only the experts in* $\mathcal{U}$ by a factor of F: $w_i^{(j+1)} = w_i^{(j)} F$ where

$$F = 1 - (1 - \beta) f(d^{(j)}, x_i^{(j)}) \geq \beta^{f(d^{(j)}, x_i^{(j)})} .$$

This lower bound on the update factor holds because $f(d^{(j)}, x_i^{(j)})$ is expert i's regret scaled to a $[0, 1]$ range by dividing it by $\mathbf{C}$, the maximum possible regret:

$$f(d^{(j)}, x_i^{(j)}) = (\mathsf{OPT}^{(j)} - \mathsf{EX}_i^{(j)})/\mathbf{C} .$$

We let $s^{(j)} = \sum_{i=1}^{n} w_i^{(j)}$ be the total sum of weights of all experts in period j, $s_{\mathcal{U}}^{(j)} =$

$\sum_{i\in\mathcal{U}} w_i^{(j)}$ be the sum of weights of updatable experts, and $s_{!\mathcal{U}}^{(j)} = \sum_{i\in !\mathcal{U}} w_i^{(j)}$ be the sum of weights of not updatable experts. We define *init* to be the index of the first period of the subsequence in question, and fin to be the index of the last period of the subsequence.

We are interested in finding a bound for $\ln(s^{(fin+1)}/s^{(init)})$. First we note that, by the operation of WMNS, for any period j and any expert i, $w_i^{(j)} \geq \beta\delta s^{(j)}/n$. This is also true for the first period, because $1 \geq \beta\delta s^{(1)}/n = \beta\delta$.

By the update factor F and the mechanism of WMNS, we have the following bound on $s^{(j+1)}$:

$$
\begin{aligned}
s^{(j+1)} &= \sum_{i\in\mathcal{U}} w_i^{(j)} \left[1-(1-\beta)f(d^{(j)},x_i^{(j)})\right] + \sum_{i\in !\mathcal{U}} w_i^{(j)} \\
&= s^{(j)} - (1-\beta)\sum_{i\in\mathcal{U}} w_i^{(j)} f(d^{(j)},x_i^{(j)}) \\
&= s^{(j)} - (1-\beta)\sum_{i\in\mathcal{U}} \left[\frac{w_i^{(j)}d^{(j)}(r-c) - w_i^{(j)} r\min\{d^{(j)},x_i^{(j)}\} + w_i^{(j)} c_u \max\{d^{(j)}-x_i^{(j)},0\} + w_i^{(j)}x_i^{(j)}c}{\mathbf{C}}\right] \\
&\leq s^{(j)} - (1-\beta)\left[\frac{s_{\mathcal{U}}^{(j)}d^{(j)}(r-c) - r\min\{s_{\mathcal{U}}^{(j)}\gamma^{(j)}, s_{\mathcal{U}}^{(j)}d^{(j)}\} + c_u\max\{s_{\mathcal{U}}^{(j)}(d^{(j)}-\gamma^{(j)}),0\} + s_{\mathcal{U}}^{(j)}\gamma^{(j)}c}{\mathbf{C}}\right] .
\end{aligned}
$$

We arrive at the last line by the definition of $s_{\mathcal{U}}^{(j)}$ and $\gamma^{(j)}$, as well as by virtue of the fact that the summation over a minimum is less then or equal to the minimum of two summations. [1]

Next we need a lower bound for $s_{\mathcal{U}}^{(j)}$:

$$
s_{\mathcal{U}}^{(j)} = s^{(j)} - \sum_{i\in !\mathcal{U}} w_i^{(j)} \geq s^{(j)} - \sum_{i\in !\mathcal{U}} \delta s^{(j)}/n \geq s^{(j)}(1-\delta) .
$$

[1]We also use the fact that the summation over a maximum is greater than or equal to the maximum of summations. Here is where the proof breaks down if salvage profits are a consideration: salvage values add a term of $-s \cdot max\{x_i^{(j)} - d^{(j)}, 0\}$, into which we cannot move the summation and keep the bound.

So, we have that

$$\begin{aligned} s^{(j+1)} &\leq s^{(j)} - (1-\beta) s_{\mathcal{U}}^{(j)} f(d^{(j)}, \gamma^{(j)}) \\ &\leq s^{(j)} \left[1 - (1-\beta)(1-\delta) f(d^{(j)}, \gamma^{(j)})\right] \ . \end{aligned}$$

Over all periods in this subsequence,

$$\begin{aligned} s^{(fin+1)} &\leq s^{(init)} \prod_{j=init}^{fin} \left[1 - (1-\beta)(1-\delta) f(d^{(j)}, \gamma^{(j)})\right] \ , \\ \ln\left(\frac{s^{(fin+1)}}{s^{(init)}}\right) &\leq \sum_{j=init}^{fin} -(1-\beta)(1-\delta) f(d^{(j)}, \gamma^{(j)}) \ , \\ \sum_{j=init}^{fin} f(d^{(j)}, \gamma^{(j)}) &\leq \frac{\ln(s^{(init)}/s^{(fin+1)})}{(1-\beta)(1-\delta)} \ . \end{aligned}$$

In any period j, because WMNS doesn't update weights below $\beta\delta s^{(j)}/n$, we know that $w_i^{(init)} > \beta\delta s^{(init)}/n$. If we let $m_i = \sum_{j=init}^{fin} f(d^{(j)}, x_i^{(j)})$, we have by the lower bound on the update factor F:

$$\begin{aligned} s^{(fin+1)} &\geq w_i^{(fin+1)} \\ &\geq w_i^{(init)} \beta^{m_i}, \ \forall i \\ &\geq \frac{\beta\delta s^{(init)}}{n} \beta^{m_i}, \ \forall i \end{aligned}$$

Consequently, for all experts i:

$$\sum_{j=init}^{fin} f(d^{(j)},\gamma^{(j)}) \leq \frac{\ln\left(\frac{s^{(init)}}{s^{(fin+1)}}\right)}{(1-\beta)(1-\delta)}$$
$$\leq \frac{\ln\left(\frac{s^{(init)}}{\beta\delta s^{(init)}\beta^{m_i}/n}\right)}{(1-\beta)(1-\delta)}$$
$$= \frac{\ln\left(\frac{n}{\beta\delta}\right)+\ln\left(\frac{1}{\beta}\right)\sum_{j=init}^{fin} f(d^{(j)},x_i^{(j)})}{(1-\beta)(1-\delta)} .$$

Since this holds with respect to all experts, it also holds with respect to the best performing expert. Multiplying both sides by $\mathbf{C}$, we finally arrive at Theorem 1.

B Proof of Lemma 1

Suppose the static offline algorithm chooses a value which lies in the i^{th} bucket. The simulated expert who minimizes his difference in profit is the i^{th} expert, since regret increases as demand moves further from the expert's prediction, and each expert has the same regret at his bucket boundaries. For a single period, the true demand could fall in one of three places: below the bucket, in the bucket, or above the bucket.

If the demand d falls below the bucket ($d < q_{i-1}$), the maximum difference in profit occurs if the static algorithm has chosen the lowest point in the bucket at q_{i-1}. The difference in profit is then

$$dr - q_{i-1}c - (dr - x_i^{(j)}c) = \frac{c(M-m)(r-c)}{nr} \leq \frac{c(M-m)(r-c+c_u)}{n(r+c_u)} .$$

If the demand falls in the i^{th} bucket, we know from the definition of the buckets that the maximal difference in profit (which is now equivalent to minimax regret within this bucket, since the static algorithm can now predict the demand exactly) is the same thing. Similarly, if the demand falls above the i^{th} bucket, the worst case is when the static algorithm is at the top of the bucket at q_i, and the difference in profit can again be shown to be the same.

All three cases give identical worst case profit difference. Summing over all h periods, we have the lemma.

C Proof of Theorem 2

Theorem 1 bounds the regret of WMNS in terms of any expert (including the best expert) for any subsequence, even if the experts are intelligent and dynamic. Now we consider the case of our simulated experts from Section 3.2. Here let $initl$ and $finl$ actually define the l^{th} subsequence as chosen by SSTOPT, and $t_l = finl - initl + 1$ be the length of the subsequence.

Using the current notation, Lemma 1 implies that there exists an expert i such that

$$\sum_{j=initl}^{finl} (\mathsf{SSTOPT}^{(j)} - \mathsf{EX}_i^{(j)}) \leq \frac{c(M-m)(r-c+c_u)t_l}{n(r+c_u)} \ .$$

Using this implication and the following substitution,

$$\sum_{j=init}^{fin} (\mathsf{OPT}^{(j)} - \mathsf{EX}_i^{(j)}) = \sum_{j=init}^{fin} (\mathsf{OPT}^{(j)} - \mathsf{SSTOPT}^{(j)}) + \sum_{j=init}^{fin} (\mathsf{SSTOPT}^{(j)} - \mathsf{EX}_i^{(j)}) \ ,$$

we can modify the bound of Theorem 1 for the case of static, simulated experts:

$$\sum_{j=initl}^{finl} (\mathsf{OPT}^{(j)} - \mathsf{WMNS}^{(j)}) \leq \frac{\mathbf{C} \ln\left(\frac{n}{\beta\delta}\right)}{(1-\beta)(1-\delta)} + \frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c+c_u)t_l}{n(r+c_u)(1-\beta)(1-\delta)} + \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=initl}^{finl} (\mathsf{OPT}^{(j)} - \mathsf{SSTOPT}^{(j)})}{(1-\beta)(1-\delta)} \ .$$

Summing over all k subsequences used by SSTOPT, we arrive at Theorem 2 of Section 3.3.