8-2016

Newsvendor Models With Monte Carlo Sampling

Ijeoma W. Ekwegh
East Tennessee State University

Newsvendor Models with Monte Carlo Sampling

A thesis

presented to

the faculty of the Department of Mathematics

East Tennessee State University

In partial fulfillment

of the requirements for the degree

Master of Science in Mathematical Sciences

by

Ijeoma Winifred Ekwegh

August 2016

Jeff Knisley, Ph.D., Chair

Ariel Cintron-Arias, Ph.D.

Robert Price, Ph.D.

Keywords: newsvendor models, Monte Carlo sampling, bootstrap resampling

ABSTRACT

Newsvendor Models with Monte Carlo Sampling

by

Ijeoma Winifred Ekwegh

The newsvendor model is used in solving inventory problems in which demand is random. In this thesis, we will focus on a method of using Monte Carlo sampling to estimate the order quantity that will either maximizes revenue or minimizes cost given that demand is uncertain. Given data, the Monte Carlo approach will be used in sampling data over scenarios and also estimating the probability density function. A bootstrapping process yields an empirical distribution for the order quantity that will maximize the expected profit. Finally, this method will be used on a newsvendor example to show that it works in maximizing profit.

ACKNOWLEDGMENTS

Foremost, I would like to thank God Almighty for bringing me this far and for giving me this wonderful opportunity.

I would also like to express my sincere gratitude to my thesis advisor Dr. Jeff Knisley, for his immense support of my Masters study and research, for his patience, motivation, enthusiasm, and extensive knowledge. His guidance and patience helped me in all the time of research and writing of this thesis.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Robert Price and Dr. Ariel Cintron-Arias, for their encouragement and support.

My sincere thanks also goes to all my professors that impacted knowledge to me and encouraged me in so many ways.

I would also like to thank my husband, Eric Akhiwu for his constant support and prayers. I appreciate my parents, Dr. and Mrs. Ekwegh for supporting me greatly and believing in me. To my siblings, God bless you all for your prayers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION AND BACKGROUND

A newsvendor model is used in solving inventory problems in which demand is random. The newsvendor model occurs whenever a decision must be made regarding the amount of a resource to have available prior to finding out how much is needed. Since the demand is random, we have to design a model that will help us know what quantity to order in order to maximize profit. The newsvendor problem is therefore the problem of deciding the size of a single order that must be placed before observing demand when there are overage and underage costs. The problem is particularly important for items with significant demand uncertainty and large overage and underage costs [6].

The issue of maximizing profit in the presence of random demand highly depends on revenue management [1]. An example of such a situation is the newsvendor problem, which is summarized in [14]. In fact, Bertsimas notes that "as one of the building blocks of inventory theory, it has received much attention in the literature, often under the assumptions that the demand distribution is known exactly and that the decision-maker is risk-neutral. In practice however, the unpredictable nature of the demand for most perishable products makes it difficult to obtain accurate forecasts" [1]. In Scarf's words [17], "we may have reason to conclude that the future demand will come from a distribution which differs from that governing past history in an unpredictable way". This fluctuation in demand incites the decision-maker to implement solutions, which will work out well for different actual demand outcomes. In 1958, Scarf [17] derived the optimal ordering quantity for the classical newsboy problem with mean and variance given, and his work was later extended by Gallego et. al [7, 12, 8]. In the news vendor problem, a manager sells a product during a particular season with the demand for the product being stochastic. In order for clients to have this product available during this season, the manager has to order his inventory before the selling season. If the order quantity is is lower than the demand, the manager might loose clients and profit. If the order quantity is more than the demand for products, the manager has to dispose of the remaining stock, which will bring loss to the business. As a result of this, there has to be a balance in the cost of ordering the too little and ordering too much.

The newsvendor problem is to find the optimal amount Q for a decision, where demand D is a random variable defined by the demand distribution and estimates of the distribution parameters. Given that $R(Q)$ is the revenue function or the profit function, $R(Q)$ is given as

$$R(Q) = \begin{cases} (p-c)Q & \text{if } Q \leq D \\ (p-c)D - c(Q-D) & \text{if } Q > D \end{cases}$$

The maximum revenue occurs when $Q = D$. In the newsvendor model, we either maximize $R(Q)$ or minimize $-R(Q)$. Since $R(Q)$ is a concave function, we find its optimal decision Q^* as

$$Q^* = argmax(R(Q)).$$

For $-R(Q)$ which is a convex function, we find its optimal decison Q^* as

$$Q^* = argmin(-R(Q)).$$

In order to estimate the order quantity Q, we introduce the concept of Monte Carlo sampling. Monte Carlo is the art of approximating an expectation by the sample mean of a function of simulated random variables [11].

In this thesis, we will use Monte Carlo sampling techniques to estimate the probabiility distribution of the order quantity of bread, which is used as an example. This will be done by sampling over given data and scenarios. We will also show that this method works in estimating the order quantity that maximizes the revenue of a newsvendor model.

1.1 Newsvendor Model

We begin by defining the concept of the news vendor model.

Definition 1.1 *The newsvendor model is a mathematical model in operations management used to determine optimal inventory levels. It is characterized by fixed prices and uncertain demand for a perishable product. If the inventory level is Q, each unit of demand above Q is lost in potential sales, and each level of demand below Q, is sold at the unit salvage price[12].*

The newsvendor problem occurs frequently in business and industry, because it allows decisions to be made about order quantity in the present even though future demand is uncertain.

All the newsvendor models have a common mathematical structure with the following elements:

- A Decision Variable (Q): The newsvendor problem is to find the value of Q that leads to an optimal decision. This value for Q is denoted by Q^*.

- Uncertain Demand (D): Demand is a random variable defined by the demand distribution and estimates of the parameters of the demand distribution.

- Unit Overage Cost (C_o): This is the cost of buying one unit more than the demand during a selling season. Also, $C_o = c - s$, where c is the unit cost and s is the unit salvage value (the value of an asset at the end of its life).

- Unit underage Cost (C_u): This is the cost of buying one unit less than the demand during a selling season. Also, $C_u = p - c$, where p is the unit price.

1.2 Applications of the Newsvendor Model

The newsvendor problem has a very broad range of applications. Here are some examples:

- Capacity Management:

  Other than determining stock levels, the newsvendor model can be used in making other decisions. For example, an automobile manufacturer must decide on how much manufacturing capacity to devote to each of its models for the approaching model year [13]. The overage cost includes the savings from providing one unit less capacity, and the underage cost includes the lost profit from not being able to make one more car of that type.

- One Stocking Decision for a Nonperishable Product:

  For products that require long term production and seasonal sale, the newsvendor model can be used for determining how much product to order. The overage cost is the unit materials, assembly, and transportation costs less end of season scrap value [13].

- Airline Overbooking:

Another application of the newsvendor model is airline overbooking of flights. In the airline company, there is a possibility that not all potential passengers who made flight reservations will show up for their flight. If one of the flights in a company has 50 first class seats and all the 50 are reserved, but only 35 passengers turn up to board their flight, then 15 seats will not generate revenue since they are empty. Also, if the company allowed 15 extra reservation to be made for the flight, and all the 65 passengers turn up, the problem of overbooking will arise.

The decision Q is the number of overbooked seats, and the demand D is the number of passengers who do not turn up. If $Q > D$, then there is a problem of overbooking, since the number of overbooked seats is more than the demand. In order to find a value for the number of overbooked seat Q to allow, such that an optimal decision will be made, the newsvendor model is applied. This also applies to hotels, restaurants, and other services that take reservations [13].

- Fashion Attire Retailers:

  The news vendor model is also applicable to fashion attire retailers who have to submit orders well before a selling season without an opportunity for replenishment during the season [18].

1.3 Monte Carlo Methods

Monte Carlo Methods are statistical methods used to find solutions to things such as expected values of a function, or integrating functions which cannot be integrated analytically [15]. A Monte Carlo method relies on our ability to randomly sample a variable given its probability distribution.

In order to understand the concept of Monte Carlo better, we define the following terms.

Definition 1.2 *A probability function P on a sample space Ω assigns to each event A in Ω a number $P(A)$ in $[0,1]$ such that*

1. *$P(\Omega) = 1$, and*

2. *$P(A \cup B) = P(A) + P(B)$ if A and B are disjoint.*

The number $P(A)$ is called the probability that A occurs [4].

Definition 1.3 *A random variable X is said to be continuous if its distribution function can be written as*

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt$$

where $f(t)$ is integrable over $\mathbb{R}$ and non-negative.

Definition 1.4 *The cumulative distribution function of a continuous random variable X is defined as*

$$F(x) = P(X \leq x) = \int_{-\infty}^{x} f(t)dt$$

Note that $F(x)$ is a non-decreasing continuous function [3].

This implies that for a given value x, $F(x)$ is the probability that the observed value of X will be at most x.

Definition 1.5 *Let X be a random variable whose distribution function F_X has a derivative. The function f_X satisfying*

$$F_X(x) = \int_{-\infty}^{x} f_X(t)dt$$

is called the probability density function [3].

The density function has two major properties:

- $f_X(x) \geq 0$ for all x in the state space.
- $\int f_X(t)dt = 1$.

Probability density functions can be used to determine the probability that a continuous random variable lies between two values, say a and b. This probability is denoted by $P(a \leq X \leq b)$ and is given by,

$$Pa \leq X \leq b = F_X(b) - F_X(a) = \int_a^b f_X(t)dt.$$

Theorem 1.6 *(Mean Value Theorem) If f is continuous on the closed interval $[a, b]$, then there exists a number c in the closed interval $[a, b]$ such that*

$$\int_a^b f(x)dx = f(c)(b - a)$$

In other words, the area under the curve is the same as the area of a rectangle over the same interval for some specific height, $f(c)$ [19].

Definition 1.7 *Two events are independent if the outcome of the first event does not affect the outcome of the second event, and the outcome of the second does not affect the first.*

Definition 1.8 *A sequence of random variables is independent and identically distributed (i.i.d.) if each random variable has the same probability distribution as the others and all are mutually independent [3].*

The principles of Monte Carlo methods are based on the strong law of large numbers defined below.

Theorem 1.9 *(The Strong Law of Large Numbers) If $X_1, X_2, \ldots, X_n,$ are independent and identically distributed random variables with $EX_k = \mu$ for $k = 1, 2, \ldots$ then*

$$P\left(\lim_{n\to\infty} \frac{\sum_{k=1}^{n} X_k}{n} = \mu\right) = 1$$

[9].

The law of large numbers says that the sample mean approaches the theoretical mean as the number of identically distributed, randomly generated variables increases.

Definition 1.10 *The expectation of a continuous random variable X with probability density function $f(x)$ is the number*

$$\mu = E(x) = \int_{-\infty}^{+\infty} x f(x) dx,$$

provided the integral

$$\int_{-\infty}^{+\infty} |x| \, f(x) dx$$

is finite.

Suppose we have an integral

$$I = \int_{a}^{b} g(x) f(x) dx$$

that needs to be evaluated. The Monte Carlo method is to take a random sample $X_1, X_2, \ldots, X_n$ from this distribution, and then form the mean

$$g_n = \frac{1}{n}\sum_{k=1}^{n} g(X_k).$$

From the strong law of large numbers, we have that

$$\frac{1}{n}\sum_{k=1}^{n} g(X_k) \to \int_a^b g(x)f(x)dx$$

with probability 1 [9]. For a better evaluation of the integral, we need to make n as large as possible.

2 DIFFERENT APPROACHES TO THE NEWSVENDOR MODEL

For newsvendor cost models, we minimize cost which is a convex function. Newsvendor profit models are maximization models, which are the negative of a minimization model. Hence, the newsvendor problem is to find the optimal amount Q for a one-time decision, where demand D is a random variable.There are different approaches to the newsvendor model, for better understanding, we discuss four of these approaches.

2.1 Newsvendor Model with a Known Density for Demand.

In this case, demand, which is a random variable, is defined by a given demand distribution from which we can obtain a density if necessary. This distribution could be a normal distribution, a Poisson distribution, a Gaussian distribution, etc. Also, demand may be either discrete (integer) or continuous [10].

In an optimization problem, we want to minimize the loss function or the cost function. The objective function is either a loss function or its negative (sometimes called a reward function, a profit function, a utility function, a fitness function, etc.) which is maximized. In the newsvendor model the cost function is minimized and the quantity, Q^* is the value at which the cost is minimized. The expected cost $ECost(Q)$, is a convex function, so its minimum occurs. We assume that D is a continuous random variable with density function $f(D)$ and cumulative distribution function $F(D)$. For order quantity Q and demand D, the cost is given by

$$Cost(Q,D) = \begin{cases} c_o(Q-D) & \text{if } D \leq Q \\ c_u(D-Q) & \text{if } D > Q \end{cases}$$

and the expected cost is given by

$$\begin{aligned} ECost(Q) &= \int_{D=0}^{\infty} Cost(Q,D)f(D)dD \\ &= c_o \int_{D=0}^{Q} (Q-D)f(D)dD + c_u \int_{D=Q}^{\infty} (D-Q)f(D)dD \\ &= c_oQ \int_{D=0}^{Q} f(D)dD - c_o \int_{D=0}^{Q} Df(D)dD \\ &\quad + c_u \int_{D=Q}^{\infty} Df(D)dD - c_uQ \int_{D=Q}^{\infty} f(D)dD \\ &= c_oQF(Q) - c_oT(Q) + c_u(\zeta - T(Q)) - c_uQ(1-F(Q)) \\ &= c_oQF(Q) + c_uQF(Q) - c_oT(Q) - c_uT(Q) + c_u\zeta - c_uQ \\ &= (c_u + c_o)(QF(Q) - T(Q)) + c_u(\zeta - Q) \end{aligned}$$

Here $F(Q)$ is the demand distribution function evaluated at Q and

$$\int\limits_{D=Q}^{\infty} Df(D)dD = \zeta - T(Q).$$

In order to obtain the derivative of $ECost$ with respect to Q, we use Leibniz's rule which is stated in theorem 2.1

Theorem 2.1 *(Leibniz's Rule) If f is continuous on [a, b] and if u(x) and v(x) are differentiable functions of x whose values lie in [a, b], then*

$$\frac{d}{dx} \int\limits_{u(x)}^{v(x)} f(t)dt = f(v(x))\frac{dv}{dx} - f(u(x))\frac{du}{dx}.$$

Since we know that $F'(Q) = f(Q)$ from definition 1.5 and by Leibniz's Rule, $T'(Q) = Qf(Q)$, where $T(Q) = \int\limits_{D=0}^{Q} Df(D)dD$ and $\zeta = \int\limits_{D=0}^{\infty} Df(D)dD$.

Taking the first derivative of $ECost(Q)$ with respect to Q, we have

$$\begin{aligned}\frac{dECost(Q)}{dQ} &= (c_u + c_o)(QF'(Q) + F(Q) - T'(Q)) - c_u \\ &= (c_u + c_o)(Qf(Q) + F(Q) - Qf(Q)) - c_u \\ &= (c_u + c_o)F(Q) - c_u\end{aligned}$$

Equating this derivative to zero, we have

$$F(Q) = \frac{c_u}{c_u + c_o}$$

where $\alpha = \frac{c_u}{c_u+c_o}$ is called the critical fractile or the critical ratio.

Taking the second derivative of $ECost(Q)$ we have

$$\frac{d^2ECost(Q)}{dQ^2} = (c_u + c_o)f(Q),$$

which is non-negative for all values of Q. Hence, $ECost$ is a convex function, which implies that it has a unique minimum. The critical fractile is gotten when the cost is at its minimum and $Q^* = F^{-1}\left(\frac{c_u}{c_u+c_o}\right)$ is the optimal order quantity [10].

2.2 Newsvendor Model with a Known Probability Distribution for Demand.

In the case of a newsvendor model with a known probability for demand, different probabilities are assigned to possible demand outcomes. Given $F(x)$, with finitely many outcomes, $S = \{1, 2, 3, 4, 5, 6...n\}$, we have that

$$F(x) = \sum_{s_i \leq x} Pr(S_i), \text{for} 1 \leq i \leq n.$$

In order to find the optimal order quantity when the probability distribution is known, we derive the following. The expected overage cost of order Q^* is given as

$$P(D < Q^*)c_o = F(Q^*)c_o.$$

The expected underage cost of order Q^* is given as

$$P(D > Q^*)c_u = (1 - F(Q^*))c_u.$$

The optimal quantity occurs where the underage cost is equal to the overage cost, so equating both we get

$$F(Q^*)c_o = (1 - F(Q^*))c_u.$$

Therefore, $P(D \leq Q^*) = F(Q^*) = \frac{c_u}{c_u+c_o}$. The probability of demand being satisfied is called the critical ratio or critical fractile.

Here is an example. Mr. John owns a Christmas tree shop and he has to place his orders for Christmas trees in October. Mr. John does not know what the exact demand for Christmas trees will be during the Christmas season, but notwithstanding he has to go ahead and place his order. The unit cost of each tree is \$35, the unit price for each tree before 25th of December is \$70 and after 25th of December the unit salvage value is \$20. If he orders too few, the unit underage cost (c_u) is $\$70 - \$35 = \$35$. If he orders too many, the unit overage cost (c_o) is $\$35 - \$20 = \$15$. Mr. John has only one chance to make his decision, in order to decide what quantity will minimize cost for him, he looks at the past data Christmas tree sales which is in Table 1.

In order to find out how much he can order, we calculate the cumulative probability distribution and assume that demand is discrete.

Table 1: Demand Distribution in the Past

Demand	200	400	600	800	1000	1200	1400	1600	1800	2000
Probability	0.05	0.10	0.15	0.10	0.10	0.20	0.15	0.05	0.05	0.05

Table 2: Demand Distribution in the Past with Cumulative Probabilities

Demand	200	400	600	800	1000	1200	1400	1600	1800	2000
Probability	0.05	0.10	0.15	0.10	0.10	0.20	0.15	0.05	0.05	0.05
F(D)	0.05	0.15	0.30	0.40	0.50	0.70	0.85	0.90	0.95	1.0

Plugging in values for the critical fractile,

$$\alpha = \frac{c_u}{c_u + c_o} = \frac{35}{35 + 15} = 0.7$$

From Table 2, 0.7 corresponds to the Christmas tree demand of 1200. Therefore the optimal order quantity is 1200, since the critical ratio determines the optimum order point.

Assuming we have a continuous demand uniformly distributed between 200 and 2000, with a mean of 1100 and a standard deviation of 200, we know that

$$P(D \leq Q^*) = \frac{c_u}{c_u + c_o},$$

so we want to find Q^*. The critical fractile remains 0.7, so $P(D \leq Q^*) = 0.7$ and the z-score is 0.52. From $z = \frac{Q-\mu}{\sigma}$, we find our order quantity, Q to be 1204.

2.3 A Newsvendor Model Where its Empirical Risk is Based on Data

Now let's consider a model of the form

$$E(x) = \int_{\mathbb{R}} L(x,u)\ f(u)du$$

The corresponding Empirical Risk is

$$Emp(x) = \frac{1}{N} \sum_{u_i \in Data} L\left(x, u_i\right)$$

where N is the number of data points.

For example, if our loss function (deterministic profit function for Newsvendor) with argument x as number of items ordered is

$$L(x,u) = \left(\begin{cases} px & if \quad x \leq D \\ (p-s)D + sx & if \quad x \geq D \end{cases} \right) - cx$$

where p is unit price, c is unit cost, s is unit salvage, and D is demand, then our Empirical Risk is

$$\begin{aligned} Emp(x) &= \frac{1}{N} \left(\sum_{u_i \in Data} \begin{cases} px & if \quad x \leq u_i \\ (p-s)u_i + sx & if \quad x \geq u_i \end{cases} - cx \right) \\ &= \frac{1}{N} \left(\sum_{u_i \in Data} \begin{cases} px & if \quad x \leq u_i \\ (p-s)u_i + sx & if \quad x \geq u_i \end{cases} \right) - cx \end{aligned}$$

where $F(x)$ is given by

$$F(x) = \int_{-\infty}^{x} f(u)du = \frac{p-c}{p-s}$$

2.4 Stochastic Linear Programming

In order to understand stochastic linear programming, we explain linear programming and stochastic programming.

A linear programming problem is the problem of maximizing or minimizing a linear function subject to linear constraints, where the constraints could be equality, inequality or a combination of both. The standard form of the linear program is

$$\begin{aligned} \min c^T x \\ \text{s.t. } Ax = b \\ x \geq 0 \end{aligned}$$

Stochastic programs are mathematical programs where some of the data incorporated into the objective or constraints are uncertain. When some of the data is random, then solutions and the optimal objective value to the optimization problem are themselves random [16].

The newsvendor model is an example of a stochastic programming problem with a continuous random variable. The major idea behind stochastic linear programming is the concept of recourse, also known as two-stage stochastic programing. Recourse implements a corrective action after a random event has occurred. For example, we choose a random variable, a, to control what happens now. After a random event has occurred, we take a recourse action, b, to correct anything that went wrong as a result of the random event.

To formulate a stochastic linear program, we minimize the cost of the first-period decision plus the expected cost of the second-period recourse decision.

$$\begin{aligned} \min \quad & c^T x + E_\omega \phi(x, \omega) \\ \text{s.t} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

where

$$\begin{aligned} \phi(x,\omega) \quad & = \min\ d(\omega)^T y \\ \text{s.t} \quad & T(\omega)x + W(\omega)y = h(w) \\ & y \geq 0. \end{aligned}$$

The first linear program minimizes the first-period direct costs, $c^T x$ plus the expected recourse cost, $\phi(x,\omega)$, over all of the possible scenarios and the first-period constraints, $Ax = b$. The recourse cost ϕ depends on x, the first-period decision and on the random event, ω.

The second linear program describes how to choose $y(\omega)$ (a different decision for each random scenario ω). It minimizes the cost $d^T y$ subject to some recourse function, $Tx + Wy = h$. This constraint can be thought of as requiring some action to correct the system after the random event occurs [2].

2.5 Monte Carlo Approach to the Newsvendor Model

The Monte Carlo approach will be used in this thesis, where we have that demand is often given as a data set. Depending on the business being considered, the demand data could be the number loaves of bread sold per day for several weeks; or number of automobiles of a certain type sold each week; or the number of people who make first class reservations in an airline company.

The purpose of this thesis is to design a Monte Carlo approach to the newsvendor problem in the case that demand is given as a data set. The Monte Carlo approach is broken down into three steps shown below.

1. Demand data is resampled with replacement. The generated sample has the same size as the original data. The goal is to use Monte Carlo Sampling to generate an empirical distribution for the optimal order quantity, which is the order quantity for which the objective function is optimal.

2. In order to achieve this goal, we need to generate scenarios. A scenario is a collection of representative values of the demand data with a corresponding probability for which these value occurs. These scenarios are constructed by partitioning the demand data, such that the scenarios represent all the data. For our model, we generate k scenarios from the generated sample.

3. After the scenarios are generated from the demand data set, the scenarios are used in a standard OR solver to obtain the order quantity. GLPK/MathProg will be used in this thesis as the standard OR solver. After solving, we obtain a corresponding probability, order quantity, sold quantity, salvage and profit. Since our goal is to use Monte Carlo sampling over demand to generate an empirical distribution for order data, we continually repeat this process to generate an empirical distribution.

Also, we use bootstrap resampling of the demand variable to generate scenarios for the stochastic linear program. The empirical distribution for order quantity is generated by solving for the argmin of the optimal objective for each scenario constructed from bootstrap reasmpling.

2.6 Bootstrap Resampling

In this thesis, the bootstrap resampling will be used more specifically as a Monte Carlo method. The basic idea of the bootstrapping method is that, in absence of any other information about the distribution, the observed sample contains all the available information about the underlying distribution, and hence resampling the sample is the best guide to what can be expected from resampling from the distribution [5]. Bootstrapping can refer to any test or metric that relies on random sampling with replacement. Bootstrapping allows assigning measures of accuracy (defined in terms of bias, variance, confidence intervals, prediction error or some other such measure) to sample estimates. This technique allows estimation of the sampling distribution of almost any statistic using random sampling methods.

Generally, to carry out the bootstrap resampling, we independently draw samples $D_1^*, D_2^*, D_3^*, \ldots, D_n^*$ of the same size from $D_1, D_2, D_3, \ldots, D_n$. These samples are called bootstrap samples. Also, these samples are identical and independently distributed from the empirical distribution of the real data with replacement, and as a result, in each sample some values in the data do not show up and some values show up more than once. Assuming we generate many samples, say Z in number, the empirical distribution of the resulting values $\hat{Q}_1^*, \hat{Q}_2^*, \hat{Q}_3^*, \ldots, \hat{Q}_Z^*$, is an approximation to the distribution function $\hat{Q}$. This sampling is done repeatedly with replacement, until a large number of new samples is obtained, and for every sample, the statistic of interest is recorded. As a result of the large samples drawn, we can generate an empirical sampling distribution for the statistic of interest, this distribution is called the bootstrap distribution. The bootstrap distribution is an estimate of the sampling distribution.

3 IMPLEMENTATION OF MONTE CARLO SAMPLING IN AN EXAMPLE OF A NEWSVENDOR MODEL

Before we go into details of the results of our newsvendor model, we will explain an example of the newsvendor problem we used in this thesis.

3.1 Description of our Newsvendor Model

Our goal is to estimate the quantity order that will maximize profit in the newsvendor model. As a result of the randomness of demand, our newsvendor model incorporates the two-stage stochastic programming. Our newsvendor example is the case of selling bread. We want make an optimum decision about the quantity of bread to order, so that we can minimize cost or maximize profit. The order for bread has to be made before we know what the actual bread demand will be. With respect to this example, we have the following parameters; unit price (r) = \$4, unit cost (c) = \$2, unit salvage value (w) = \$1, P_r = Probability, $R(Q, D)$ = Profit or revenue, ExD = Expected Demand, where the salvage value has to be less than the price and the cost less than the price $(w < c < r)$, $D(k)$ = demand, is a random variable and $y(K)$ = quantity sold. The expected demand, denoted as ExD, is defined as

$$\sum^{k} P_r(k)D(k),$$

where k represents the scenarios. Our objective function is to

$$\text{minimize } (-cx + r(y)k + w(x - y(k)))$$
$$\text{s.t. } y(k) \leq x, \; y(k) \leq D(k).$$

3.2 Implementation of Monte Carlo Sampling and Bootstrap Resampling in a Newsvendor Example

Our newsvendor model gives three forms of solution. The first one is the expected value for the mean solution.

The expected value of the mean solution (EVM) gives the least amount of profit that can be realized from our model. This solution is the least amount that can be made in the bread sales, thereby giving us a lower bound for the profit. Expected value of mean solution is given by

$$EVM = -c(ExD) + \sum^{k} P_r(k)(r(\min(ExD, D(k)) + w(\max(ExD - D(k), 0)))).$$

The second form of solution our model gives is the expected value with perfect information(EVPI). With the perfect information, we have complete and accurate information about the future demand. When we have the perfect information we would know in advance how many loaves of bread will be demanded each day, stock the exact amount of bread and sell all the bread. In order words, there is no salvage with the perfect information. The EVPI solution gives the upper bound on profit or the maximum profit that can be realized. Expected value of perfect information is given by

$$EVPI = \sum^{k} P_r(r - c)D(k).$$

The third form of solution our model is the expected value by the two Stage Stochastic programming (SP), this is of great interest to us in the newsvendor model. For the stochastic programming, we have the order quantity (x) to be the independent variable and the control variable. Actual sales for the different scenarios depends on the order quantity, this is because sales cannot be made once the quantity ordered is exhausted. Also the expected profit depends on the order quantity. The expected value for stochastic programming (expected profit) is given by

$$R(Q, D) = -cx + r(y)k + w(x - y(k)).$$

The optimum order quantity Q^* is given by

$$Q^* = argmax(E(R(Q, D)).$$

The expected value for stochastic programming is likely to fall between the expected value for mean and expected value of perfect information.

3.3 Consequence of Bootstrap Resampling on the Newsvendor Model

In our newsvendor example, we were given a data set that contained 100 values for bread demand. The demand data was resampled with replacement, and as a result of the replacement, some of the bread demand values occurred more than once and some values did occur at all in each sample. Since the demand data was between the range of 86 and 114, we generated 19 scenarios by partitioning the demand data. We used GLPK to using solve for each of these scenarios and got the results for the stochastic programming in Table 3.

From Table 3, we can observe that each scenario has a unique bread demand that corresponds to the partition of bread demand for the scenario. The bread demand was partitioned as follows:

Scenario 0: 85 to 86.5789

Scenario 1: 86.5789 to 88.157895

Scenario 2: 88.1578 to 89.736842

Scenario 3: 89.7368 to 91.3157

Scenario 4: 91.3157 to 92.8947

Scenario 5: 92.8947 to 94.4736

Scenario 6: 94.4736 to 96.0526

Scenario 7: 96.0526 to 97.6315

Scenario 8: 97.6315 to 99.2105

Scenario 9: 99.2105 to 100.7894

Scenario 10: 100.7894 to 102.3684

Scenario 11: 102.3684 to 103.9473

Scenario 12: 103.9473 to 105.5263

Scenario 13: 105.5263 to 107.1052

Scenario 14: 107.1052 to 108.6842

Scenario 15: 108.6842 to 110.2631

Scenario 16: 110.2631 to 111.8421

Scenario 17: 111.8421 to 113.4210

Scenario 18: 113.4210 to 115.

For each sample we generated, each scenario has a corresponding probability as seen in Table 3 also. From Table 3, we observe that while scenarios $5, 6, 9$ and 10 occur frequently, scenarios $0, 1, 3$ and 4 occurs less frequently and scenarios 16 and 17 do not occur at all. The values under the demand column are selected values for bread demand that falls into the partition for the corresponding scenario. The values in the order column are the same because for each sample we generate, the stochastic programming gives us an estimate of the order quantity. Since the order quantity

Table 3: Result From Stochastic Programming in GLPK

Scenario	Probability	Demand	Order	Sold	Salvage	Profit
0	0.02	85.7895	101.5789	85.7895	15.7895	155.79
1	0.01	87.3684	101.5789	87.3684	14.2105	160.53
2	0.04	88.9474	101.5789	88.9474	12.6316	165.26
3	0.03	90.5263	101.5789	90.5263	11.0526	170
4	0.01	92.1053	101.5789	92.1053	9.4737	174.73
5	0.12	93.6842	101.5789	93.6842	7.8947	179.47
6	0.15	95.2632	101.5789	95.2632	6.3158	184.21
7	0.05	96.8421	101.5789	96.8421	4.7368	188.95
8	0.08	98.4211	101.5789	98.4211	3.1579	193.68
9	0.12	100	101.5789	100	1.5789	198.42
10	0.11	101.5789	101.5789	101.5789	0	203.16
11	0.05	103.1579	101.5789	101.5789	0	203.16
12	0.06	104.7368	101.5789	101.5789	0	203.16
13	0.05	106.3158	101.5789	101.5789	0	203.16
14	0.02	107.8947	101.5789	101.5789	0	203.16
15	0.05	109.4737	101.5789	101.5789	0	203.16
16	0	111.0526	101.5789	0	101.5789	-101.58
17	0	112.6316	101.5789	0	101.5789	-101.58
18	0.03	114.2105	101.5789	101.5789	0	203.16

for each sample is constant, the values in the sold column gives us the amount of bread that will be sold in each scenario. In a situation where all the bread is sold, we maximize profit and in such case no salvage is involved. If all the breads are not sold, we have to sell the bread at the salvage cost, thereby not maximizing profit. In the scenarios that did not occur, all the bread that was ordered will be sold at the unit salvage cost, thereby giving us a negative profit.

Since we are interested in the optimal order quantity, we recorded the value of the order quantity for each resampling done. Bootstrap resampling was used in generating 100 samples for order quantity and expected value of profit.

From the bootstrap samples of the order quantity, we estimated some statistical parameters. The values of the mean, median, mode, standard deviation, variance, minimum and maximum of the order quantity were estimated as 103, 103, 103, 1.2596, 1.5866, 100, 105, respectively. The mean of the order data gives us an average value for the bootstrap samples of order quantity, which is a good approximation. The standard deviation and the variance, gives an idea of the variability of the order quantity samples about the mean. If the variance and the standard deviation is high, it tell us that the samples of order quantity are spread out over a wide range of values, if otherwise, we know that the samples of order quantity are close to the mean. Also, with the minimum and maximum, we can estimate the minimum and maximum quantity of bread we can order.

The Values of the mean, median, standard deviation, variance, minimum and maximum of the expected profit were also estimated as 192.7997, 192.9026, 1.2408, 1.5397, 189.6579, 195.5947, respectively.

From the samples of the order quantity, we plotted an empirical distribution for the order quantity, as shown in Figure 1. Figure 2 shows the empirical distribution for the Expected profit .

Figure 3 is a plot of the expected profit with the order quantity.

3.4 Result of the Model

The argmax of the newsvendor model is the order quantity that optimizes profit given demand as a random variable. From our example of the newsvendor model, we estimated the mean order data that will maximize profit to be 103. Also, from Figure 1, we can observe that the order data that will optimize revenue varies between

Figure 1: Empirical distribution for Order Data

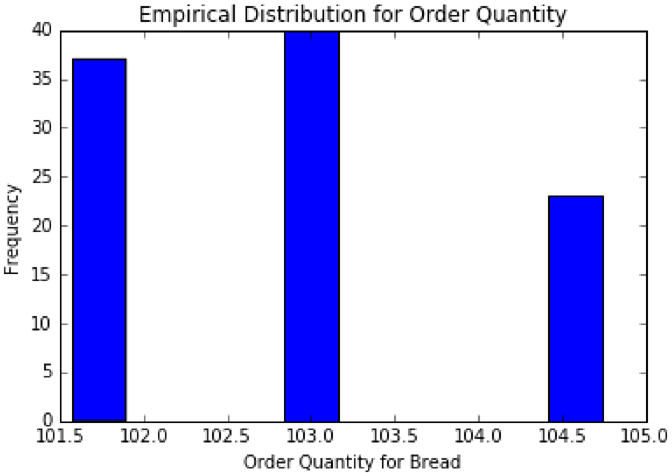


Figure 2: Empirical distribution for Profit Data

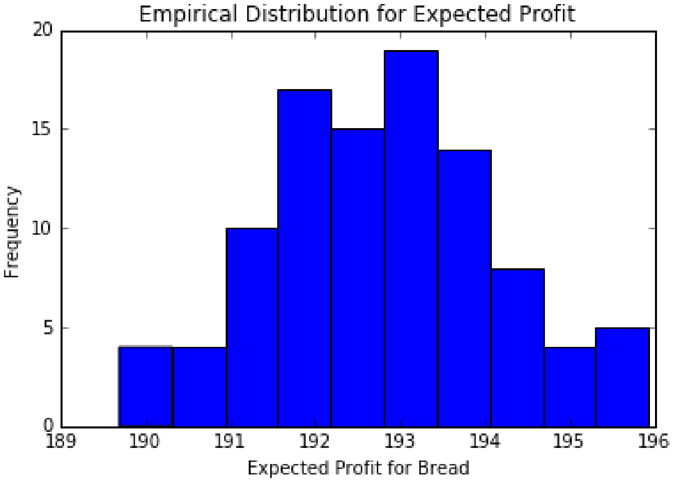
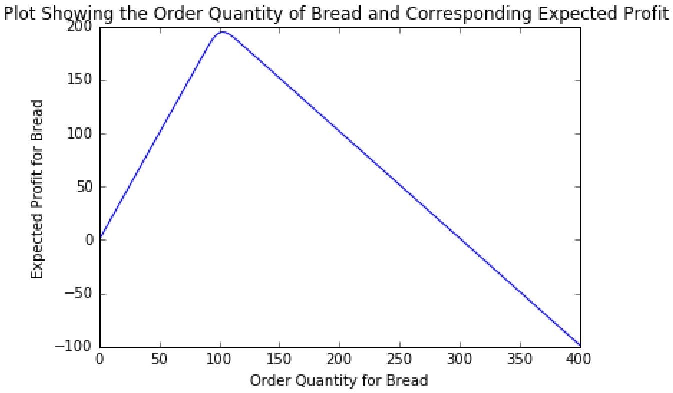


Figure 3: Plot of expected profit against order size



102, 103 and 105. From Figure 2, we can observe that a profit of 192.8026 is most likely to be made and a profit of 189.0895 is least likely to be made from the sales of bread. From Figure 3, we observe that profit is not maximized until the order size is about 100 to 104, and when the order size is about 105 and above, the expected profit starts to decrease until the values become negative. These results were produced from our implementation in Python and GLPK, which are included in the appendix.

BIBLIOGRAPHY

[1] Dimitris Bertsimas and Aurélie Thiele. A data-driven approach to newsvendor problems. Technical report, Technical report, Massechusetts Institute of Technology, Cambridge, MA, 2005.

[2] John R Birge and Francois Louveaux. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

[3] Paul Dawkins. Pauls online math notes. *Accessed October*, 11:2012, 2012.

[4] Frederik Michel Dekking. *A Modern Introduction to Probability and Statistics: Understanding why and how.* Springer Science & Business Media, 2005.

[5] Bradley Efron and Robert Tibshirani. An introduction to the bootstrap, monograph in applied statistics and probability, no. 57, 1994.

[6] Guillermo Gallego. 1 the newsvendor problem. 1995.

[7] Guillermo Gallego and Ilkyeong Moon. The distribution free newsboy problem: review and extensions. *Journal of the Operational Research Society*, pages 825–834, 1993.

[8] Guillermo Gallego, Jennifer K Ryan, and David Simchi-Levi. Minimax analysis for finite-horizon inventory models. *IIE Transactions*, 33(10):861–874, 2001.

[9] Dani Gamerman and Hedibert F Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference.* CRC Press, 2006.

[10] Arthur V Hill. The newsvendor problem. *White Paper*, pages 57–23, 2011.

[11] Christiane Lemieux. *Monte carlo and quasi-monte carlo sampling.* Springer Science & Business Media, 2009.

[12] Ilkyeong Moon and Guillermo Gallego. Distribution free procedures for some inventory models. *Journal of the Operational research Society*, pages 651–658, 1994.

[13] Evan L Porteus. *Foundations of stochastic inventory theory.* Stanford University Press, 2002.

[14] Evan L1990 Porteus. Stochastic inventory theory. dp heyman, mj sobel, eds. *Handbooks in operations research and management science*, 2:605–652, 1990.

[15] Richard E Quandt. Computational problems and methods. *Handbook of econometrics*, 1:699–764, 1983.

[16] Bhaskar Saha, Cuong C Quach, and Kai Goebel. Optimizing battery life for electric uavs using a bayesian framework. In *Aerospace Conference, 2012 IEEE*, pages 1–7. IEEE, 2012.

[17] Herbert Scarf, KJ Arrow, and S Karlin. A min-max solution of an inventory problem. *Studies in the mathematical theory of inventory and production*, 10:201–209, 1958.

[18] Maurice E Schweitzer and Gérard P Cachon. Decision bias in the newsvendor problem with a known demand distribution: Experimental evidence. *Management Science*, 46(3):404–420, 2000.

[19] George Brinton Thomas, Maurice D Weir, Joel Hass, and Frank R Giordano. *Thomas' Calculus Early Transcendentals*. Pearson, 2010.

APPENDIX: Python/GLPK Code

```
%pylab inline
from IPython.display import display
from pandas import DataFrame
from __future__ import division
import pandas as pd
import matplotlib.pyplot as plt

import subprocess

ls

BreadDemand = [100, 98,  86, 110,  99,  94, 114, 108, 100,  95, 100, 103, 104,
        91,  97, 110, 109,  99, 103, 108, 105, 114, 105, 102,  98, 105,
       110,  91,  94,  86,  98, 102,  93, 100,  93, 105, 107, 101, 104,
        95, 101,  96, 106, 105,  98,  90,  94, 102, 105,  96, 100,  95,
        98,  95, 100,  93, 106,  95,  89,  97, 103, 101, 105, 105, 100,
        92,  88,  89,  99, 101, 106,  89,  94, 102,  93, 107, 103,  95,
        95, 107, 105, 108,  96,  97,  95, 114, 101, 100, 100,  95,  96,
        97, 109, 109,  96, 101, 104,  93, 103,  97]

df = pd.DataFrame(BreadDemand, columns = ['Demands'])

Partition = linspace( min(BreadDemand)-1, max(BreadDemand)+1, 20)
Partition

Delta = Partition[1] - Partition[0]
Tags = [ p + Delta/2 for p in Partition ]
Tags

TaggedPartition = []
```

```
for i in range(1,len(Partition)):
    TaggedPartition.append( ( Partition[i-1], Partition[i], Tags[i-1], 0) )
TaggedPartition = pd.DataFrame(array(TaggedPartition), columns=[’LeftEndpt’,’RightEndpt’,’Tag’,’Count’])
TaggedPartition

TaggedPartition.Count = zeros(len(TaggedPartition))
samples = df.Demands.sample( len(df), replace = True )

for dat in samples:
    ind = TaggedPartition.RightEndpt.searchsorted(dat, side=’right’)
    TaggedPartition.Count[ind] += 1
TaggedPartition

TaggedPartition.index.tolist()

ScenariosData = pd.DataFrame( list(range(len(TaggedPartition))), columns = [’Scenario’] )
ScenariosData[’D’]  = TaggedPartition.Tag
ScenariosData[’Pr’] = TaggedPartition.Count/TaggedPartition.Count.sum()
ScenariosData

ls

import os

ScriptString = """
# Example: Newsvendor.mod

/* Unit Price Data */
param r >= 0;                            # Price
param c >= 0;                            # Cost
param w >= 0;                            # Salvage value
```

```
/* Price data makes sense only if  Price > Cost > Salvage */
check: c <= r;
check: w <= c;

/* Probabilistic Demand Forecast */
set SCENS;                                 # Scenarios
param D{SCENS} >= 0;                       # Demand
param Pr{SCENS} >= 0;                      # Probability

/* Fill the set/parameters with CSV data */
table Tab_SCENS IN "CSV" "scenarios.csv":
   SCENS <- [ Scenario ], Pr, D;

/* Probabilities must sum to one.
check: sum{k in SCENS} Pr[k] = 1; */

/* Expected Demand */
param ExD := sum{k in SCENS} Pr[k]*D[k];

/* Lower Bound on Profit: Expected Value of the Mean Solution */
param EVM := -c*ExD + sum{k in SCENS} Pr[k]*(r*min(ExD,D[k])+w*max(ExD-D[k],0));

/* Upper Bound on Profit: Expected Value with Perfect Information */
param EVPI := sum{k in SCENS} Pr[k]*(r-c)*D[k];

/* Two Stage Stochastic Programming */
var x >= 0;                     # Stage 1 (Here and Now): Order Quqntity
var y{SCENS}>= 0;               # Stage 2 (Scenario Dep): Actual Sales
var ExProfit;                   # Expected Profit
```

```
/* Maximize Expected Profit */
maximize OBJ: ExProfit;

/* Goods sold are limited by the order quantities and the demand  */
s.t. PRFT: ExProfit = -c*x + sum{k in SCENS} Pr[k]*(r*y[k] + w*(x-y[k]));
s.t. SUPL {k in SCENS}: y[k] <= x;
s.t. DMND {k in SCENS}: y[k] <= D[k];

solve;


table Table_EVM {k in SCENS} OUT "CSV" "evm.csv" "Table":
   k~Scenario,
   Pr[k]~Probability,
   D[k]~Demand,
   ExD~Order,
   min(ExD,D[k])~Sold,
   max(ExD-D[k],0)~Salvage,
   -c*ExD + r*min(ExD,D[k]) + w*max(ExD-D[k],0)~Profit;

table Table_EVPI {k in SCENS} OUT "CSV" "evpi.csv" "Table":
   k~Scenario,
   Pr[k]~Probability,
   D[k]~Demand,
   D[k]~Order,
   D[k]~Sold,
   0~Salvage,
   -c*D[k] + r*D[k]~Profit;

table Table_SP {k in SCENS} OUT "CSV" "evsp.csv" "Table":
   k~Scenario,
   Pr[k]~Probability,
   D[k]~Demand,
   x~Order,
   y[k]~Sold,
   x-y[k]~Salvage,
   -c*x + r*y[k] + w*(x-y[k])~Profit;

data;

/* Bread Parameters: buy for $2, sell for $4, salvage for $1 */

param r := 4.00;                         # Unit Price
param c := 2.00;                         # Unit Cost
param w := 1.00;                         # Unit Salvage Value

end;
"""

from IPython.display import display, clear_output

OrderData  = []
ProfitData = []
for i in range(100):
    TaggedPartition.Count = zeros(len(TaggedPartition))
    Sample = df.Demands.sample( len(df), replace = True )

    for dat in Sample:
        ind = TaggedPartition.RightEndpt.searchsorted(dat, side=’right’)
        TaggedPartition.Count[ind] += 1
    TaggedPartition
    ScenariosData = pd.DataFrame( list(range(len(TaggedPartition))), columns = [’Scenario’] )
    ScenariosData[’D’]  = TaggedPartition.Tag
    ScenariosData[’Pr’] = TaggedPartition.Count/TaggedPartition.Count.sum()

    ScenariosData.to_csv(’scenarios.csv’, index=False)

    get_ipython().run_cell_magic(’script’, ’glpsol -m /dev/stdin’, ScriptString)
    clear_output( )

    evsp = pd.read_csv("evsp.csv")
    OrderData.append(evsp.Order[0])
    ProfitData.append( sum(evsp[’Probability’]*evsp[’Profit’] ) )
    #if( not i % 50 ): print(’’)
    #print(’.’*i, end=’’)

OrderData

hist(OrderData)

mean(OrderData)

hist(ProfitData)

evm = pd.read_csv("evm.csv")
display(evm)

ev_evm = sum(evm[’Probability’]*evm[’Profit’])
print( "Expected Profit for the Mean Scenario = {:6.2f}".format(ev_evm) )

evpi = pd.read_csv("evpi.csv")
display(evpi)
```

```
ev_evpi = sum(evpi['Probability']*evpi['Profit'])

print( "Expected Profit with Perfect Information = {:6.2f}".format(ev_evpi) )


evsp = pd.read_csv("evsp.csv")

display(evsp)


ev_evsp = sum(evsp['Probability']*evsp['Profit'])

print( "Expected Value by Stochastic Programming = {:6.2f}".format(ev_evsp) )


print( "Value of Perfect Information = {:6.2f}".format(ev_evpi-ev_evsp) )


print( "Value of the Stochastic Solution = {:6.2f}".format(ev_evsp-ev_evm) )


r = 4.00

c = 2.00

w = 1.00


def profit(D,x):

    return r*min([D,x]) + w*max([0,x-D]) - c*x


def exprofit(x):

    v = 0

    for s in ScenariosData.as_matrix():

        v += s[2]*profit(s[1],x)

    return v


x = linspace(0,400,400)

exprofit(106)


x = linspace(0,400,400)
```

```
y2 = list(map(exprofit,x))

plot(x,y2)

xlabel('Order size')

ylabel('Expected Profit')
```

VITA

IJEOMA WINIFRED EKWEGH

Education:	B.S. Mathematics, Michael okpara university, Abia state, Nigeria 2011
	M.S. Mathematical Sciences, East Tennesse State University Johnson City, Tennessee 2016
Professional Experience:	High School Teacher, Chokhmah International Academy, Port Harcourt, Nigeria, 2012–2014
	Graduate Assistant, East Tennessee State Johnson City, Tennessee, 2015–2016