## Profiting from investors' herd behaviour following analyst recommendations

*Abstract: Investors are often most at ease buying a stock with a high percentage of analyst BUY recommendations. This is herd behaviour bias in action; that is, the tendency to feel more comfortable belonging to the consensus or the herd. In reality, companies with fewer BUY recommendations have historically outperformed those with more. This creates a problem for investors who rely on analysts' calls and who stay with the herd.*

*In this paper, we share the story of how we researched a strategy designed to profit from this herd behaviour bias. In addition, we demonstrate how we improved the efficacy of the strategy by applying machine learning.*

Craig Basinger, Chris Kerlow, Shane Obata and Derek Benedet – May 2018

## Introduction

Many investors, advisors and portfolio managers rely on sell-side analyst ratings. This complete or partial outsourcing of due diligence is attributable to a number of behavioural biases, including *overconfidence*, *confirmation* and *herd behaviour*.

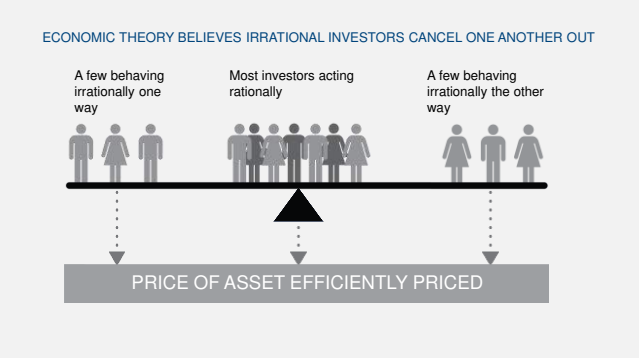
**Overconfidence**: An expert (the analyst) knows better than the market does where companies should trade

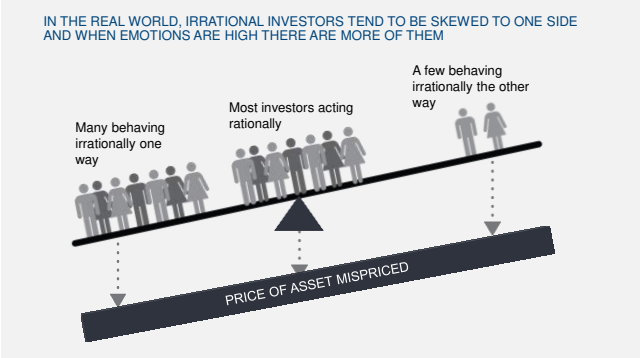
**Confirmation**: When investors often seek out analyst recommendations that are aligned with their opinions

**Herd behaviour**: Most investors feel most comfortable owning companies that many analysts also like

We believe these flaws are present in many investment processes and can lead to mispriced assets in the marketplace. For active managers such as ourselves, a mispriced asset provides an opportunity to profit on behalf of our investors.

In theory, most investors behave rationally and most assets are priced fairly. However, in practice, the markets are more complex. Sometimes, behavioural biases – driven by emotions – cause investors to act irrationally. This can lead to mispriced assets. These biases are human in nature. As a result, they should continue to elicit bad behaviours that result in predictable mispricings. Investors who identify these patterns are set to potentially reap the benefits.




In this report, we share our analysis on how to profit from the herd behaviour elicited by analyst recommendations. We also show how investing in unloved companies may lead to outperformance versus the benchmark and relative to companies that are more loved by analysts. Lastly, we demonstrate how applying machine learning can help to improve upon a base strategy. In sum, we believe that investors can benefit from using strategies that are built to capitalize on mistakes caused by other investors' behavioural biases.

## The problem: the herd is often wrong

The herd mentality is a behavioural tendency that is hard-coded in our DNA. We tend to feel more comfortable when we are in agreement with the consensus. On the subject of analyst ratings, investors generally prefer to buy, own or add to companies that have a higher percentage of BUY ratings. Conversely, investors tend to neglect or overlook unloved companies with a very low percentage of BUY ratings.

Two factors likely drive herd behaviour. First, when more analysts have positive ratings for a stock, there is a preponderance of reports with positive views. This makes it easier for investors to come across reports that recommend a company. The second factor is the desire to belong to the herd. If you invest in a company with mostly BUY recommendations, then you are part of the consensus. This has the benefit of protecting you because you can say that "everyone said to buy" if the recommendation does not work out. Alternatively, if you invest in an unloved company with few BUY ratings, then you own that decision because if it does not go well then there is no one else to blame.

*Behavioural finance*

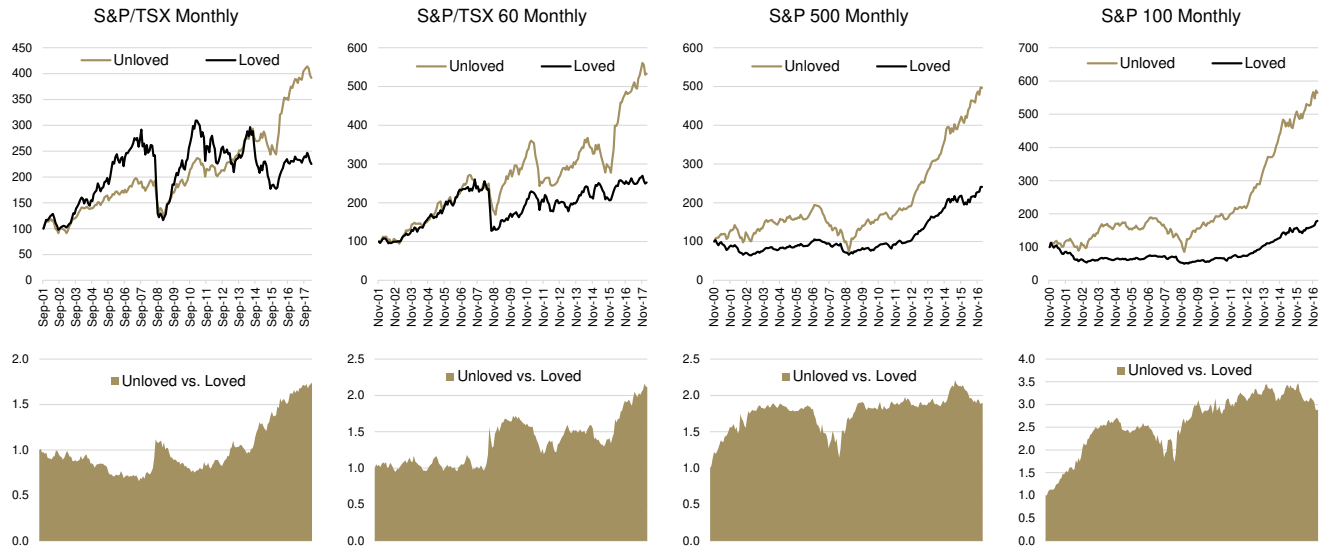
Behavioural finance, which combines finance and psychology, is the study of how we, as investors, make decisions, and more poignantly, how we often make poor investment decisions. Psychology has uncovered many heuristics that our brains use to help us make decisions quickly. These are rules of thumb or mental shortcuts that enable us to navigate the enormous number of decisions we make every day. Unfortunately, many of these heuristics can lead to predictable decision-making errors.

In a low-stakes environment, our heuristics often help us because emotions are low. In a high-stakes investing environment; however, the opposite is true. Investing is emotionally charged because our successes or failures have real consequences. That is especially true during periods of heightened volatility.

Research has shown that behavioural biases are often systematic. There are certain market events – such as large price moves, overreactions to earnings and heavy news flow – that trigger the same biases in many investors. These biases cause them to act in irrational ways, resulting in mispriced assets. If certain identifiable circumstances cause investors to act in predicable and irrational ways, then perhaps we can profit from their misbehaviour. That is exactly what we set out to do with the *unloved to less unloved* strategy.

***History has shown that companies with fewer BUY recommendations typically outperform the companies with many BUY recommendations***

In both Canada and the U.S., we found that companies with a lower percentage of BUY recommendations outperformed those with a higher percentage. In our analysis, we ranked index constituents based on their percentage of BUY recommendations over total recommendations. We then tracked the subsequent performance of the top and bottom quintile until the next rebalancing. The analysis covered the past 20 years and our findings were consistent across the S&P/TSX Composite Index, S&P/TSX 60 Index, S&P 500 Index and S&P 100 Index. We used multiple indices so as to avoid a potential company size influence. The analysis was rebalanced monthly and quarterly, with similar results.




*Sources: Bloomberg, Richardson GMP. As at March 30th, 2018.*

The outperformance of unloved over loved companies was not consistent during all periods of the analysis. That said, based on our analysis, on average, investors may want to avoid loved companies and to seek unloved ones.

There is some logic behind these findings. The market tends to move on new information. In this case, the new information was analysts changing their recommendations. The market likely knows and discounts 10 pre-existing BUY ratings. If a company has all BUY recommendations, then a downgrade is the next logical ratings change. Conversely, for a company with very few BUYs, many more analysts could upgrade their ratings should they reconsider the prospects of or see more value in the company.

If the herd, based on the percentage of BUY recommendations, is so often wrong, how can we profit from this?

## The solution: the *unloved to less unloved* investment strategy

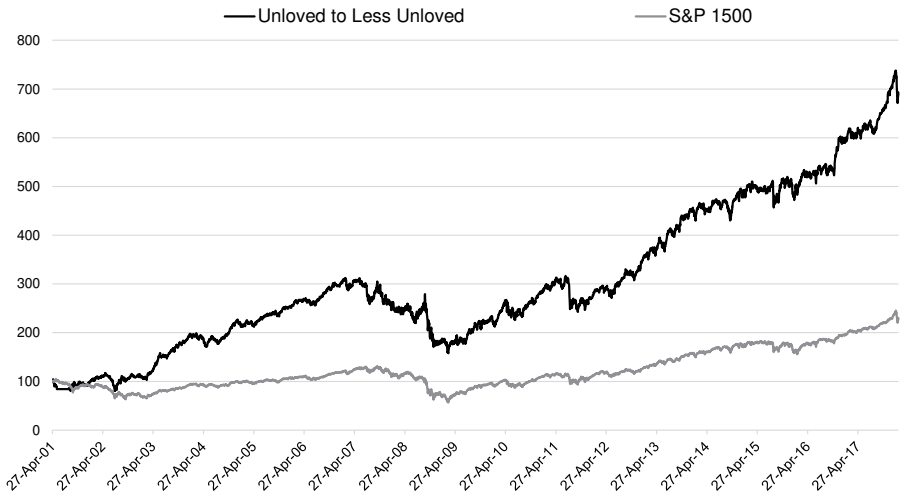
The simplest solution based on the above findings would have been to develop a strategy that went long companies with low percentages of BUY recommendations (unloved) and went short those with high percentages (loved). However, that would have proved difficult given that loved companies may outperform unloved ones for extended periods. Instead, we chose to focus more on unloved companies that began to receive upgrades.

The *unloved to less unloved* strategy attempts to harvest gains from hated or neglected companies that have remained unloved for an extended period. A company that starts to receive upgrades that move its BUY / TOTAL recommendations ratio above a predefined threshold would trigger a potential investment opportunity. If those early upgrades are from forward-looking analysts, then more upgrades could follow, creating the potential for a recovery in the share price.

## Strategy development

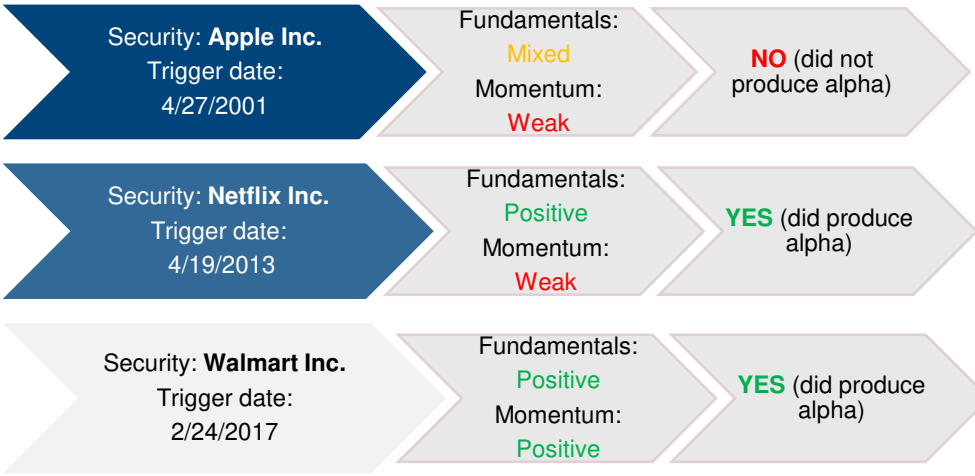
The base strategy was good before we started refining it; our goal was to make it better. The first step was to create a screening tool. Excel was too slow, so we partnered with Bloomberg's quant team to develop a system that performed most of the work in the cloud. The result was a dynamic Python-based tool that allowed us to screen various indices for trading instances. We used this tool to compile a database that goes back 20 years.


The second step was to automate our trading rules, which involved setting ideal stop-loss and reset (effectively trailing stop) levels. Optimization greatly improved the results of the unloved strategy, which outperformed the S&P 1500 benchmark over the entire data range.




*Back-tested, not actual results - April 27, 2001 to February 14, 2018.*

The third step was to apply machine learning. Before we continue, let us briefly review the concept of supervised learning. Supervised learning typically involves either regression or classification. In our case, the task was binary classification. We wanted to ask a question (X) to help us to arrive at an answer (Y). The question was "which variables will help to predict whether or not a trade will produce alpha?" The answer was YES or NO, depending on whether or not the trade outperformed the index.

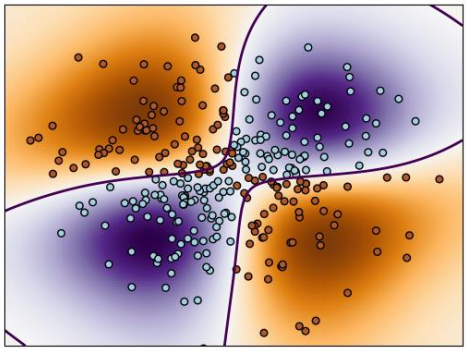



*This figure is a stylized example of some of the trading instances generated by our quant screen. Specific features have been omitted.*

We used our database to pull historical data (fundamental, momentum, price, etc.) for each trading instance going back three months. Our belief was that this data would help us refine and improve our trade selection process.

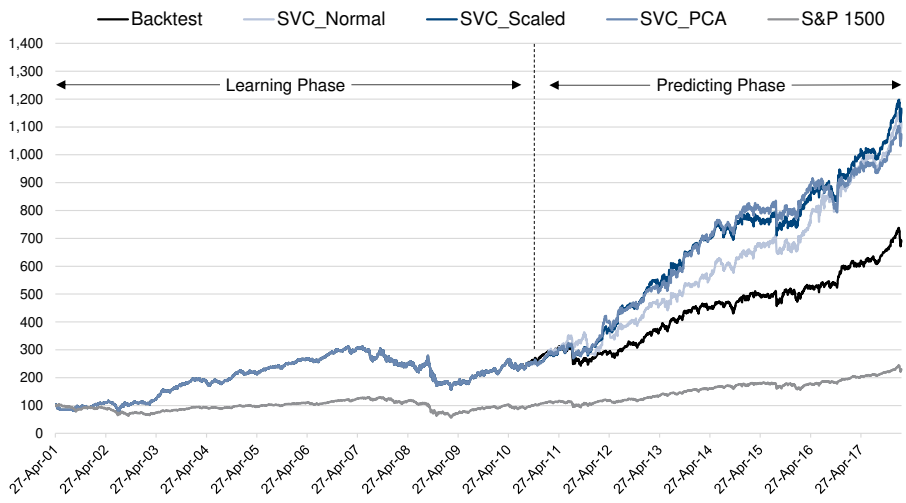
In supervised learning, we provide the machine with both the question (X) and the answer (Y). It then learns from the data by identifying patterns, using whatever algorithm we choose to apply. We tried using various models but settled on *non-linear* Support Vector Machines (SVMs), since they consistently performed well.

SVMs are a set of supervised learning models that tend to work well in high-dimensional spaces. These machines are used to separate data into classes using "decision boundaries." The next figure shows how a non-linear SVM classifies blue versus brown points with a high degree of accuracy based on such boundaries.



*Source: "Non-linear SVM" – scikit-learn*

For our purposes, the aim was to separate the trading instances into YES or NO classes. The results were very promising. Three of our four SVM models outperformed the base strategy…



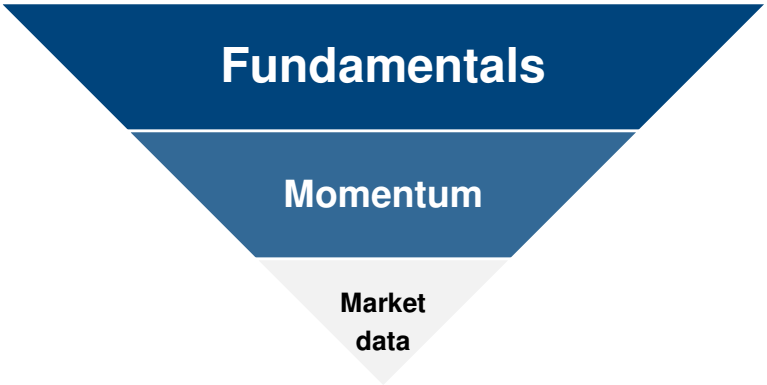

*Backtested, not actual results - April 27, 2001 to February 14, 2018.*

…while taking on ~90% of the trades:

| Model | Optimized PnL | Stop | Reset | Trade Count | % | Prior |
|---|---|---|---|---|---|---|
| Backtest | 592.59% | -18.00% | 1.00% | 640 | 100.00% | 253 |
| SVC_Normal | 1003.73% | | | 575 | 89.84% | 322 |
| SVC_Scaled | 1042.37% | | | 594 | 92.81% | 341 |
| SVC_PCA | 960.85% | | | 585 | 91.41% | 332 |

*Backtested – not actual results, April 27, 2001 to February 14, 2018.*

Machine learning is often framed as a black box. Inputs go in, magic happens, and outputs come out. That is not how it actually works. The models have improved over time and are now more transparent than ever. We determined which features (variables + values) were the most important by running a series of simple functions. Some of the features that came up repeatedly were fundamental revisions, trend scores (momentum) and lagged returns. We are optimistic that data science will continue to add value in our investment process. In our view, the more data the better.




*The above is a stylized example of feature importances. Specifics have been omitted.*

## Risk management

Trade selection is important, but risk management is what matters most. We should evaluate each potential trade on actual risk and expected profitability. These inputs can guide our sizing and help us to avoid falling prey to our own biases. We should also consider how each trade will fit into the existing portfolio so as to limit concentration in select strategies and or sectors.

## Conclusion

We believe that investors can benefit from using strategies that are built to capitalize on mistakes caused by behavioural biases. Our research confirmed our suspicions by showing that investing in unloved companies may lead to outperformance versus the benchmark. It also demonstrated how applying machine learning can help to improve upon a base strategy.

## Appendix

This section was written for those who are interested in what went on behind the curtains

*The Base Models*

| |
|---|
| # In[0]<br><br>Import data with features = X and targets = y |
| # In[1]<br><br>Split the data into training (80%) and test (20%) sets in order to prevent contamination of the test set (i.e. learning and predicting on the test set) |
| # In[2]<br><br>Build pipeline and parameter grid to allow for multi-step operations (scaling, PCA, etc.) and optimal model selection, respectively |
| # In[3]<br><br>Perform a 5x stratified cross-validation using the pipeline and parameter grid |
| # In[4]<br><br>Perform a 2nd 5x stratified cross-validation while using a different scoring system (“roc_auc”) |
| # In[5]<br><br>Select the best model based on test scores so as to avoid overfitting |
| # In[6]<br><br>Predict on the test set using the best model from above |
| # In[7]<br><br>Evaluate the model by examining the classification report, the precision-recall curve and the ROC curve |
| # In[8]<br><br>Tweak the model’s decision function in order to maximize true positives and to minimize false negatives |
| # In[9]<br><br>Make new predictions based on the tweaked model |
| # In[10]<br><br>Re-evaluate the tweaked model |

## The Live Models

Moving from production (development) to deployment (going "live") took a lot of effort. The hardest part was conceptualizing how to build a proper historical backtest. We had to go from the past to the present while avoiding look-ahead bias along the way. To do this, we built massive loops around our existing models. This allowed us to grow our dataset from 50% (2010) to 100% (the present) while maintaining an 80%/20% train/test split. Each iteration involved growing the dataset by 1% at a time and making a small number of predictions. When we finally arrived at the present we reverted back to our base models.

Now that we are live, the process is straightforward…

i) Quant screen generates a new trading instance
ii) Pull historical data for that company
iii) Re-run machine learning models with updated data
iv) Record new prediction
v) Trade the model

We understand that backtesting is not entirely realistic. That said, we are encouraged by the results thus far.

For an in depth look into SVMs, please click here.