A Solution to The Shopping List Problem

Frank Rodriguez

August 3rd, 2018

Abstract

An arbitrary shopping list can be optimally satisfied from a suitable stock list by analyzing all possible consequences of opening each container. These consequences can form a tree of potential decisions for the next container to purchase. By recursively calculating the average cost of each node in the tree, one can prune the tree of all suboptimal paths. This reveals the optimal "purchasing plan" for any outcome, and by starting from the container with the lowest average tree cost, one will optimally satisfy the entire list.

1 Definition

Assume we have a shopping list of items which need to be purchased. The list contains pairs of items and quantities. Here is an example structure:

$$L = [(i_1, q_1)), (i_2, q_2), (i_3, q_3) \ldots (i_n, q_n)]$$

We will also have a stock list which contains pairs of containers and their prices:

$$S = [(c_1, p_1), (c_2, p_2), (c_3, p_3) \ldots (c_n, p_n)]$$

Each container will have several items which each have an *independent* probability of being present in the container. In other words, if a container has a $\frac{1}{2}$ chance of containing a single apple, and a $\frac{3}{4}$ chance of containing a single orange, then there are appropriate odds of the container having either an apple, an orange, or both. They also have an absolute quantity. If a container may contain 6 apples, then there will either be exactly 6 or exactly 0 apples inside.

Containers can thus be represented as a list of 3-tuples of items, quantities, and probabilities:

$$C = [(t_1, u_1, r_1), (t_2, u_2, r_2), (t_3, u_3, r_3) \ldots (t_n, u_n, r_n)]$$

The goal of this paper is to outline a method for optimally satisfying any given shopping list with an arbitrary stock list.[1]

[1]That is, unless the stock list simply does not have the requested items.

2 Prerequisite Calculations

2.1 Consequences

Consider a container $\left[\left(x,1,\frac{1}{2}\right),\left(y,1,\frac{1}{2}\right)\right]$. There are a few potential outcomes when opening it:

- We can get nothing
- We can get a single x
- We can get a single y
- We can get both a single x and a single y

These potential outcomes will be referred to as the *consequences* of opening the container. Consequences will influence further decisions.

2.2 Consequence Probabilities

In the above example, while there is certainly a $\frac{1}{2}$ chance of finding an x in the container, there is *not* a $\frac{1}{2}$ chance of finding *only* an x. To get the odds of each consequence, we can visualize a table of rows, where each potential item makes up a row:

x	N
y	N

Figure 1: One row per potential item.

Probabilities can be found via combinations of taking a single item from each row, where in this case N means "nothing". For an item i with a probability of $\frac{a}{b}$, there will be a i's and $b-a$ N's.

In this case, the possibilities are:

$x \to y$	$x \to \mathrm{N}$
$\mathrm{N} \to y$	$N \to N$

Figure 2: Possible selections. If there are n rows, each selection will have n elements.

One possibility contains a single x, one contains a single y, one contains both an x and a y, and one has nothing. Thus in this case our consequences are:

$\frac{1}{4}$ Nothing	$\frac{1}{4}$ $1\ x$
$\frac{1}{4}$ $1\ y$	$\frac{1}{4}$ $1\ x,\ 1\ y$

Figure 3: Final probabilities taken by combining selections.

Counting is tedious. Multiplication can be used along with some search instead. Consider a container $\left[\left(x, 1, \frac{2}{5}\right), \left(y, 2, \frac{1}{2}\right), \left(z, 3, \frac{1}{3}\right)\right]$. This has 3 potential items. Constructing a table leads to:

x	x	N	N	N
y	N			
z	N	N		

Figure 4: Rows for $\left[\left(x, 1, \frac{2}{5}\right), \left(y, 2, \frac{1}{2}\right), \left(z, 3, \frac{1}{3}\right)\right]$.

We can condense this:

2x	3N
1y	1N
1z	2N

Figure 5: Condensing the rows.

Given the input container and its probabilities, we can tell what the consequences will be. In this case, we can see that we will either get:

$1x$	OR	$1x,\ 2y$	OR	$1x,\ 2y,\ 3z$
$2y$	OR	$2y,\ 3z$	OR	$3z$
	OR	Nothing		

Figure 6: All possible consequences from figure 5.

Finding the "just x" consequence requires multiplying the coefficient of x by the coefficients of the N's in the remaining rows. Thus, there are $3 \cdot 1 \cdot 2$, or 6 selections containing a single x. This forms the numerator, where the denominator is the total combinations, or $5 \cdot 2 \cdot 3$. Thus, the odds of getting a single x from this container are $\frac{6}{30}$, or $\frac{3}{10}$. The rest are calculated similarly.

The potential consequences themselves can be derived from this condensed table, as well. If any row has only a single item, then all rows above it must include it.

3 Consequence Trees

3.1 Definition

A *consequence tree* will be a complete representation of every way in which a shopping list can be satisfied from a given container. Each node represents a container, with branches for each significant consequence of that container. The branches represent a change in the shopping list and themselves connect to all available containers which are significant to this new shopping list.

3.2 Context

A consequence tree is contextual and depends on both the shopping list and stock list for construction. We will construct a tree in the context of these lists:

$$\begin{aligned} L = [\,&(x, 1)\,, \\ &(y, 1)] \end{aligned}$$

Figure 7: The example shopping list.

$$\begin{aligned} S = \Bigg[\, &\left(\left[\left(x, 1, \frac{1}{2}\right)\right], 10\right), \\ &\left(\left[\left(y, 1, \frac{1}{2}\right)\right], 10\right), \\ &\left(\left[\left(x, 1, \frac{1}{3}\right), \left(y, 1, \frac{1}{2}\right)\right], 20\right), \Bigg] \end{aligned}$$

Figure 8: The example stock list.

The shopping list requests one x and one y. The stock list has three containers available:

- The first has a $\frac{1}{2}$ chance of containing one x and costs 10.
- The second has a $\frac{1}{2}$ chance of containing one y and costs 10.
- The third has a $\frac{1}{3}$ chance of containing one x and a $\frac{1}{3}$ chance of containing one y, and costs 20.

3.3 The Root Node

Consequence trees are based on containers, so we will need to choose a container from the stock list to start with. To make referencing them easier, we will refer to the containers in the stock list as A, B, and C, respectively.

Let us start by building a consequence tree for container A. It has a $\frac{1}{2}$ chance of having an x, and nothing else. This means that the possible consequences are either we receive an x or we do not.

It is assumed that, if we were to choose this container first, we have decided that this container makes for the best first purchase. It would be unfortunate if we were to open the container and find ourselves empty-handed. However, in that event, the shopping list and stock list remain unaltered. Thus, this container will still be the next best choice.

Using this logic, we will assume we purchase this container repeatedly "until something happens," where our "something" is any nonempty consequence. If we purchase the container and get nothing, we do so again until we receive our x.

We will say that this node for container A therefore has a *loop chance* of $\frac{1}{2}$. As for the consequences, there is a single relevant one with a $\frac{1}{2}$ chance of giving us an x.

Once we have our x, our shopping list changes. In this case, we no long need any xs, and it is removed. Our new shopping list has only a single y.

Any container which does not have at least one y is ignored for further processing of this tree along the x branch. The remaining containers are B and C. This consequence's branch will connect to a node for B and a node for C, where each node is its own consequence tree.

This is depicted in the following figure:

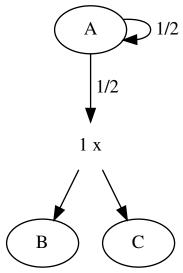

Figure 9: The beginning of a consequence tree for container A. It has a $\frac{1}{2}$ chance of looping, and a $\frac{1}{2}$ of having the consequence of giving 1 x. In the event of the 1 x consequence, we can choose to satisfy the y requirement from either the B container or the C container.

3.4 Branching Consequences

Once we branch, we reanalyze the potential containers in their new context. Since we no longer need an x after receiving one, we have no need to branch to the A container. In C's case, we can ignore its potential to contain an x.

We can fill in the tree with the rest of the information as follows:

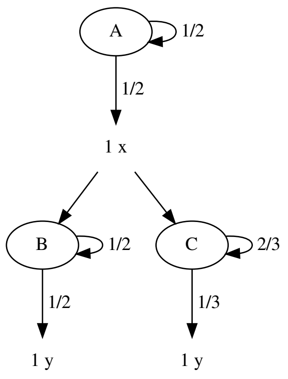

Figure 10: The complete consequence tree for A.

3.5 Average Cost

A complete tree tells us all possible ways that a shopping list can be satisfied after starting with a specific container. For this tree, we start with A. The only consequence available is receiving one x.

After receiving the x, our shopping list changes and we no longer need any more xs. Thus we focus our attention on constructing consequence trees for containers with relevant items. In this case, there are two relevant containers: B and C.

We again see what consequences each of them have. In both cases, the only relevant consequence is one y. After this, our shopping list will be fully satisfied and we will be done shopping.

With a completed tree, it is possible to calculate the average cost of each node. Of special note is A's consequence of containing one x. While the consequences are out of our control (as A may contain nothing), which container we pursue after attaining a consequence is a choice we have the power to make.

In this case, we have two choices: we can either pursue the tree down the B branch or the C branch. Thus, we can optimize the tree by discarding suboptimal branches.

3.6 Computing Costs

The average cost of any node can be calculated as follows, where l is the node's loop chance:[2]

$$\sum_{n=0}^{\infty}\left[(l)^n(b_1)(b_2)\dots(b_k)\right]$$

There will be one b term for each of the node's possible consequences. Each term will take the following form of $(r(p(n+1)+s))$, where r is that consequence's probability, p is the container's price, and s is the average cost of the most efficient container this branch leads to.[3]

Let us work through the A consequence tree as an example. The cost of A is 10 and a loop chance of $\frac{1}{2}$, so its summation will appear as follows:

$$\sum_{n=0}^{\infty}\left[\left(\frac{1}{2}\right)^n\left(\frac{1}{2}(10(n+1)+s_1)\right)\right]$$

Now, $s1$ is the average cost of the next nodes in the tree. First we calculate B's cost. It is terminal, so no s value is required:

$$\sum_{n=0}^{\infty}\left[\left(\frac{1}{2}\right)^n\left(\frac{1}{2}(10(n+1)+0)\right)\right]=20$$

Next is C's cost:

$$\sum_{n=0}^{\infty}\left[\left(\frac{2}{3}\right)^n\left(\frac{1}{3}(20(n+1)+0)\right)\right]=60$$

[2]There are simpler methods for calculating costs of nodes with only a single consequence, such as multiplying by the reciprocal of the consequence's probability. This should certainly be done when possible. However, this summation method works for all nodes, no matter how many consequences they may have.

[3]This is 0 if it is a terminal node.

3.7 Optimized Trees

It has become apparent that the cost of choosing the C branch is vastly more than the cost of choosing the B branch. Thus we can optimize our tree by discarding the possibility of choosing C, as the optimal solution will not include it:

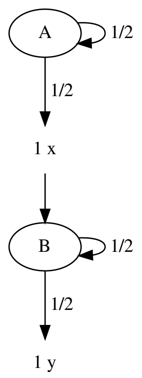

Figure 11: The optimized consequence tree for A.

We can now plug in the cost of B into s_1 when calculating the cost of A:

$$\sum_{n=0}^{\infty}\left[\left(\frac{1}{2}\right)^n\left(\frac{1}{2}\left(10\left(n+1\right)+20\right)\right)\right]=40$$

Thus, the average cost of starting with A is 40.

4 Iteration

4.1 Finalizing the ABC List

By using the above methods to calculate the average cost of starting with every container, we can find the container with the lowest average starting cost. Its optimized consequence tree will offer a plan in order to satisfy the shopping list in the most efficient way for any outcome. In this section, we will see this in action while continuing to use containers A, B, and C.

The average cost of B works out to be the same as A, as it is just A backwards.

However, C has a more interesting consequence tree:

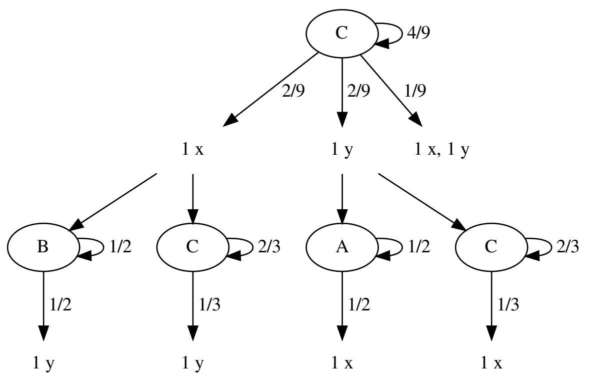

Figure 12: The complete consequence tree for C.

Upon optimization, we find:

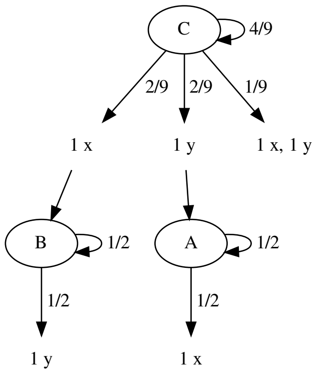

Figure 13: The optimized consequence tree for C.

From here, we can calculate its cost:

$$\sum_{n=0}^{\infty}\left[\left(\frac{4}{9}\right)^{n}\left(\frac{2}{9}\left(20\left(n+1\right)+20\right)+\frac{2}{9}\left(20\left(n+1\right)+20\right)+\frac{1}{9}\left(20\left(n+1\right)\right)\right)\right]=52$$

Thus in the case of the example shopping list problem, we can see that the average cost of starting with A and B are both 40, whereas the average cost of starting with C is 52. The optimal solution, then, would be to either start with A or B.

4.2 Conclusion

By analyzing the possible consequences of opening any container, it is possible to build a tree of all possible outcomes. From this tree, we can focus on the decision points, the branches after each consequence, and prune the more expensive ones. In doing so, we create an optimized consequence tree. We can then recursively calculate the average cost of the entire tree.

We can then repeat this process for each container, finding the average cost of all choices. If we choose the container with the lowest average cost and decide further containers based on its optimized consequence tree, we can be sure that we are satisfying our list for the lowest cost under the circumstances.