Sawtooth Software

RESEARCH PAPER SERIES

Stepwise TURF + Swaps

David Squire and Bryan Orme
Sawtooth Software, Inc.

Stepwise TURF + Swaps

David Squire and Bryan Orme, Sawtooth Software

I was looking for love in all the wrong places
Looking for love in too many faces

—Waylon Jennings, "Lookin' for Love"

Looking for Love in Truly Large Spaces

As with the refrain popularized by the Waylon Jennings song "Lookin' for Love," market researchers sometimes spend an awful long time looking in the wrong places when employing exhaustive optimization searches such as with TURF. TURF stands for "total unduplicated reach and frequency." It is an optimization approach for finding a subset of items (portfolio) that "reaches" the maximum number of respondents possible. Standard TURF searches look at every possible combination of portfolios, but this can become very time-consuming and even infeasible as the size of the problems increases. We test a heuristic shortcut method of stepwise TURF plus additional swaps[1] that requires a fraction of the effort involved in exhaustive search, yet finds just as much "love" as the exhaustive search for three real data sets. The Stepwise + Swaps approach described here (as well as the Exhaustive Search) are available within our MaxDiff Analyzer software tool.

What is TURF?

The classic TURF problem is one of choosing which flavors of ice cream to stock in the freezer at a grocery store. The grocer may decide that he/she has limited space and can only include up to seven flavors (out of 42 possible flavors). The grocer wants to maximize the chance that shoppers will find a flavor in the freezer that they are likely to choose. With the "Threshold" method, if the respondent encounters a flavor that meets a certain criterion (such as at least a four on a five-point purchase intent scale), the respondent is counted as "reached."[2] The problem isn't as simple as including the seven most

[1] Other non-exhaustive search methods such as Genetic Algorithms could be examined, though we haven't extended our research to assess the performance and speed of competing methods.

[2] The "threshold" reach method is not the only approach that may be used in TURF to evaluate the quality of a portfolio. Frequency may also be computed as the number of items in the portfolio that reach each respondent. Search results may be prioritized by reach and frequency. If two portfolios have identical reach, but one has significantly higher total frequency, then the one with higher frequency would be preferred. If raw MaxDiff scores (as estimated via HB) are used, another approach called "Weighted by Probability" is offered in the MaxDiff Analyzer TURF routine wherein the sum of the antilogs of the scores for the items in the portfolio is used to evaluate the quality of the portfolio.

preferred flavors on average across the sample. Niche flavors that appeal to segments of the population (and that can increase total reach) would be overlooked.

Let's consider a smaller example to illustrate the niche flavors problem. Assume we need to find two flavors out of three possible flavors to stock in the freezer at the grocery store. The three flavors are chocolate, strawberry and banana. Ten respondents participate in this three-flavor MaxDiff[3] study. Eight of the respondents prefer chocolate and strawberry but hate banana; whereas the other two respondents love banana and don't like the other two flavors. The two most preferred flavors on average across this sample are chocolate and strawberry. But if we choose these two flavors then we will only "reach" the eight respondents that love chocolate and strawberry and we will miss the two respondents that love banana. Therefore, if we choose banana and chocolate or banana and strawberry then we will be able to reach all ten respondents. Furthermore, the difference in reach between the banana/chocolate portfolio and the banana/strawberry portfolio will probably not be very large. Knowing that either of these two portfolios will reach a similar number of respondents allows us to use additional information about the flavors to help us choose which portfolio we stock in the end. For example, if chocolate has a higher profit margin than strawberry, then we may decide to choose chocolate and banana as our two flavors.

Exhaustive Search TURF

Now that we have described TURF analysis, let's consider the methods we can use to find optimal portfolios. First, we'll consider an exhaustive search, the most popular approach that is guaranteed to find the globally optimal solution. For the above three-flavor example, we would first enumerate all of the possible portfolios (the order of the flavors within the portfolio doesn't matter):

1. Chocolate and Strawberry
2. Chocolate and Banana
3. Strawberry and Banana

This is a very trivial problem as it only has three possible portfolios. It is very easy to evaluate the reach of each of these portfolios for each of the ten respondents and then sort the results in order of reach. The exhaustive search guarantees that we identify which portfolio is the best as well as know how all of the other portfolios compare.

The downside to the exhaustive method is that in the real world the number of possible portfolios that need to be evaluated can grow to become extremely large. Let's consider the mathematics behind the portfolio combinations.

In mathematics, each portfolio is called a k-combination or a subset of k distinct elements from the total set of elements (where n is the number of items in the total set of items). (In TURF and mathematical combinations the order of the elements in the portfolio or the k-combination does not

[3] Even though we describe the use of TURF with MaxDiff (best-worst) data, the software tool described here (MaxDiff Analyzer) can use data collected in any way, such as Likert scales, etc.

matter.) For a set of n items, the number of k-combinations may be written using factorials as $\frac{n!}{k!(n-k)!}$ where $k \leq n$.

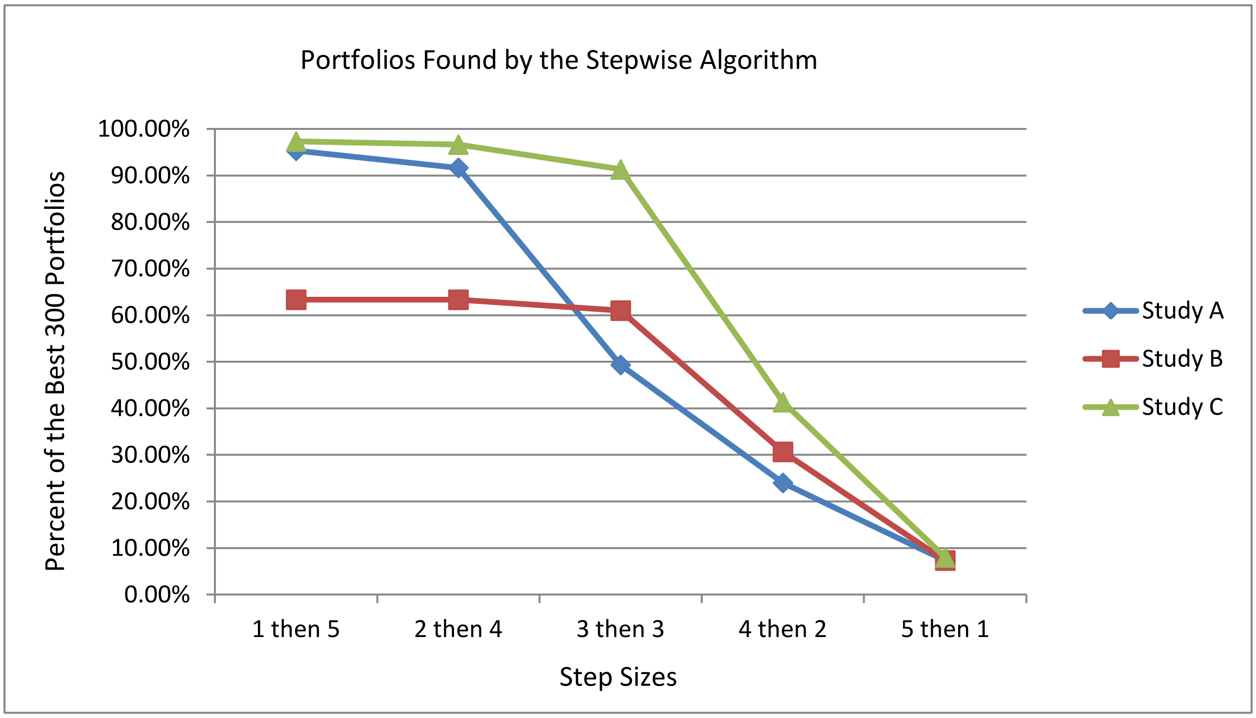
Due to the nature of the equation described above, the search space grows very quickly as the total number of items grows and as the number of items in the portfolio approaches half the total number of items. The size of the search space can easily grow to a point where it is no longer practical to try an exhaustive search due to time limitations. For example, to search for 21 flavors of ice cream out of 42 possible flavors requires examining 538,257,874,440 portfolios for each respondent.

Stepwise TURF

We can cut down the number of portfolios examined by using a stepwise approach. With this approach, first a smaller portfolio is found using the exhaustive search algorithm. We then force the top portfolio from this first search into subsequent TURF runs. This method reduces the size of the search space dramatically. For example, in a study with 63 items, to find a portfolio of seven using the exhaustive algorithm would require considering 553,270,671 portfolios for each respondent. Using the stepwise approach we could first find the best portfolio of size three using the exhaustive algorithm. This would only require considering 39,711 portfolios. If the top portfolio of size three for our example includes items 2, 38, and 62, for the second step we run another exhaustive TURF with a portfolio size of seven while forcing items 2, 38, and 62 into each possible portfolio. This second step would require searching through 487,635 possible portfolios. The two-step approach will thus require examining a total of 527,346 portfolios, which is only 0.095% of the portfolios that would be searched by the exhaustive algorithm.

One potential problem is that the stepwise method may miss some of the best portfolios and is not even guaranteed to find the one optimal portfolio. In our sample 63 item study, the stepwise approach found 211 of the top 300 portfolios found by the exhaustive method (choose 7 out of 63). So we were able to find 70.333% of the top 300 portfolios by only searching through 0.095% of the total possible portfolios. In all of our testing, the stepwise method nearly always found the top (optimal) portfolio.

For the remainder of the paper we will discuss results we obtained using three real MaxDiff data sets involving 42, 63 (previously introduced), and 118 items respectively. To make the search space quite big (but still solvable by exhaustive) we searched for optimal portfolios of six items for all three datasets. Regarding the shortcut stepwise approach, our initial intuition was that it would be best to take as big a first step as computationally feasible (without taking such a big step that it took many hours or even days to compute). Interestingly, we found from our analysis of these three real MaxDiff datasets (see graph below) that it is better to take the biggest step *last*. For this analysis, we chose to use two steps (of differing step size combinations) to search for portfolios of size six. The five two-step combinations are 1 item then 5, 2 then 4, 3 then 3, 4 then 2, and 5 then 1. As a benchmark for success, we first ran the exhaustive TURF search for these three datasets so we'd be certain of the globally optimal top 300 portfolios of six items (the largest of these runs took 10 days to compute!). The following chart shows how each of these two-step options performed compared to the exhaustive search.



Study B leads to a peculiar outcome. Even though it has 63 items, which is more than Study A's 42 items and less than Study C's 118 items, its results are not between the results of the other two studies. This is due to the data set itself. The number of items is not what determines how well the stepwise approach will perform. Rather, it is the heterogeneity of the data and the presence of niche items that will more strongly determine the difficulty of the problem facing a heuristic algorithm such as stepwise.

Can We Find Even More?

After reviewing our results, we decided to see how many more of the top 300 portfolios (as determined through exhaustive search) we would be able to find by performing swaps on the top portfolios found with Stepwise.

For the swap algorithm, we hold all items in a portfolio constant except one of the items. The item not held constant is then swapped for each of the items that is not already included in the portfolio. Each of the items in the portfolio is swapped in this manner.

For example, consider a project with six items where we are trying to find portfolios of three items. Assume that the top portfolio found by the stepwise algorithm is 2, 3, and 5. The items that we will be swapping in are 1, 4, and 6, which are the items not already in the portfolio. For the swaps, we first hold the second and third items constant and create new portfolios by swapping out the first item like this:

- 1, 3, and 5
- 4, 3, and 5
- 6, 3, and 5

Then, we hold the first and third items constant and create new portfolios by swapping out the second item like this:

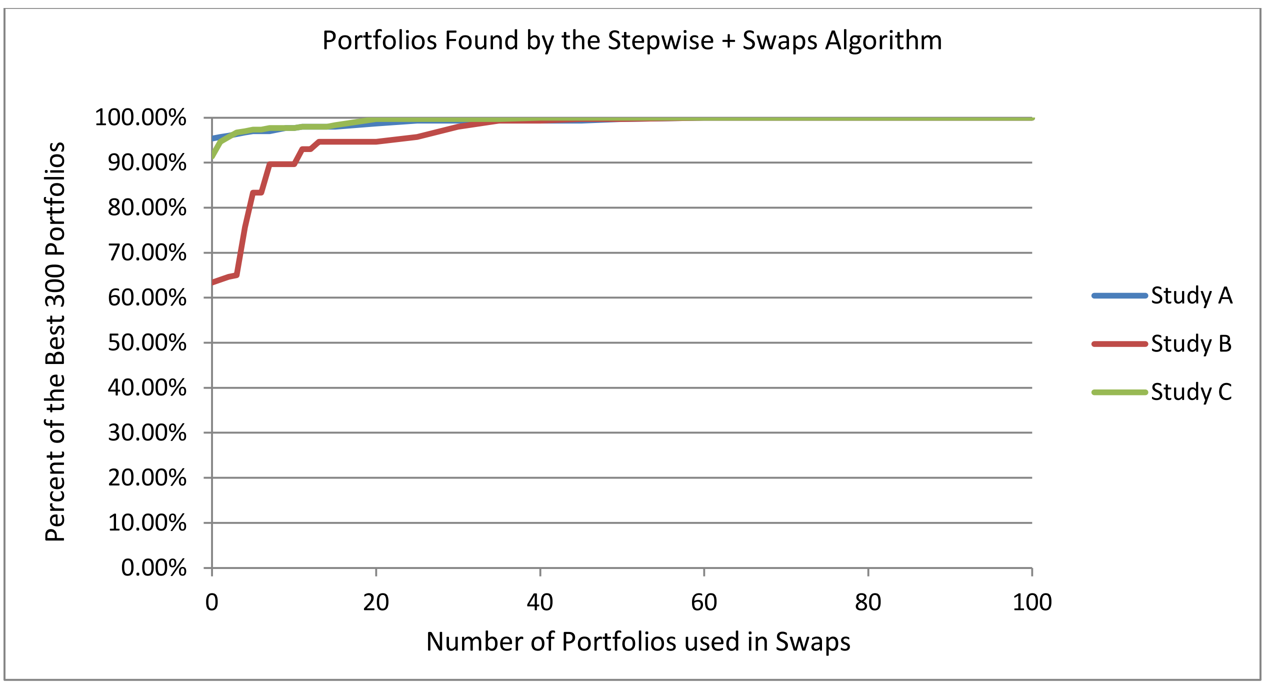
- 2, 1, and 5
- 2, 4, and 5
- 2, 6, and 5

And finally we swap the third item:

- 2, 3, and 1
- 2, 3, and 4
- 2, 3, and 6

Each of these new portfolios is then evaluated to see if it has a better reach than those already in the list of top portfolios found by the stepwise algorithm. If swapping an item leads to the discovery of a new top-performing portfolio, that new portfolio will be submitted again to the swaps routine until no new top-performing portfolios are discovered.

How many of the top portfolios found by the stepwise algorithm should we include in the swapping algorithm? We want to find as many of the best portfolios as we can without adding too much processing time. To evaluate the value (and cost) of including more or fewer top portfolios in the swaps, we return to our three studies (with 42, 63, and 118 items). Below is a chart that shows how many more of the top portfolios are found as we increase the number of portfolios involved in the swaps[4]:



[4] For this chart, the step sizes used in the Stepwise part of the algorithm were determined by the MaxDiff Analyzer software. They do not match the step sizes in the previous chart, so the starting points (at 0 swaps) are slightly different on the Y axis.

As the results show, the swaps work beautifully! With additional swaps, we find all of the top 300 globally optimal solutions for these three data sets. The algorithm achieved this by investigating swaps involving the top 60 portfolios found by the stepwise approach. It turns out that the processing time involved in swapping items for even the top 100 portfolios was only a couple seconds at most. Therefore, we have decided to include all of the top portfolios (100 in the MaxDiff Analyzer software) in the swapping algorithm for good measure.

So, returning to our problem of finding a portfolio of size seven with our 63 item study, we can now invoke the swaps. The additional swaps add 154,448 more portfolios to consider (beyond the portfolios examined by the stepwise algorithm). Adding this to the portfolios searched by the stepwise algorithm (527,346) now brings us to a total of 681,794 portfolios searched (which takes 25 seconds to run). This is still only 0.123% of the portfolios that would be searched by the exhaustive algorithm. The upside is we now find all 300 of the globally top portfolios found using the exhaustive algorithm (instead of 211 out of 300). The extra swaps are well worth the small amount of additional effort!

Conclusion

Researchers can spend a lot of time in exhaustive search “looking for love in all the wrong places.” Although we cannot conclude that Stepwise TURF + Swaps will *always* find your one and true love (the globally optimal portfolio) for every data set, we are very impressed with its ability to find the optimum plus the other top 299 portfolios for three challenging, real-life data sets. The Stepwise TURF + Swaps capability is currently available in our MaxDiff Analyzer software. Rather than “spending a lifetime” exhaustively “looking for love in too many faces,” your TURF problems will run typically in about 30 seconds to 10 minutes for even the largest problems that would take exhaustive search years or even a lifetime to compute.