

Comparing Online Learning Algorithms to Stochastic Approaches for the Multi-Period Newsvendor Problem

Shawn O’Neil and Amitabh Chaudhary <soneil,achaudha@cse.nd.edu>

Abstract

The multi-period newsvendor problem describes the dilemma of a newspaper salesman—how many papers should he purchase each day to resell, when he doesn’t know the demand? We develop approaches for this well known problem based on two machine learning algorithms: Weighted Majority of Warmuth and Littlestone, and Follow the Perturbed Leader of Kalai and Vempala. With some modified analysis, it isn’t hard to show theoretical bounds for our modified versions of these algorithms. More importantly, we test the algorithms in a variety of simulated conditions, and compare the results to those given by traditional stochastic approaches which assume more information about the demands than is typically known. Our tests indicate that such online learning algorithms can perform well in comparison to stochastic approaches, even when the stochastic approaches are given perfect information.

1 Introduction

On each morning of some sequence of days, a newspaper salesman needs to decide how many newspapers to order at a cost of c per paper, so that he can resell them for an income of r per paper. Unfortunately, every day it is unknown how many papers d will be demanded. If too many are ordered, some profits are lost on unused stock. If too few are ordered, some profits are lost due to unmet demand. The actual profit seen by a vendor who orders x items on a day with demand d is given by $r \min\{d, x\} - xc$. The papers ordered for a single day are of course only useful for that day; leftover papers cannot be sold in any later period.

This model describes a wide variety of products in industry. Fashion items and the trends they rely on are typically short lived, inducing many manufacturers to introduce new product lines every season[17]. Consumer electronics also have a short selling season due to their continuously evolving nature; cellular

phones can have a lifecycle as short as six months[2]. Some vaccines such as those for influenza are only useful for a single season[6].

For many such products, due to required minimum manufacturing or processing times, the vendor must finalize his order before any demand is seen. Further, because properties of the products themselves can vary markedly between selling periods, so too can the demand seen each period. This *demand uncertainty* is the most challenging hallmark of the newsvendor model.

A common approach taken to resolve the demand uncertainty issue is using a stochastic model for the demands; assuming, for example that for each period the demand is drawn independently from some known distribution. In using such an approach, the goal is then to choose an order amount which maximizes expected profit (see, e.g., [12]). However, such approaches are commonly inadequate, as the quality of the final result depends heavily on the quality of the assumptions made about the distribution. Given the strong uncertainty inherent in many newsvendor items, such quality is usually low. (See [22] for a lengthier discussion on the shortcomings of this approach.)

Alternate approaches to the newsvendor problem are more “adversarial” in nature. In these models, very little is assumed about the nature of the demands, and worst-case analysis is used. Typically, only a lower bound m and upper bound M on the range of possible demand values are assumed. One solution in this area develops a strategy to minimize the *maximum regret*:

$$\max_{\text{demand values}} (\text{OPT} - \text{ALG}),$$

where OPT denotes the profit of the *offline optimal* algorithm which knows the demand values, and ALG is the profit of the strategy used (see [18, 22, 23]).

Another method used to evaluate and design online algorithms for such problems is competitive ratio, where the goal is to minimize the ratio OPT/ALG in the worst case. However, one can show, using Yao’s

technique, a lower bound of $\Omega(M/(mk))$ for this ratio in the single period case when $r = kc$. This bound is tight, as a simple balancing algorithm can guarantee profits of this form.

Similarly restrictive results can be found for the worst case approach to regret with respect to OPT seen above. The single period *minimax regret* solution results in a maximum regret of $c(M - m)(r - c)/r$ [22], which implies that for t periods of a newsvendor game it is possible to suffer a regret of $tc(M - m)(r - c)/r$, even for the best possible deterministic algorithm.

For these reasons, we turn away from evaluating the performance of algorithms in terms of the *dynamic* offline optimal, and consider a more realistic target: the *static* offline optimal, which we denote here by STOPT. STOPT is a weaker version of OPT which makes an optimal decision based on perfect knowledge of the demands, but is required to choose one single order quantity to use for all periods.

Comparing the performance of algorithms with the performance of STOPT has practical significance, because any bounds for an algorithm with respect to STOPT also hold with respect to an algorithm which makes decisions based on stationary stochastic assumptions. Much of the inventory theory literature deals with algorithms of this type[16, 12].

We look at adaptations of two Expert Advice algorithms: Weighted Majority, developed by Littlestone and Warmuth[15], and Follow the Perturbed Leader, developed by Kalai and Vempala[13].

In the expert advice problem, the algorithm designer is given access to n experts, each of whom make a prediction for each period, and suffer some cost for incorrect predictions. The goal is to design an algorithm that makes its own predictions based on the experts' advice, and yet does not suffer much more cost than the best performing expert in hindsight.

In our setting, we use naive experts which make fixed predictions in the range $[m, M]$, and the cost they suffer in each period is the regret (difference in profit) from the dynamic offline OPT. Adapting the Weighted Majority algorithm to the non linear profit function of the newsvendor problem requires some careful attention if one wants to show theoretical performance bounds, whereas Follow the Perturbed Leader is a more straightforward implementation. Details of the algorithms' operation and theoretical performance bounds in this setting can be found in the appendices.

2 Goals of This Paper

In Section 4, we'll give overviews of the operation of three algorithms, two based on Weighted Majority variants which we call WMN and WMNS, and one based on Follow the Perturbed Leader which we call FPL. Each of these algorithms takes parameters which are chosen by the experimenter as input, which affect their operation and the performance bounds they achieve.

The primary interest of this paper, then, is to empirically evaluate the performance of these algorithms and compare the results to those generated by STOPT as well as more traditional stochastic approaches. Each of the stochastic solutions takes as input the assumptions made by the experimenter about the mean and standard deviation of the input distribution.

Further, the specifics of the problem instance itself may lead to interesting observations about all of the solutions specified. For instance, we know that the relationship of r and c can make a large difference on the performance of the minimax regret solution; does this ratio also affect the performance of other approaches we are going to test? Do certain types of input distributions favor one approach over the other?

Given such a large number of possible experimental variables, we are forced to select those which we believe will be most interesting, and design experiments using simulated data which are most likely to highlight the advantages and deficiencies of the different approaches.

3 Related Work

The Newsvendor Problem

The origins of the newsvendor problem can be traced as far back as Edgeworth's 1888 paper[10] in which the author considers how much money a bank should keep in 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 the expected profit is maximized at x , where $\phi(x) = (r - c)/r$ and $\phi(\cdot)$ is the cumulative probability density function for the distribution. Gallego's lecture notes[12] as well as the book by Porteus[16] have useful overviews. When only the mean and standard deviation are known, Scarf's results[19] give the optimal stocking quantity

which maximizes the expected profit assuming the worst case distribution with those two moments (a *maxi-min* approach). In some situations this solution prescribes ordering no items at all.

Among worst-case analyses, one of the earliest uses of the minimax regret criterion for *decision making under uncertainty* was introduced by Savage[18]. Applying the techniques to the newsvendor problem, Vairaktarakis describes adversarial solutions for several performance criteria in the setting of multiple item types per period and a budget constraint[22]. Bertsimas and Thiele give solutions for several variants of the newsvendor problem which optimize the order quantity based on historical data[3]. The solutions discussed take into account risk preferences by “trimming,” or ignoring, historical data which leads to overly optimistic predictions.

Learning from Experts

Weighted Majority is a very adaptable machine learning algorithm developed by Littlestone and Warmuth[15]. There are several versions of the weighted majority algorithm, including discrete, continuous, and randomized. Each consults the predictions of experts, and seeks to minimize the regret (in terms of prediction mistakes) with respect to the best expert in the pool.

Weighted Majority and variations thereof have been applied to a wide variety of areas including online portfolio selection[8, 7] and robust option pricing[9]. Other variants include the WINNOW algorithm also developed by Littlestone[14], which has been applied to such areas as predicting user actions on the world wide web[1].

Follow the Perturbed leader is a general algorithm for online decision making which is also applicable to the learning from experts problem. It’s creators, Kalai and Vempala[13], apply the algorithm to such problems as online shortest paths[21] and the tree update problem[20].

4 Algorithms

For these experiments, we impliment the following algorithms as described:

STOPT

This approach is given perfect information about the demand sequence, and chooses the single order quantity to use for all periods which maximizes the overall

profit (and thus also minimizes the total regret). As Bertsimas and Thiele discuss[3], the static offline optimal choice is the $\lceil t - t(c/r) \rceil^{th}$ order statistic of the demand sequence.

NORMAL

This stochastic solution assumes the demands will be drawn from a known normal distribution, and maximizes the expected profit. This approach prescribes ordering the amount $\mu + \sigma\phi^{-1}((r - c)/r)$, where $\phi^{-1}(\cdot)$ is the inverse of the standard normal cumulative distribution function[12].

SCARF

This stochastic solution is described in Scarf’s original paper[19] as well as in [12]. The solution maximizes the expected profit for the worst case distribution (a *maximin* approach in the stochastic sense) with first and second moments μ and σ . The order quantity is prescribed to be $\mu + \frac{\sigma}{2}(\sqrt{(r - c)/c} - \sqrt{c/(r - c)})$ if $c(1 + \sigma^2/\mu^2) < r$, and 0 otherwise.

MINIMAX

This is the *minimax regret* approach mentioned in Section 1. Described by [22], the algorithm orders the quantity $(M(r - c) + mc)/r$ for every period, which minimizes the maximum possible regret from the optimal for each period. As such, it also minimizes the maximum possible regret for the whole sequence.

The solution works by balancing the regret suffered by the two worst case possibilities: the demand being m or M . As such, its order never changes (as long as the range $[m, M]$ doesn’t change), and is very pessimistic in nature.

WMN

We develop this algorithm (Weighted Majority Newsvendor) as an adaptation of the Weighted Majority algorithm of Littlestone and Warmuth[15]. The algorithm takes two parameters: n , the number of “experts” to consult, and $\beta \in (0, 1]$, the weight adjustment parameter. Essentially, we divide up the range $[m, M]$ into n buckets, and have expert i predict the minimax regret order quantity for the i^{th} bucket. Buckets and experts are set up so that each bucket/expert pair has the same minimax regret.

As per the standard operation of Weighted Majority, each expert is given an initial weight of 1. After each round, we decrease each expert i ’s weight by

some factor F , where F depends on β and the regret that expert would have suffered on the demand seen using its prediction. If an expert is often wrong, its weight will be decreased faster than others. This punishment happens faster overall with smaller β 's.

The amount ordered by WMN in a given period is the weighted average of all experts. The intuition is that wherever the static optimal choice is, it must fall in one of the n buckets, and thus one of our experts will be close to this static optimal choice. Further, because experts' weights are decreased according to how poorly they do, the algorithm is able to learn where the static optimal choice is after a few periods, and even adapt to changing inputs over time.

Adapting the analysis of Weighted Majority to the non linear newsvendor profit function requires special care to ensure bounds similar to that of Weighted Majority can still be given. In Appendix A, we give a detailed description of WMN and a proof of the following theorem:

Theorem 4.1. *The total regret experienced by WMN for a t period newsvendor game with per item cost c , per item revenue r , and all demands within $[m, M]$ satisfies*

$$\begin{aligned} & \text{WMN}_{\text{TotalRegret}} \\ & \leq \frac{\mathbb{C} \ln(n)}{1-\beta} + \frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c)t}{nr(1-\beta)} \\ & \quad + \frac{\ln\left(\frac{1}{\beta}\right) \text{SSTOPT}_{\text{TotalRegret}}}{1-\beta} \end{aligned}$$

where $\mathbb{C} = \max\{(M-m)(r-c), (M-m)c\}$ is the maximum possible single period regret, n is the number of buckets used by WMN, and β is the update parameter used.

WMNS

WMNS, for Weighted Majority Newsvendor Shifting, is based on the "shifting target" version of the standard Weighted Majority algorithm. Here, if the input sequence can be decomposed into subsequences such that for each subsequence a particular expert does very well, then WMNS will do nearly as well for that subsequence. WMNS needs no information about how many shifts there will be, or when they will be. For example, if for the first third of the sequence all demands are near m , WMNS will initially adjust the weights of the experts so that it is ordering near m as well. If the sequence shifts so that demands are then drawn from near M , WMNS will adjust the weights quickly (quicker than WMN) so that the order quantities will match.

This ability comes from WMNS's use of a weight limiting factor $\delta \in (0, 1]$, so that no expert's weight will be less than δ times the average weight. When a new expert starts doing significantly better, the old best expert's weight is decreased to below the new expert's weight more rapidly, as the new expert's weight is guaranteed not to be too low in relation.

Theorem 4.2. *The total regret experienced by WMNS for a t period newsvendor game with per item cost c , per item revenue r , and all demands within $[m, M]$ satisfies*

$$\begin{aligned} & \text{WMNS}_{\text{TotalRegret}} \\ & \leq \frac{k\mathbb{C} \ln\left(\frac{n}{\beta\delta}\right)}{(1-\beta)(1-\delta)} + \frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c)t}{nr(1-\beta)(1-\delta)} \\ & \quad + \frac{\ln\left(\frac{1}{\beta}\right) \text{SSTOPT}_{\text{TotalRegret}}}{(1-\beta)(1-\delta)} \end{aligned}$$

where $\mathbb{C} = \max\{(M-m)(r-c), (M-m)c\}$ is the maximum possible single period regret, n is the number of buckets used by WMNS, β is the update parameter used, and δ is the weight limiting parameter used. SSTOPT is allowed to use a static optimal choice for k subsequences (i.e., is allowed to change order values $k-1$ times, see below).

A consequence of the $1/(1-\delta)$ factor in the bound is that WMNS does not perform as well as WMN when the best value for k is 1, i.e., SSTOPT happens to be no better than STOPT. Details of WMNS's operation and proof of the above bounds are given in Appendix B.

SSTOPT

This "optimal," which makes its decisions based on the entire sequence, is a slightly stronger version of STOPT, which is allowed to change its order quantity exactly $k-1$ times during the sequence.

FPL

Similar to WMN, FPL is based upon the Follow the Perturbed Leader approach developed by Kalai and Vempala[13]. As a general algorithm it is well suited to making decisions a number of times, when one wants to minimize the total cost in relation to the best single decision for all periods. Here, decisions will be of the form "use expert i 's prediction," where the experts again predict minimax values in buckets which divide the $[m, M]$ range. FPL as we use it takes two parameters, n , for the number of experts/buckets, and ϵ , which affects the final cost bound in relation to the best static decision.

Theorem 4.3. *The total regret experienced by FPL for a t period newsvendor game with per item cost c , per item revenue r , and all demands within $[m, M]$ satisfies*

$$\begin{aligned} \text{FPL}_{\text{TotalRegret}} & \\ & \leq \frac{4\mathbb{C}(1 + \ln(n))}{\epsilon} + \frac{(1 + \epsilon)c(M - m)(r - c)t}{nr} \\ & \quad + (1 + \epsilon)\text{STOPT}_{\text{TotalRegret}} \end{aligned}$$

where $\mathbb{C} = \max\{(M - m)(r - c), (M - m)c\}$ is the maximum possible single period regret, n is the number of buckets used by FPL, and ϵ is the randomness parameter used.

Details of the algorithm and proof of the bounds it gives in our application appear in Appendix C.

5 Experiments

In order to evaluate the online learning algorithms for the newsvendor problem, we run them on simulated demand sequences comparing the total regret suffered by each approach to the regret suffered by other the stochastic algorithms SCARF and NORMAL, as well as MINIMAX and STOPT.

Unless otherwise noted, all experiments consist of 100 demand newsvendor sequences, and each data point represents the average of 100 such trials. Thus, data points in the following figures typically represent the average total regret of various approaches on newsvendor sequences of length 100. Also, due to space limitations, we won't experiment with the affect of the upper and lower demand bounds $[m, M]$; we'll instead fix these bounds to $[10, 100]$ for all tests. Whenever a normal distribution is used, we restrict it to this range by resampling if a demand fall outside the range, and we further restrict all demands to be integers.

5.1 Algorithm Parameters

5.1.1 β , ϵ , and μ

For this first batch of tests, we investigate the performance of our three machine learning approaches while varying some of the parameters they accept as input. WMN and WMNS use β as a weight adjustment parameter: the smaller β is, the quicker expert weights are adjusted downward. WMNS also uses a "weight limiting" parameter δ , which we hold constant at 0.3 for these tests.

FPL uses the parameter ϵ , which affects the amount of "randomness" used in deciding which expert to follow. Smaller ϵ values lead to more randomness

being used. Even though the bounds discussed for FPL are only valid for $\epsilon \in (0, 1]$, the algorithm is still operable for larger values, so we test $\epsilon \in (0, 5]$. While we test the effects of varying β and ϵ , we hold the number of experts, n , at 32.

For Figures 1, 2, and 3, the distribution is normal with mean demand of 25 and standard deviation 15. Note that because the distribution is bounded to $[10, 100]$ via resampling, the actual mean of the distribution used is about 29.3. The per item cost c is held at 1, and the per item profit r is 4.

Figure 1 plots the average total regret of WMN, WMNS, and FPL as we vary β and ϵ . We also show the average total regret of STOPT as a baseline for comparison. One thing to notice in this figure is that while WMNS is adapted to be useful in situations where the distribution makes drastic changes over time, it does very nearly as well as WMN in this case.

On the other hand, even though FPL suffers a respectably low amount of regret, it's performance is only comparable to the other two approaches when rather large ϵ 's are used which aren't valid for theoretical analysis.

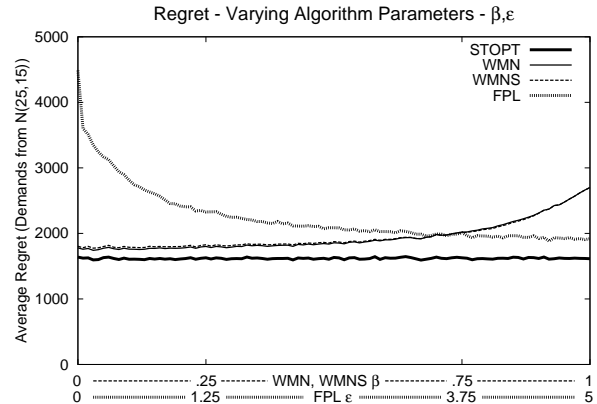


Figure 1: Average regret suffered on a 100 period newsvendor sequence, varying algorithm parameters. Even though ϵ must be less than or equal to 1 for FPL's theoretical bounds to hold, we see that in this situation it performs well with larger values also.

Figure 2 shows the regret of NORMAL and SCARF on the same test, varying the mean assumed about the demand distribution. Both approaches assume the correct standard deviation of 15. As this figure shows, the consequences of assuming incorrect information can be quite drastic for such stochastic algorithms.

In fact, it is interesting to look at the range of μ values used by NORMAL for which it suffers less regret than WMN. When WMN uses a β of 0.5, a rather naive choice, the average regret suffered is about 1856. NORMAL suffers less regret than this only when it assumes $\mu \in [21.7, 37]$, or within about 7.6 units on either side of the actual mean.

In Figure 2 we also plot the rather large regret suffered by the pessimistic worst case algorithm MINIMAX.

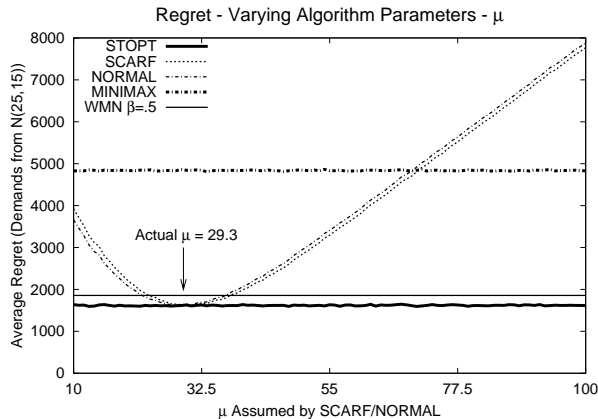


Figure 2: Average regret suffered on a 100 period newsvendor sequence, varying the mean assumed by NORMAL and SCARF. This plot shows the consequences to the stochastic approaches of assuming incorrect information. For comparison, we also plot WMN’s regret of 1856 when WMN uses a $\beta = 0.5$.

In Figure 3 we indicate what the theoretical bounds for WMN, WMNS, and FPL would be in the previous experiment. That is, given the values used for r, c, m, M, t , as well as the parameters used by the algorithms, we use the actual regret suffered by STOPT to compute the worst case regret for the algorithms no matter the input sequence. We see that the theoretical bounds are much higher than the actual empirical performance seen in figure 1, by as much as an order of magnitude.

As is reflected in Figure 3, the theoretical bounds given in Theorems 4.1 and 4.2 increase without bound as β is reduced to 0, because of the $\ln(1/\beta)$ term.

Because of this, we begin to notice the trade off between minimizing the theoretical bounds and getting good performance in actual simulation. (Later, in Figures 9 and 10, we’ll see the same phenomenon.) Similar to MINIMAX, the three experts algorithms give theoretical worst case bounds (though in relation to STOPT, rather than OPT), and as such have

a pessimistic nature to them as well. Using a β which decreases weights rapidly will quickly find the correct amount to order, however may be more susceptible to poor performance with very adversarial sequences.

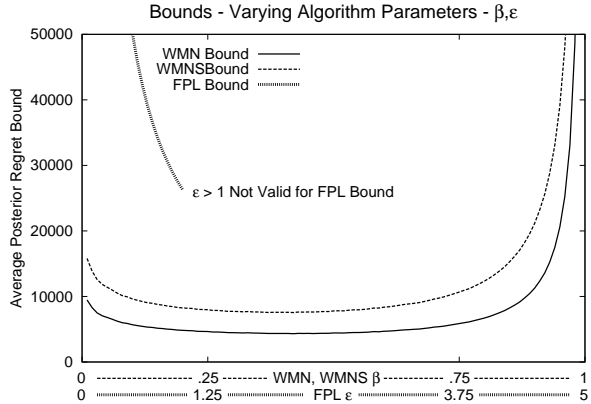


Figure 3: Average worst case theoretical bounds computed using algorithm parameters, problem parameters, and the actual regret suffered by STOPT. Comparing to Figure 1, we see a trade off between low worst case bounds and better empirical performance.

5.1.2 Number of Experts: n

Having looked at the effects of varying β and ϵ , we now turn our attention to the other main parameter of WMN, WMNS, and FPL: the number of experts/buckets used. Intuitively, using a larger value for n means that we are more likely to have an expert close to STOPT’s order value. On the other hand, all of the theoretical bounds grow as n becomes very large.

For Figures 4 and 5 we run the same test as section 5.1.1, with all demands drawn from $N(25,15)$ bounded to $[10,100]$. Here, WMN uses $\beta = 0.5$, WMNS uses $\beta = 0.5, \delta = 0.3$, and FPL uses $\epsilon = 0.75$.

Figure 4 plots the average total regret of the three algorithms varying the number of experts used from 1 to 100. Like the last test, WMN and WMNS perform remarkably similar, and FPL performs somewhat worse. In this plot, it appears that above a certain point, between 5 and 10, increasing the number of experts is ineffective. One possible reason for this is that because WMN and WMNS use the weighted average of experts, it is possible for them to settle upon an order quantity between two experts, making the number of experts somewhat less important. This cannot be the case for FPL, however, as FPL always goes with a single expert’s choice.

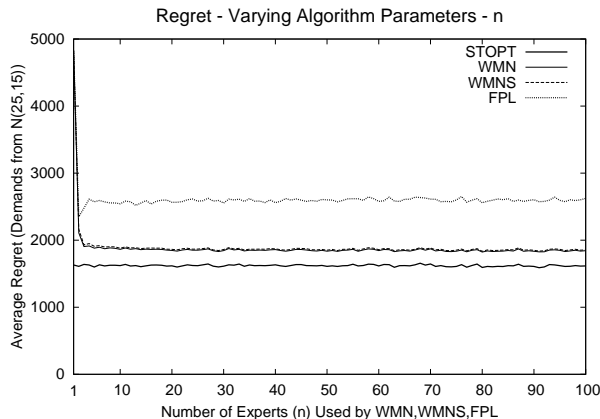


Figure 4: Average regret suffered while holding β , δ , and ϵ constant and varying the number of experts/buckets used n . While small values of n result in poor performance, past a certain point increasing the number doesn't help.

Figure 5 shows the computed theoretical bounds (in a manner similar to Figure 3) given by varying the number of experts for this test. Again, we see that a modest number of experts appears to be best, and increasing beyond this point has no benefit. Though it is difficult to see, there is a slight upcurve for WMN and WMNS toward the right side of the graph; theoretically, there will always be a minimizing value of n given a value for STOPT's regret.

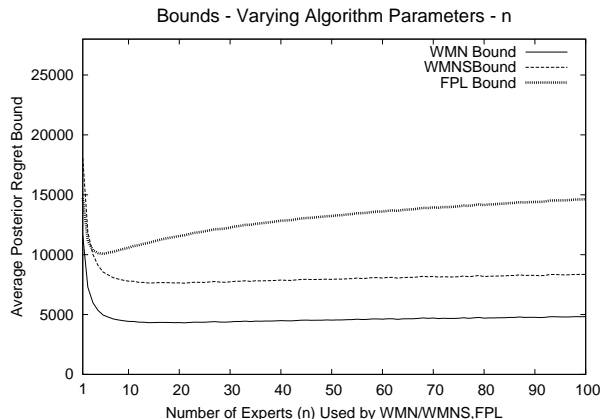


Figure 5: Theoretical bounds while varying the number of experts used, computed using the actual average regret suffered by STOPT.

5.2 Problem Parameters

Now we turn our attention to how the various approaches perform under different problem conditions. For all of the tests in this section, WMN uses a $\beta = 0.5$, WMNS uses $\beta = 0.5, \delta = 0.3$, and FPL uses $\epsilon = 0.75$. Then number of experts/buckets, n , used by all three is 32. Given the results so far, these seem to be typical naive choices which one might use in practice if no information about the problem is given.

In contrast, for all the tests in this section, we give SCARF and NORMAL the actual mean and standard deviation of the sequence to be used. Assuming such perfect information about the input distribution represents a best case for these; comparing with typical naive implementations of the experts algorithms should give some insight as to their real-world applicability.

5.2.1 Per Item Profit: r

Since we know that the worst case regret that can be suffered by MINIMAX depends on r and c , it will be interesting to look at a situation where we hold c constant to 1, and vary the per item profit r .

As in Section 5.1.1, all demands are drawn from the bounded normal $N(25,15)$. Figure 6 shows the average regret for the various algorithms as we increase the value of r from 1 to 10. Notice that when $r = 1$, the correct order quantity is 0, as no net profit is possible in this situation. STOPT, MINIMAX, and the stochastic approaches all take this into account, and as such suffer no regret. The experts algorithms on the other hand aren't given information about r and c , and must adjust their operation over time as they normally do.

Overall, as r increases with respect to c , the cost of poor decisions is amplified in comparison with OPT (which is how regret is measured). Thus, we see that all regret curves increase as r increases, with NORMAL and SCARF tracking STOPT most closely, followed by WMN and WMNS, whose plots nearly overlap.

5.2.2 Distribution: m, M Mix

Perhaps the most important consideration for a multi-period newsvendor algorithm is how well it deals with the inherent demand uncertainty. We've already looked at the effects of various algorithm parameters, but there we used a fairly "tame" distribution for demand values based on the normal. Here, we'll look at a somewhat more difficult distribution:

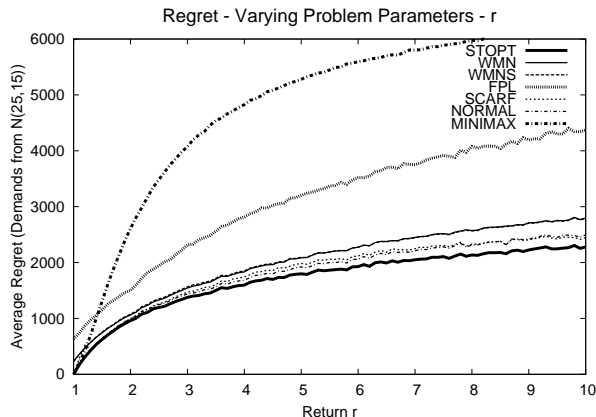


Figure 6: Average regret on sequences with demands drawn from $N(25,15)$ bounded to $[10, 100]$, varying r and holding c at 1. SCARF and NORMAL are given perfect information, while WMN, WMNS, and FPL use relatively naive operating parameters.

all 100 demand values will either be the minimum value m or the maximum value M . (Recall that these are 10 and 100, respectively.) Further, the sequences will be randomized so that where each type occurs is unknown. We still fix r to be 4 and c to be 1.

Figure 7 plots the regret of the WMN, WMNS, and FPL as we vary the number of minimum value m 's which appear in the sequence. Thus, at 0 on the left half of the graph, all demands in the sequence are M 's. In the middle at 50, each sequence is a random mixture of 50 m 's and 50 M 's.

In this figure we see a performance difference between WMN and WMNS, though this difference is still fairly small. All three algorithms do fairly well despite the large variance in demand values, tracking STOPT's regret in a somewhat linear fashion throughout the mix range.

Figure 8, which plots the regret of the stochastic approaches and MINIMAX, shows several interesting characteristics. There are a few points in the curve where SCARF and NORMAL perform as well as STOPT, but for much of the range they suffer a significant amount of regret in spite of the fact that they are working with perfect information about the actual mean and standard deviation. In comparison, the other algorithms perform similarly, if not better in some areas, given no a-priori information about the demand sequence.

MINIMAX manages the same regret for the entire range because whether a period demand is m or M , MINIMAX is designed to suffer the same regret.

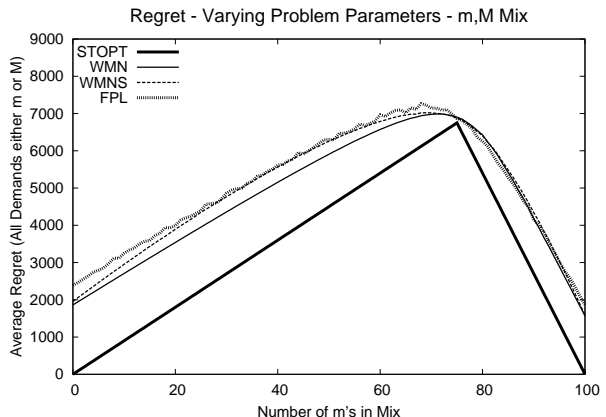


Figure 7: Average regret for m, M mix. Here, all demands in the 100 period sequence are either $m = 10$ or $M = 100$, and we vary how many of the demands are randomized.

STOPT experiences the most regret when there are $\lceil t - t(c/r) \rceil = 75$ minimum demands and 25 maximum demands, where its regret meets that of MINIMAX (as can also be shown algebraically).

5.2.3 Distribution: Shifting Normals, Algorithm Parameters Revisited

Finally, for this last set of tests, we explore if WMNS can perform better than WMN when the input is characterized by dramatic "shifts" in the demand sequence. Because WMNS employs a weight limiting factor δ , it is theoretically able to adjust the relative weights of the experts more quickly, and thus change decisions more rapidly.

Here, our sequence length is now 400 periods. The first 100 demands are drawn from $N(25,15)$, the second 100 are drawn from $N(75,15)$, the third 100 are again from $N(25,15)$, and the last 100 demands are drawn from $N(75,15)$. As usual, all demands are bounded to $[10, 100]$, so the true means of each subsequence are about 29.3 and 73.4, respectively.

Figure 9 plots WMN's regret when WMN uses $\beta = 0.5$, FPL's regret when $\epsilon = 0.75$, and WMNS's regret varying the δ used from 0 to .99. WMNS also used a constant β of 0.5. As we can see, up to a point increasing δ leads to less regret, such that WMNS can outperform STOPT and achieve regret closer to that of SSTOP. If δ is too high, however, we see an increase in regret, as fairly little weight adjustment is happening at all, limiting WMNS's learning ability.

The increase in performance, however, comes at a

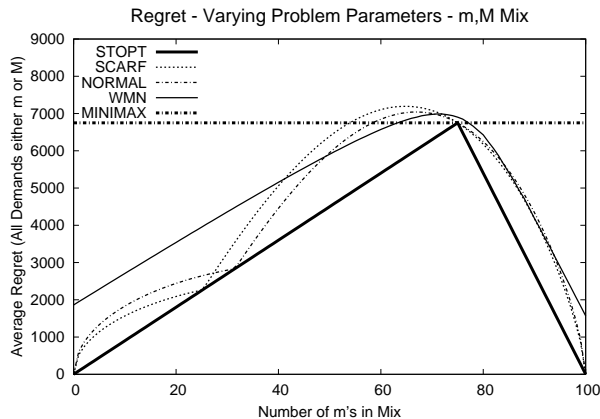


Figure 8: Average regret for m, M mix for the stochastic approaches as well as MINIMAX. Despite SCARF and NORMAL being given perfect information about the mean and standard deviation, they still suffer regret comparable to the other algorithms. (WMN’s regret using $\beta = 0.5$ is also shown for comparison.)

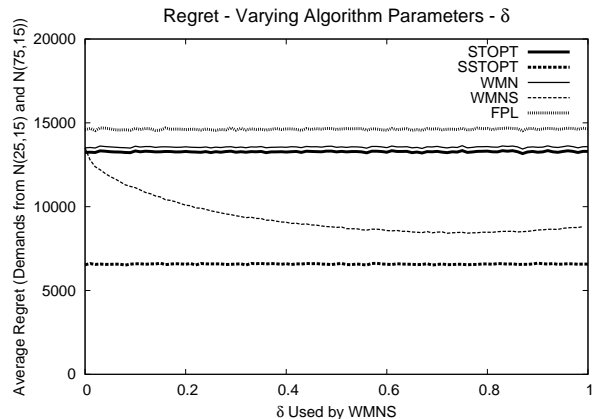


Figure 9: Regret of WMNS varying δ , the weight limiting parameter. For this plot, demand sequences were 400 periods long, with the first and third sets of 100 demands being drawn from $N(25,15)$, and the second and fourth being drawn from $N(75,15)$. SCARF and NORMAL, not shown in this plot, do approximately as well as STOPT given perfect information.

step price in terms of the theoretical regret bound, shown in Figure 10. bound for WMNS is computed from the actual average regret of SSTOP which used the single best static order value on each of the four subsequences. This increase in the computed bound happens because of the $(1 - \delta)$ term in the denominator of the bound (in Theorem 4.2).

6 Conclusion

We have presented an extensive empirical evaluation of the known stochastic approaches for the multi-period newsvendor problem in relation to new online learning approaches. The latter are unique in that they do not need any information about the demand distribution and have worst case performance bounds with respect to the static optimal solution.

Looking at all of the figures and discussion in aggregate, we see that overall WMN and WMNS perform comparably to the traditional stochastic approaches SCARF and NORMAL, even when those approaches are given perfect information about the demand distribution. When the stochastic methods assume incorrect information, they suffer as expected. This is significant, since there are many instances of real-world newsvendor applications where practitioners make large errors in determining or forecasting demand distributions. Fisher et al., for instance, report an average error of 55% in forecasting demand

for fashion apparel [11].

Though the bounds given for FPL are comparable to the bounds for WMN, the actual performance for this problem wasn’t as good, except for in the more difficult $[m, M]$ distribution mix scenario. Nevertheless, all algorithms significantly outperformed the bounds given for all tests that we ran.

We believe part of the reason for this is that the worst-case scenario in the general experts problem does not arise when using naive experts. In the general experts problem, it is possible to have a single expert perform well in a period while all other experts simultaneously suffer maximum regret. In the newsvendor setting, this is not possible, as the optimal order for a single period suffers no regret, and the regret suffered increases linearly as one looks at order quantities on either side. Designing an algorithm which exploits this fact will be the focus of future research.

Philosophically speaking, designing approaches which successfully balance the competing goals of good worst case performance and acceptable average case performance is one of the most interesting and challenging areas of online algorithms research. Sometimes, it seems to be necessary to further restrict the input criteria to achieve good average case results. Other times, simple extensions to an algorithm can improve average case results without sacrificing worst case performance, such as the THREAT

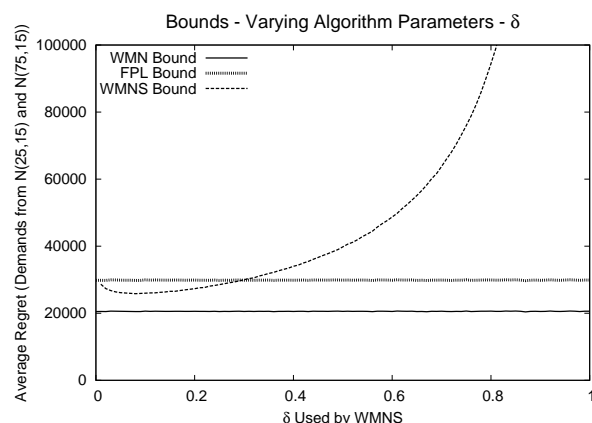


Figure 10: Regret bound of WMNS varying δ . The bound was computed in terms of SSTOPT’s regret, where SSTOPT was allowed to use the static optimal decision for each 100 period subsequence.

algorithm discussed in [4].

Of course, a solution isn’t worth much if no one uses it. Brown and Tang surveyed 250 MBA students and 6 professional buyers, supplying them with simple newsvendor problems[5]. Very few of the subjects used the classical newsvendor solution as prescribed by NORMAL, though the approach was known to almost all. One possible explanation given is that the classical solution doesn’t take into account risk preferences—buyers may be more comfortable underestimating demand to have a stronger guarantee on a particular profit rather than shoot for a higher profit with less certainty.

References

- [1] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Webwatcher: A learning apprentice for the world wide web. In *1995 AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments*, 1995.
- [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] D. Bertsimas and A. Thiele. A data driven approach to newsvendor problems. Technical report, Massachusetts Institute of Technology, Cambridge, MA, 2005.
- [4] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.
- [5] A. O. Brown and C. S. Tang. The single-period inventory problem: A new perspective. *Decisions, Operations, and Technology Management*, 2000.
- [6] S.E. 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.
- [7] T. M. Cover and E. Ordentlich. On-line portfolio selection. In *Proceedings of the ninth annual conference on Computational Learning Theory*, pages 310–313, 1996.
- [8] T. M. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2), 1996.
- [9] P. Demarzo, I. Kremer, and Y. Mansour. Online trading algorithms and robust option pricing. In *Proceedings of the thirty-eighth annual AMC Symposium on Theory of Computing*, pages 477–486, 2006.
- [10] F. Y. Edgeworth. The mathematical theory of banking. *Journal of the Royal Statistical Society*, 1888.
- [11] M.L. Fisher, A. Raman, and A.S. McClelland. Rocket-science retailing is almost here: Are you

ready? *Harvard Business Review*, 78(4):115–124, July-August 2000.

- [12] G. Gallego. Ieor 4000: Production management lecture notes. Available as <http://www.columbia.edu/~gmg2/4000/pdf/lect.07.pdf>.
- [13] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3):291–307, 2005.
- [14] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, pages 285–318, 1988.
- [15] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, pages 212–261, 1994.
- [16] E. L. Porteus. *Foundations of Stochastic Inventory Theory*. Stanford University Press, Stanford, CA, 2002.
- [17] 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.
- [18] L. J. Savage. The theory of statistical decisions. *Journal of the American Statistical Association*, 46:55–67, 1951.
- [19] H. E. Scarf. A min-max solution of an inventory problem. In *Stanford University Press*, 1958.
- [20] Daniel Dominic Sleator and Robert Endre Tarjan. Self-adjusting binary search trees. *J. ACM*, 32(3):652–686, 1985.
- [21] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *J. Mach. Learn. Res.*, 4:773–818, 2003.
- [22] C. L. Vairaktarakis. Robust multi-item newsboy models with a budget constraint. *International Journal of Production Economics*, pages 213–226, 2000.
- [23] G. Yu. Robust economic order quantity models. *European Journal of Operations Research*, 100(3):482–493, 1997.

A WMN Operation and Bounds

Definitions

In this section, we suppose we are going to play t periods of the newsvendor problem, with item cost c and item revenue r . We have access to n experts, each of which makes a prediction of the demand each period. (In Section A.1, we’ll discuss how WMN actually chooses experts’ predictions and give the final bounds.) In period j , each expert i predicts a demand of $x_i^{(j)}$ 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 . WMN (Weighted Majority Newsvendor) will aggregate the predictions of the experts, and order an amount $\gamma^{(j)}$ for the j^{th} period.

Clearly, the optimal choice for period j would be $d^{(j)}$, so the true dynamic offline optimal’s profit for period j , which we’ll denote as $\text{OPT}^{(j)}$, is $d^{(j)}(r - c)$. WMN’s profit is $\text{WMN}^{(j)} = \min\{d^{(j)}, \gamma^{(j)}\}r - \gamma^{(j)}c$, and the profit each expert i would have made is $\text{EX}_i^{(j)} = \min\{d^{(j)}, x_i^{(j)}\}r - x_i^{(j)}c$.

Algorithm WMN

Algorithm WMN operates as follows: each expert i is assigned an initial weight $w_i^{(1)} = 1$. Also, a weight adjustment parameter $\beta \in (0, 1]$ is chosen. In each period j , WMN orders an amount $\gamma^{(j)}$ which is the weighted average of the predictions of the experts: $\gamma^{(j)} = \sum_{i=1}^n w_i^{(j)} x_i^{(j)} / \sum_{i=1}^n w_i^{(j)}$.

In every period, after $d^{(j)}$ is revealed, we update each expert i ’s weight by some factor F : $w_i^{(j+1)} = w_i^{(j)} F$, where F satisfies

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

We use $f(d^{(j)}, x_i^{(j)}) = (d^{(j)}(r - c) - \min\{d^{(j)}, x_i^{(j)}\}r + x_i^{(j)}c) / \mathbb{C}$, where $\mathbb{C} = \max\{(M - m)(r - c), (M - m)c\}$ is the maximum possible regret any prediction can suffer. Choosing \mathbb{C} in this fashion guarantees that $0 \leq f(d^{(j)}, x_i^{(j)}) \leq 1$ for any valid $d^{(j)}$ and $x_i^{(j)}$, which allows us to ensure that such an update factor F exists[15]. Intuitively, $f(d^{(j)}, x_i^{(j)})$ gives a sense of the regret expert i would have suffered. In practice, we use the upper bound on F as the update factor.

Analysis

For clarity, we define $s^{(j)} = \sum_{i=1}^n w_i^{(j)}$ to be the total sum of weights over the experts in period j . To prove bounds on the total regret of WMN, we begin as in [15] by showing a bound on $\ln(s^{(t+1)}/s^{(1)})$. ($s^{(t+1)}$ is the sum of weights at the end of the game, $s^{(1)}$ is the sum of weights at the start.) Because of the upper bound on each update factor F , we have that $s^{(j+1)}$ is less than or equal to:

$$\begin{aligned}
& \sum_{i=1}^n w_i^{(j)} \left[1 - (1 - \beta) f(d^{(j)}, x_i^{(j)}) \right] \\
&= s^{(j)} - (1 - \beta) \sum_{i=1}^n w_i^{(j)} f(d^{(j)}, x_i^{(j)}) \\
&= s^{(j)} - (1 - \beta) \left[\sum_{i=1}^n \frac{w_i^{(j)} d^{(j)} (r - c)}{\mathbb{C}} \right. \\
&\quad \left. - \sum_{i=1}^n \frac{w_i^{(j)} \min\{d^{(j)}, x_i^{(j)}\} r}{\mathbb{C}} + \sum_{i=1}^n \frac{w_i^{(j)} x_i^{(j)} c}{\mathbb{C}} \right] \\
&= s^{(j)} - (1 - \beta) \left[\frac{s^{(j)} d^{(j)} (r - c)}{\mathbb{C}} \right. \\
&\quad \left. - \frac{r}{\mathbb{C}} \sum_{i=1}^n \min\{w_i^{(j)} d^{(j)}, w_i^{(j)} x_i^{(j)}\} + \frac{\gamma^{(j)} s^{(j)} c}{\mathbb{C}} \right].
\end{aligned}$$

We arrive at the last line by the definition of $s^{(j)}$ and $\gamma^{(j)}$. (Also, it must be noted that $d^{(j)}, x_i^{(j)}, w_i^{(j)} \geq 0$.) Now, by virtue of the fact that the summation over a minimum is less than or equal to the minimum of two summations, the above is less than or equal to:

$$\begin{aligned}
& s^{(j)} - (1 - \beta) \left[\frac{s^{(j)} d^{(j)} (r - c)}{\mathbb{C}} \right. \\
&\quad \left. - \frac{r}{\mathbb{C}} \min \left\{ \sum_{i=1}^n w_i^{(j)} d^{(j)}, \sum_{i=1}^n w_i^{(j)} x_i^{(j)} \right\} + \frac{\gamma^{(j)} s^{(j)} c}{\mathbb{C}} \right] \\
&= s^{(j)} - s^{(j)} (1 - \beta) \frac{1}{\mathbb{C}} \left[d^{(j)} (r - c) \right. \\
&\quad \left. - \min\{d^{(j)}, \gamma^{(j)}\} r + \gamma^{(j)} c \right] \\
&= s^{(j)} \left[1 - (1 - \beta) f(d^{(j)}, \gamma^{(j)}) \right].
\end{aligned}$$

So, over the entire sequence,

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

Going a step further, we have the beginnings of a bound on the total regret of WMN:

$$\sum_{j=1}^t f(d^{(j)}, \gamma^{(j)}) \leq \frac{\ln(s^{(1)}/s^{(t+1)})}{1 - \beta}.$$

Now we'll bound $s^{(t+1)}$. We let $m_i = \sum_{j=1}^t f(d^{(j)}, x_i^{(j)})$ be the total ‘‘adjusted regret’’ for expert i . By the lower bound on our update factor F :

$$\begin{aligned}
s^{(t+1)} &\geq \sum_{i=1}^n w_i^{(1)} \beta^{m_i} = \sum_{i=1}^n \beta^{m_i} \\
&\geq \beta^{m_i}, \forall i, \\
\ln(s^{(1)}/s^{(t+1)}) &\geq \ln(n) - m_i \ln(\beta), \forall i.
\end{aligned}$$

Combining this with the above, we finally get, for any expert i ,

$$\begin{aligned}
& \sum_{j=1}^t f(d^{(j)}, \gamma^{(j)}) \\
&\leq \frac{\ln(n) + \ln\left(\frac{1}{\beta}\right) \sum_{j=1}^t f(d^{(j)}, x_i^{(j)})}{1 - \beta}, \\
& \sum_{j=1}^t \left(\text{OPT}^{(j)} - \text{WMN}^{(j)} \right) \\
&\leq \frac{\mathbb{C} \ln(n)}{1 - \beta} + \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=1}^t \left(\text{OPT}^{(j)} - \text{EX}_i^{(j)} \right)}{1 - \beta}.
\end{aligned}$$

Thus, we have our total regret for WMN bounded in terms of the total regret for our best performing expert (since the bound holds for all experts).

A.1 Placing Experts in Buckets

So far, we have a bound on the regret of WMN in terms of the regret of the best expert. Now, we're interested in deriving a similar bound in terms of the regret of the offline static optimal, STOPT .

The approach we take is to let each of our n experts consistently predict a unique demand for all t newsvendor periods. We divide the overall range $[m, M]$ into n ‘‘buckets,’’ such that each bucket has the same minimax regret should the demand fall in that bucket. There are $n + 1$ bucket endpoints, $\{q_0, q_1, \dots, q_n\}$. As Vairaktarakis shows[22], for a given bucket i (with endpoints q_{i-1} and q_i) the minimax regret order quantity is $(q_i(r - c) + cq_{i-1})/r$, which results in a maximum regret of $(c(q_i - q_{i-1})(r - c))/r$ when the demand is at either endpoint.

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

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

We then let expert i consistently predict the optimal order quantity for the i^{th} bucket:

$$x_i^{(j)} = \frac{q_i(r - c) + cq_{i-1}}{r} = \frac{(ir - c)(M - m)}{rn} + m, \forall j.$$

Claim A.1. For a t -period newsvendor game, there exists an expert i such that the difference in i 's profit and any given static offline algorithm is at most

$$\frac{c(M-m)(r-c)t}{rn}.$$

Proof. Suppose the static offline algorithm chooses a value which lies in the i^{th} bucket. The 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

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

If the demand falls in the i^{th} bucket, we know from above that the maximal difference in profit (which is now equivalent to 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 t periods, we have the claim. \square

Since the claim holds for any static offline algorithm, it also holds for the static offline optimal algorithm, STOPT . Using the notation from Section A, the claim implies that there exists an expert i such that

$$\sum_{j=1}^t (\text{STOPT}^{(j)} - \text{EX}_i^{(j)}) \leq \frac{c(M-m)(r-c)t}{nr}. \quad (1)$$

Using the substitution

$$\begin{aligned} \sum_{j=1}^t (\text{OPT}^{(j)} - \text{EX}_i^{(j)}) &= \sum_{j=1}^t (\text{OPT}^{(j)} - \text{STOPT}^{(j)}) \\ &\quad + \sum_{j=1}^t (\text{STOPT}^{(j)} - \text{EX}_i^{(j)}), \end{aligned}$$

in the bound shown at the end of Section A, and the the bound implied by Equation 1, we have the following theorem:

Theorem A.2. The total regret experienced by WMN for a t period newsvendor game with per item cost c , per item revenue r , and all demands within $[m, M]$ satisfies

$$\begin{aligned} \text{WMN Total Regret} &= \sum_{j=1}^t (\text{OPT}^{(j)} - \text{WMN}^{(j)}) \\ &\leq \frac{\mathbb{C} \ln(n)}{1-\beta} + \frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c)t}{nr(1-\beta)} \\ &\quad + \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=1}^t (\text{OPT}^{(j)} - \text{STOPT}^{(j)})}{1-\beta} \end{aligned}$$

where $\mathbb{C} = \max\{(M-m)(r-c), (M-m)c\}$ is the maximum possible single period regret, n is the number of experts used by WMN, and β is the update parameter used.

The first term, which depends on the maximum possible single period regret, is independent of the number of periods. This gives the following corollary:

Corollary A.3. The average per period regret of WMN approaches

$$\begin{aligned} &\frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c)}{nr(1-\beta)} \\ &\quad + \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=1}^t (\text{OPT}^{(j)} - \text{STOPT}^{(j)})}{t(1-\beta)} \end{aligned}$$

as the number of periods becomes arbitrarily large.

B WMNS Operation and Bounds

In this section, we'll look at an extension of the results and provide an algorithm which does well in the face of an input sequence which can be decomposed into subsequences, where for each subsequence a different expert does well. This algorithm is analogous to the "shifting target" version of Littlestone and Warmuth's Weighted Majority algorithm. Using the same method of selecting experts as in Section A.1, we show a bound on the regret of the algorithm in terms of the "semi-static offline optimal" algorithm SSTOPT . SSTOPT is a version of STOPT described above which is allowed to change its choice $k-1$ times during the sequence.

B.1 Doing as Well as the Best Expert on any Subsequence

Definitions

We again consider playing t periods of a newsvendor game, with all demands drawn from $[m, M]$, per

item cost c , and per item revenue r . The algorithm described here, WMNS (Weighted Majority News vendor Shifting), will however operate slightly differently compared to WMN. These differences will allow us to partition the sequence of periods into subsequences, and show that WMNS will perform as well as the best expert *for each subsequence*, despite the fact that nothing is known about when the subsequences start or end or how many there are.

Aside from the definitions mentioned in Section A, we need to define two subsets of the n experts: \mathcal{UPD} , those which are “updatable”, and $\mathcal{!UPD}$, those which are not. As we will see, WMNS uses a weight limiting factor δ . For any period j , \mathcal{UPD} is defined as those experts whose weights satisfy $w_i^{(j)} > (\delta \sum_{i=1}^n w_i^{(j)})/n$. $\mathcal{!UPD}$ contains all other experts.

Algorithm WMNS

WMNS operates as WMN, with a couple of notable differences. First, a weight limiting parameter $\delta \in (0, 1]$ is chosen in addition to the weight update parameter $\beta \in (0, 1]$. Initially all experts’ weights are set to 1. In each period j , WMNS orders the amount $\gamma^{(j)}$ which is the weighted average of the predictions of the *updatable* experts: $\gamma^{(j)} = \sum_{i \in \mathcal{UPD}} w_i^{(j)} x_i^{(j)} / \sum_{i \in \mathcal{UPD}} w_i^{(j)}$.

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

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

Again, we use $f(d^{(j)}, x_i^{(j)}) = (d^{(j)}(r - c) - \min\{d^{(j)}, x_i^{(j)}\}r + x_i^{(j)}c) / \mathbb{C}$, where $\mathbb{C} = \max\{(M - m)(r - c), (M - m)c\}$.

Analysis

We start by showing that for any subsequence, WMNS performs nearly as well as the best expert for that subsequence. This means that if an expert does quite well for some subsequence, and then for another (later) subsequence another expert does quite well, WMNS will track the change quickly.

We let $s^{(j)} = \sum_{i=1}^n w_i^{(j)}$ be the total sum of weights of all experts in period j , $s_{\mathcal{UPD}}^{(j)} = \sum_{i \in \mathcal{UPD}} w_i^{(j)}$ be the sum of weights of updatable experts, and $s_{\mathcal{!UPD}}^{(j)} = \sum_{i \in \mathcal{!UPD}} w_i^{(j)}$ be the sum of weights of not updatable experts. We define *init* to be the index of the first

period of the subsequence, 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 , $w_i^{(j)} > \beta \delta s^{(j)}/n$. This is also true for the first period, because $1 > \beta \delta s^{(1)}/n = \beta \delta$.

By the bound on the update factor F and the mechanism of WMNS, we have that $s^{(j+1)}$ is less than or equal to:

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

We arrive at the last line by the definition of $s_{\mathcal{UPD}}^{(j)}$ and $\gamma^{(j)}$, as well as moving the summation inside of the min expression as in Section A. Next we need a lower bound for $s_{\mathcal{UPD}}^{(j)}$:

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

So, we have that

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

Over all periods in this subsequence,

$$s^{(fin+1)} \leq s^{(init)} \prod_{j=init}^{fin} [1 - (1 - \beta)(1 - \delta) f(d^{(j)}, \gamma^{(j)})] ,$$

$$\ln(s^{(fin+1)}/s^{(init)}) \leq \sum_{j=init}^{fin} -(1 - \beta)(1 - \delta) f(d^{(j)}, \gamma^{(j)}) ,$$

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

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 :

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

By substitution and rearrangement similar to that in Section A, we arrive at the following theorem:

Theorem B.1. *For any subsequence of newsvendor periods indexed from $init$ to fin and any expert i , WMNS's regret satisfies*

$$\begin{aligned} \text{WMNS Subseq. Regret} &= \sum_{j=init}^{fin} (\text{OPT}^{(j)} - \text{WMNS}^{(j)}) \\ &\leq \frac{\mathbb{C} \ln\left(\frac{n}{\beta\delta}\right)}{(1-\beta)(1-\delta)} + \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=init}^{fin} (\text{OPT}^{(j)} - \text{EX}_i^{(j)})}{(1-\beta)(1-\delta)} \end{aligned}$$

B.2 Doing as Well as SSTOP

SSTOP, the ‘‘Semi-Static Offline Optimal’’ algorithm is a slightly stronger version of STOP, which is allowed to change its order choice k times for the whole t period newsvendor game. Consider subsequence l , ($1 \leq l \leq k$), which is the subsequence where SSTOP is using its l^{th} choice. For this subsequence, SSTOP acts as a static offline optimal for periods from $init_l$ to fin_l , the beginning and ending indices of l . We define $t_l = fin_l + 1 - init_l$; the number of periods in subsequence l . (Thus, $\sum_{l=1}^k t_l = t$.) In essence, we are now considering k individual newsvendor games against different static optimal algorithms.

By defining experts according to the same construction of Section A.1, we can show that for any subsequence l ,

$$\begin{aligned} \sum_{j=init_l}^{fin_l} (\text{OPT}^{(j)} - \text{WMNS}^{(j)}) &\leq \frac{\mathbb{C} \ln\left(\frac{n}{\beta\delta}\right)}{(1-\beta)(1-\delta)} \\ &+ \frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c)t_l}{nr(1-\beta)(1-\delta)} \\ &+ \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=init_l}^{fin_l} (\text{OPT}^{(j)} - \text{SSTOP}^{(j)})}{(1-\beta)(1-\delta)}. \end{aligned}$$

Summing over all k subsequences (again, WMNS requires no knowledge of how long subsequences are, or even how many there are), we ultimately reach the following theorem:

Theorem B.2. *The total regret experienced by WMNS for a t period newsvendor game with per item cost c , per item revenue r , and all demands within $[m, M]$ satisfies*

$$\begin{aligned} \text{WMNS Total Regret} &= \sum_{j=1}^t (\text{OPT}^{(j)} - \text{WMNS}^{(j)}) \\ &\leq \frac{k\mathbb{C} \ln\left(\frac{n}{\beta\delta}\right)}{(1-\beta)(1-\delta)} \\ &\quad + \frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c)t}{nr(1-\beta)(1-\delta)} \\ &\quad + \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=1}^t (\text{OPT}^{(j)} - \text{SSTOP}^{(j)})}{(1-\beta)(1-\delta)} \end{aligned}$$

where $\mathbb{C} = \max\{(M-m)(r-c), (M-m)c\}$ is the maximum possible single period regret, n is the number of experts used by WMNS, β is the update parameter used, and δ is the weight limiting parameter used.

When $k = t$, then SSTOP is equivalent to OPT, though in this case the bound becomes useless because of the $k\mathbb{C}$ factor in the first term. When k is constant, however, we can note the following corollary:

Corollary B.3. *When the number of changes k SSTOP is allowed is constant, the average per period regret of WMNS approaches*

$$\begin{aligned} &\frac{\ln\left(\frac{1}{\beta}\right) c(M-m)(r-c)}{nr(1-\beta)(1-\delta)} \\ &+ \frac{\ln\left(\frac{1}{\beta}\right) \sum_{j=1}^t (\text{OPT}^{(j)} - \text{SSTOP}^{(j)})}{t(1-\beta)(1-\delta)} \end{aligned}$$

as the number of periods becomes arbitrarily large.

C FPL Operation and Bounds

FPL, for Follow the Perturbed Leader, was developed by Kalai and Vempala in [13]. (In this paper, they give two versions of the algorithm, FPL and FPL*. We use the latter for our problem.) In the experts setting, the algorithm keeps track of the total regret suffered by each expert. When a decision needs to be made, a random cost is added to each expert's sum regret so far, and FPL chooses the expert with the lowest overall regret.

Definitions

FPL takes as input a “randomness” parameter ϵ . For the bounds given to hold, ϵ must be in the range $(0, 1]$, however the algorithm will still operate with larger values.

Two other values are used by FPL and the bounds given in [13], A and D . A is defined as the maximum of the sum of all experts’ regret for a single period, and D is the maximum “diameter difference” between two decisions. (Because FPL is applicable to general decision making settings, these values have more a precise meaning which we won’t go into.) However, in Section 2 of [13], the authors note that for the experts problem, D is 1, and A is the maximum regret of a single expert for one period. In our case, this is then $A = \mathbb{C} = \max\{(M - m)(r - c), (M - m)c\}$.

Algorithm FPL

For each period, let s_i be the total regret suffered by expert i so far. For each expert i , choose a perturbation factor p_i from the exponential distribution with rate $\epsilon/2A$. The best perturbed expert so far then is $\operatorname{argmax}_i\{s_i + p_i\}$. Use the prediction of this expert.

Note that this algorithm is a specific, limited version of the general algorithm FPL*.

Bounds of FPL

Kalai and Vempala give the following theorem, which bounds the regret of FPL in terms of the regret of the best performing expert:

Theorem C.1. (Due to Kalai and Vempala.) *The expected regret of FPL satisfies*

$$E[FPL_{\text{regret}}] \leq (1 + \epsilon) \operatorname{Min}_{\text{Regret}} + \frac{4AD(1 + \ln(n))}{\epsilon}$$

Where $\operatorname{Min}_{\text{Regret}}$ is the regret of the best performing expert.

If we place experts in n buckets according to Section A.1, we know that the best expert won’t suffer more than

$$\frac{c(M - m)(r - c)t}{nr}$$

extra regret from the true static optimal on any t period newsvendor sequence. Using the notation of Appendix A, we can now give a theorem similar to Theorem A.2:

Theorem C.2. *The total regret experienced by FPL for a t period newsvendor game with per item cost c , per item revenue r , and all demands within $[m, M]$ satisfies*

$$\begin{aligned} \text{FPL Total Regret} &= \sum_{j=1}^t (\text{OPT}^{(j)} - \text{FPL}^{(j)}) \\ &\leq \frac{4\mathbb{C}(1 + \ln(n))}{\epsilon} + \frac{(1 + \epsilon)c(M - m)(r - c)t}{nr} \\ &\quad + (1 + \epsilon) \sum_{j=1}^t (\text{OPT}^{(j)} - \text{STOPT}^{(j)}) \end{aligned}$$

where $\mathbb{C} = \max\{(M - m)(r - c), (M - m)c\}$ is the maximum possible single period regret, n is the number of experts used by FPL, and ϵ is the randomness parameter used.

$\text{OPT}^{(j)}$, $\text{FPL}^{(j)}$, and $\text{STOPT}^{(j)}$ are the profits of OPT, FPL, and STOPT, respectively, in period j .