Special Days and Where to Find Them

1 Formulating the Problem: A Calibration Tool For Special Days

Algorithmic trading is the automated execution of trading decisions in electronic markets, utilizing strategic order execution methods to minimize market impact and transaction costs. In futures exchanges, people can trade futures contracts to buy specific quantitites of a commodity or financial instrument at a specified price with delivery set for a specific future time. Execution strategies are developed based on predictions and sophisticated forecasting models of the behavior of certain futures on each day. Not every day in the exchange is the same. While on "normal" days, the behavior of a product in terms of intraday trade volume might be similar, there are certain days that have abnormal impact on certain products. For example, the End of Storage Index (EIA) is the measure of cleared market for total working gas in underground storage, and it comes out every Thursday. The EIA will have impact on the trade volume trend for crude oil. Note that not all products will be affected by this index. Special days like these will have impact on some products, but not for others.

This problem is two-pronged. The first problem is to be able to identify which special days have an effect on which products and which asset classes. We want to create a mapping from product or asset class (input) to a list of special days (output) that correspond to special days that have an impact on that product. The second part of the problem is that for every product and asset class, we want to fit an improved calibrated model for intraday volume forecasting for normal days as well as a separate model for each of the special days. We can cluster by similar special days, so we don't have tons of models.

2 Goals and Objectives

We want to create a calibration tool that will allow us to

1. separate special days from regular days in our forecasting and trading models
2. cluster special days based on their impacts on which products or cluster products based on their impact by special days
3. make separate models for special days and the products they affect and make calibrated models for all products on regular days.

3 Brainstorming Ideas

3.1 Distributions to Look At

- daily (over 1 day, by hour, by number of bins)
- over 1 week
- over 1 hour
- data grouped by holiday
- data grouped by special day

3.2 General Ideas:

- binning the data over the course of the day into hour bins
- binning days into groups as well

3.3 Variables:

Trade volume over one day (intraday trade volume) will be the first variable we look at in terms of characterizing market behavior. However, it might be valuable to also observe other dependent variables of the market, such as volatility and fraction of share outstanding traded, or price.

- trade volume
- trade price
- liquidity
- volatility
- margin size
- fraction of shares outstanding traded

3.4 Characteristic Statistics that Might be Used for Features (specific to trade volume for now)

- average daily trade volume for product p over all days together
- average daily trade volume for product p over each day Monday-Friday (to see if different days have different distributions)
- histogram of trade volume over the course of each day over all days together, separated by month, then separated by year (Has the trade volume trends changed? How often should models be recalibrated?)
- average hourly trade volume for product p over all days together
- average hourly trade volume for product p over each day Monday-Friday (to see if different days have different distributions)
- average change in trade volume per hour for product p over all days together

- average change in trade volume per hour for product p over each day Monday-Friday (to see if different days have different distributions)

3.5 Algorithm Ideas if the Objective is Forecasting:

- Moving average, weighted moving average
- Naive (using previous day's value, baseline)
- Linear Regression
- Random forests/regression decision trees
- Kalman + EM + Regularized Intraday Forecasting
- Autoregressive Moving Average (ARMA)
- Autoregressive Integrated Moving Average (ARIMA)
- LSTM Models
- Kalman Filtering
- Exponential Smoothing
- Hidden Markov Models
- Support Vector Regression
- Using Sliding Windows

3.6 Ideas for Clustering of Special Days:

- kmeans clustering?
- correlation clustering
- kernel PCA
- It is imperative to come up with a good similarity measure.

4 Questions We're Looking to Answer:

- How do the special days' market behavior compare to normal days?
- How can we be able to identify unknown special days? The un-encountered special days that are not on the given list?

Possible Approaches to Objective 1: Testing the Relationship Between Special Days and Products

Given two time series (assume one for all days G (you don't necessarily know which are normal and which are not) and one for special days H)

$$G = \{g_1, g_2, ..., g_n\}$$

$$H = \{h_1, h_2, ..., h_n\}$$

Alternatively, perhaps we should have two time series (assume one for normal days G (separate based on the list since we have it which are normal and which are not) and one for special days H)

Approach 1: Deviation from Basic Statistical Measures

This is the most basic approach, where we take the basic statistical measures of each daily distribution: IQR, mean, medium, standard deviation, range.

$$D(G, H) = \sum_{measures} |G(i) - H(i)|^2$$

Approach 2: Sum of Squared Differences

This is the second most simple difference identifier between distributions. It's a bit naive, but taking the mean absolute value helps correct for some of the naivety.
We divide the day into n bins to simplify the computation. For each series, we compute

$$D(G, H) = \sum_{i}^{n} |avg(G_i) - avg(H_i)|^2$$

If either curve crosses over the other, we split on the intersection, so as to not zero out the difference. Alternatively, the integral could by used instead of a difference in average after fitting a short linear curve to the bin.

Then, we could also compute variance of squared differences between values, to check for an off-shifted distribution curve.

Approach 3: Kolmogorov-Smirnov Goodness of Fit Test

The K-S Goodness of Fit Test is a statistical test used to decide if two samples come from the same distribution. Suppose the first sample has size m with an observed empirical cumulative distribution function $F(x)$ and that the second sample has size n with observed eCDF $G(x)$. Define

$$D_{m,n} = \min_x |F(x) - G(x)| \tag{1}$$

$D_{m,n}$ is the difference between the two distributions, so if we can show that $D_{m,n}$ is sufficiently small, we can show that the distribution of trade volume over two different days is similar, and vice versa for differently distributed days.

The null hypothesis is H_0: both samples come from a population with the same distribution. For the K-S test for normality, we reject the null hypothesis (at significance level α) if $D_{m,n} > D_{m,n,\alpha}$ where $D_{m,n,\alpha}$ is the critical value.

$c(\alpha)$ = the inverse of the Kolmorogov distribution at α.The values of $c(\alpha)$ are also the numerators of the last entries in the Kolmogorov-Smirnov Table. The Kolmogorov Distribution has value

$$F(x) = \frac{\sqrt{2\pi}}{x} \sum_{k=1}^{\inf} e^{\frac{-(2k-1)^2\pi^2}{8x^2}}$$

.

For every product or asset class, we will check each special day against the aggregate average daily distribution of trade volume that that product. We will run a K-S test with a high alpha to ensure confidence that the distributions differ.

Approach 4: Fit a Predictor and Check Deviation

This algorithm does a time series prediction on the hour of the day, or the bins of the day if we want to bin the day on some size k. We do some feature engineering, or just time series prediction. We use different regression models to predict the next bin's or hour's trade volume for the product. We measure the squared difference between predicted volume and actual volume.

$$D(G, H) = \sum_i^n |predictedVol(G_i) - predictedVol(H_i)|^2$$

Right off the bat, some simple models we could use are linear regression, moving average, and random forests, as well as more complex models like LSTMs. We use the best cross-validated performing regressor, measured too with a ROC curve analysis. If the difference is greater than some threshold, we can flag this special day as impactful on the product.

Alternatively, we can use percentage of unexpected/anomalous values as a measure of distance.

$$D(G,H)_{alternative} = \frac{1}{n}\sum_{i}^{n} I(G_i, H_i)$$

$$I(G_i, H_i) = \begin{cases} 1 & \text{if } |predictedVol(G_i) - predictedVol(H_i)| \geq threshold \\ 0 & otherwise \end{cases}$$

Approach 5: Dynamic Time Warping

Dynamic time warping is an algorithm used to measure similarity between two sequences which may vary in time or speed. A non-linear alignment produces a more intuitive similarity measure, allowing similar shapes to match even if they are out of phase in the time axis. It allows for stretched and compressed sections of the sequence. This is a dynamic programming solution.

Algorithm 1: Dynamic Time Warping

Result: DTW Measure of Similarity between two series

Two time series of trade volume over the course of a day;

1. Divide the two series into K equal data points;
2. Calculate the euclidean distance between the first point in the first series and every point in the second series. Store the minimum distance calculated. (this is the ?time warp? stage);
3. Move to the second point and repeat 2. Move step by step along points and repeat 2 till all points are exhausted;
4. Repeat 2 and 3 but with the second series as a reference point;
5. Add up all the minimum distances that were stored and this is a true measure of similarity between the two series;

There are optimizations that can be performed to prune the search space of the dynamic time warping function, including restrictions on monotonicity, continuity, boundary conditions, warping window, and slope constraint.

In finding the minimum distances along the grid, you create an optimal path from the bottom left of the grid to the top right. This path is called the warping path and this path follows a function called the warping function. When the warping function is applied to both time series it transformed them to two new time series that are aligned in time.

If the time-adjusted distance between daily time series differs significantly, we flag the day as a special day with impact on this product.

Approach 7: Kernel Density Estimation to Fit a Distribution and Measure Deviation from Fitted Distribution

Kernel Density Estimate approximates the probability distribution function of a dataset. KDE is a technique that let's you create a smooth distribution curve given a set of data. It essentially generates points that look like they came from a certain dataset, and this behavior can well simulate the real data.

The KDE algorithm takes a parameter, bandwidth, that affects how smooth the resulting curve is. The KDE is calculating by weighting the distances of all the data points we've seen for each location. Changing the bandwidth changes the shape of the kernel: a lower bandwidth means only points very close to the current position are given any weight, which leads to the estimate looking squiggly. A higher bandwidth means a shallow kernel where distant points can contribute.

The weighted probability distribution function is as follows:

$$\hat{f}(x) = \sum_{obs} K(\frac{x - obs}{bandwidth}) = \frac{1}{n}\sum_{i=1}^{n} K(\frac{x - x(i)}{h})$$

The kernel function for the normal distribution is

$$\frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}x^2)$$

which I will use because it's continuous and non piecewise. The rule of thumb is to use

$$\hat{h_0} = 2.78\hat{\sigma}n^{-1/5}$$

We then use 1-sample K-S testing to see if the pdf that we fit to the normal days is the same as the sample series from a special day. If it differs sufficiently, we flag this special day as having impact on this product.

Recall from Algorithm 1, Define

$$D_{m,n} = \min_x |\hat{f}(x) - G(x)| \tag{2}$$

$D_{m,n}$ is the difference between the two distributions, so if we can show that $D_{m,n}$ is sufficiently small, we can show that the distribution of trade volume over two different days is similar, and vice versa for differently distributed days.

The null hypothesis is H_0: both samples come from a population with the same distribution. For the K-S test for normality, we reject the null hypothesis (at significance level α) if $D_{m,n} > D_{m,n,\alpha}$ where $D_{m,n,\alpha}$ is the critical value.

$c(\alpha)$ = the inverse of the Kolmorogov distribution at α.

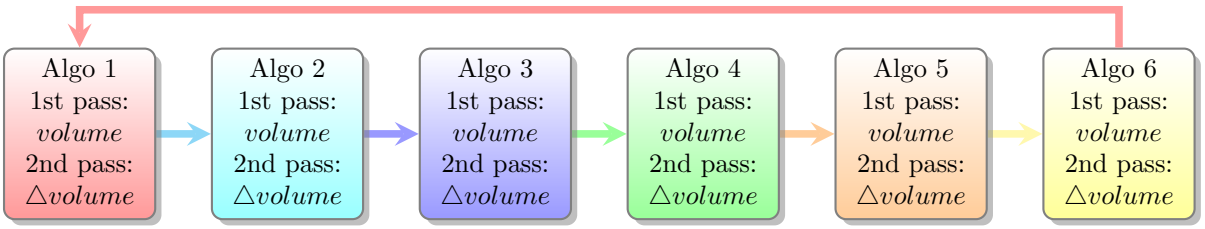
Approach 7: Extension by Using Change in Trade Volume Instead of Trade Volume

We can do the same analysis of the first 6 approaches using both trade volume and change in trade volume.

Composite Approach 1

One potential approach that combines the 7 approaches is doing a chronological systematic test. The test would basically tell you:

Does special day d have a significant impact on product p?



You would systematically proceed through the tests, and the tests proceed from simplest check advancing to more sophisticated checks. This would be computationally expensive, but if it was being run offline it might be possible, depending on the scale of the data.

If 6/12 or more checks are sufficiently satisfied, the special day is flagged, and we return Yes to the question posed. In other words, if over 50 percent of the special day's occurrences flag an impact, then we can conclude that the special day indeed has impact on the trade volume for that product.

Resources

1. Forecasting Intraday Trade Volume: A Kalman Filtering Approach (Ran Chen)
2. A theory of intraday patterns: Volume and Price Variability (A.R. Admati)
3. The behavior of daily stock market trading volume (B.B. Ajinkya)
4. Direct estimation of equity market impact (R. Almgren)
5. Return volatility and trading volume: an information flow interpretation of stochastic volatility (T.G. Andersen)
6. Optimal control of execution costs (D. Bertsimas)
7. Improving vwap strategies: a dynamic volume approach (J. Bialkowski)
8. Periodic market closure and trading volume: a model of intraday bids and asks (W.A. Brock)
9. Intradaily volume modeling and prediction for algorithmic trading (Browlees C.T)
10. Intra day bid-ask spreads, trading volume and volatility: recent empirical evidence from the london stock exchange (CX Cai)
11. Trading volume and serial correlation in stock returns (C.Y. Campbell)
12. On the volatility-volume relationship in energy futures markets using intraday data (J. Chevallier)
13. Trading halts and market activity: an analysis of volume at the open and the close (M.S. Gerety)
14. The intraday relationship between volume and volatility in liffe futures markets (O.A. Gwilym)
15. The intraday behavior of bid-ask spreads, trading volume and return volatility: evidence from dax30 (S.M. Hussain)
16. Trading volume: definitions, data analysis, and implications of portfolio theory (A.W. Lo)
17. Predicting intraday trading volume and volume percentages (V. Satish)
18. Intraday price change and trading volume relations in the stock and stock options markets (J.A. Stephan)
19. Daily Volume Forecasting Using High Frequency Predictors (L. Alvim)