Error Filter for the TRACE Database

Mariano J. Palleja*

January, 2023.

Abstract

This note presents the steps used to error-filter the Trade Reporting and Compliance Engine (TRACE) database. An adaptation to the Dick-Nielsen filter is proposed, particularly useful when processing memory and/or speed are binding. I present the structure of the code and the main filtering statistics. Considering the Academic TRACE information between January 2016 and December of 2019, our filter can match 99% of the 1.8 million targeted reports, filtering 4.2% of the 85.5 million observations.

1 Dick-Nielsen error-filter discussion and proposed adaptation

The TRACE Corporate Bond Data contains transaction-level data for US corporate bonds. Each observation is uploaded by the dealer that perform the transaction, and contains several trade specific characteristics [1]. If an observation is mistakenly uploaded or contains mistakenly uploaded characteristics, the dealer can indicate such error by uploading a subsequent referring report. The goal of an error-filter code is to delete both the referred and the referring report, leaving only those observations that are free of acknowledged mistakes. In this regard, the most widely used error-filter to clean the TRACE databases is the one proposed by Dick-Nielsen (2009, 2014, 2019). Though simple and popular, this filter is pretty time and memory consuming. In this note I propose an adaptation that can be run in a laptop computer. I test the code using the Academic version, period 2016-2019, and provide filtering stats.

In order to error-filter the TRACE data, we need to match referred and referring report. So let us firstly state some basic concepts used to perform such matching. If the referring and referred reports are uploaded within certain time window, these should share both the same trade characteristics and a unique identification number[2]. Otherwise referred and referring reports should only share the same trade characteristics. This drawback for matching this later reports is not relevant for observations uploaded after Feb 06, 2012, since in that case the referring report includes as an additional variable the identification number of its referred report. In the first case referring reports are called cancellations and corrections, in the second case are called reversals [3].

The filter proposed by Dick-Nielsen can be summarized in the following guidelines. Firstly it gathers all daily reports into a big database, and separate from such data base cancellations and corrections on the one

*University of California, Los Angeles. Email: marianopalleja@g.ucla.edu. I thank Shuo Liu and Mahyar Kargar for sharing their codes with me.

[1] If its an interdealer transaction, both dealers report, but with opposite trade signs.

[2] This time window was changed in February of 2012. Before Feb 04, 2012, referred and referring report share a unique identification number if both reports were uploaded in the same day. After that, they share a unique identification number if they are uploaded within a 21 day window.

[3] A cancellation is a referring report that indicates that the referred report didn't happened. A correction indicates that the referred report was uploaded with some error. Hence, a correction implies two reports: one that cancels the mistakenly uploaded observation and one that contains the correct information. Finally, a reversal is a cancellation or a correction uploaded after the window period. A reversal correction, just like a correction, implies a second report, which is marked as an as-of transaction.

hand, and reversals on the other. Secondly, cancellations and corrections are matched and deleted using identification numbers and characteristics. Finally, reversals are matched using characteristics. The size of the files constructed in each step turns the procedure extremely time and memory consuming, making it really hard to run in a laptop computer. For example, in 2019 we have over 24 million observations and 53 variables.

In order to reduce the size of the databases used in the filter, reports are usually split into smaller non-overlapping files, for example yearly or quarterly databases. It turns out that the Dick-Nielsen needs to be adapted before applying it to these smaller databases directly. The reasons being:

a) Cancellations and corrections reports, even if they refer to reports within the time window, might refer to reports in another data base.

b) Reversals reports refer to reports outside the time window. Hence, we should look the matching referred report in all databases with periods equal or before than the one to which the referring report belong. However, if we match using several databases we are back to the lack of memory-speed problem we initially have.

Lets state with more precision the aforementioned considerations. Suppose we build files with observations dated within a period of length k. For example, if observations run from Jan 1, 2017 to Dec 31, 2018, and we set k to be a quarter, we end up with the set $\{data_t\}_{t=1}^{T}$, with $T = 8$, where $data_1$ includes reports from January, February and March of 2017, and $data_T$ includes reports from October, November and December of 2018. Regarding the cancellations and corrections consideration, take for example a Jan 5, 2018 cancellation report, included into $data_5$, referring to an original report executed and reported 9 days before, on Dec 27, 2017, included thus in $data_4$. Obviously, if we look only within $data_5$, we wont match these reports [4]. Regarding the reversals consideration, consider a May 31, 2018 reversal report. The referred report could have been uploaded anytime before May 11, 2018. Hence we should perform matching algorithms through all $\{data_t\}_{t=1}^{6}$, which would be very time and memory consuming.

In this notes I propose an adaptation to Dick-Nielsen filter which takes into account both the memory-speed constraint and the two considerations mentioned. Particularly, I exploit the fact that referring reports are uploaded later than their referred report. This allows for a recursive structure. We start the filter within the most recent file $data_T$, store all unmatched reports referring to trading dates not in T, append them to the next most recent file $data_{T-1}$, and iterate until $data_1$. This avoids building and merging huge data bases, for which memory might be binding. Note that we suit consideration a) by keeping track of all referring reports that could not find a match because their referred report was reported on a date before the scope of the current file [5].

The filter structure can be summarized by the following steps

1. Create the set of files $\{data_t\}_{t=1}^{T}$
   (a) Merge each daily .txt file containing trade data with the corresponding daily bond characteristics file.
   (b) Stack all daily merged reports pertaining to a specific period of length k (in my code k is a quarter).
2. Filter errors
   (a) Set $t = T$.

[4] For example, the amount of canceling or correcting referring reports with such characteristic in 2017q1 was 737. The decrease in the matching rate implied if we don't take into account consideration a) is bigger the narrower the time frame included in each file.

[5] See section 4 to see how the method deals with referring reports that refer to previous referring reports.

(b) Create empty database $unmatched_{t+1}$.

(c) Load file $data_t$.

(d) Split file into $temp_t$, containing all referring reports, and $data\_clean_t$, containing all non referring reports [6].

(e) Add to $temp_t$ the reports in $unmatched_{t+1}$.

(f) Filter $data\_clean_t$ from reports matched with $temp_t$ [7].

(g) Subtract from $temp_t$ those unmatched reports referring to trades executed before the time range of $data_t$, and store such database as $unmatched_t$.

(h) Set $t = t - 1$.

(i) Iterate (c)-(h) until all files have been error filtered

Note that step 1.e and 2.e adds into $temp_t$ any referring report that calls a report executed before $data_{t+1}$ initial date. Hence, we take into account considerations a) and b) altogether. For example, say we have a referring report dated on March 4, 2017 included into $data_{2017,q1}$, which cancels a report dated on September 20, 2016 included into $data_{2016,q3}$. Firstly, when dealing with $data_{2017,q1}$, we will include the referring report into $temp_{2017,q1}$ through step 1.d. Later we will move it into $unmatched_{2017,q1}$ through step 1.g (the report we are looking for is outside of the quarter at hand and so it won't be matched). Secondly, when dealing with $data_{2016,q4}$, we will add such report to $temp_{2016,q4}$ through step 1.e, and we will further move it to $unmatched_{2016,q4}$ through step 1.g. Finally, when dealing with $data_{2016,q3}$, we will add such report to $temp_{2016,q3}$ and we will filter out the reports that needs to be canceled in step 1.f.

2 Matching specifications

We perform the algorithm in two steps. We firstly match referring and referred reports using a set of trade characteristics plus an identification number. This step cannot match all referring reports, so we perform another stage to work on those remaining unmatched reports. This second step uses the same set of variables, but not the identification number [8].

The set of variables used is:

- Bond id (cusip_id)
- Volume (entrd_vol_qt)
- Price (rptd_pr)
- Execution date and time (trd_exctn_dt, trd_exctn_tm)
- Buy/Sell indicator (rpt_side_cd)
- Reporting party anonymous id (rptg_party_id, rptg_party_gvp_id)
- Counter party anonymous identifier (cntra_party_id, cntra_party_gvp_id)

And the identification number variables are

[6] Referring reports include cancellations and corrections within and without the 20 days window, denoted by trd_st_cd="X","C","Y".

[7] Here we use a first-in-first-out approach. See the code for details.

[8] Given the different availability and quality of variables, it is useful to match reports before and after the February of 2012 using different methods. In this note I only address post 2012 files. The code and notes to filter pre 2012 change can be obtained upon request.

- "systm_cntrl_nb", when matching cancellations and corrections
- "systm_cntrl_nb" and "prev_trd_cntrl_nb" when matching reversals

Finally, in both steps we require that the reporting date and time of the referred report is not after than that of the matched referring report.

3 Results

The following tables presents the summary statistics of the filter. Considering the information between January of 2016 and December of 2019, our filter can match 99% of the $1,804,435$ referring reports, filtering 4.2% of the $85,538,612$ observations.

Table 1: Error Filter Statistics

Period	Obs	Cancellations	Corrections	Reversals	Can|Cor net	Reversals net	Can|Cor % matched	Reversals % matched
2019-q4	5,656,609	34,603	43,706	567	78,309	562	100.00	96.09
2019-q3	5,867,506	34,750	45,513	923	80,263	923	100.00	63.06
2019-q2	6,010,585	38,008	57,807	615	95,815	606	100.00	97.85
2019-q1	6,498,066	43,374	75,879	9,956	119,253	9,955	100.00	99.12
2018-q4	5,916,757	40,993	69,560	605	110,553	603	100.00	90.38
2018-q3	5,382,999	38,105	67,330	1,069	105,435	1,066	100.00	97.19
2018-q2	5,659,874	39,459	71,377	391	110,836	389	100.00	97.43
2018-q1	5,733,600	66,844	71,225	1,475	138,069	1,451	100.00	96.21
2017-q4	4,934,148	42,690	62,583	922	105,273	914	100.00	93.22
2017-q3	4,747,371	40,983	64,651	6,100	105,634	6,084	100.00	86.06
2017-q2	4,954,975	41,614	61,799	1,060	103,413	1,047	100.00	95.89
2017-q1	5,465,357	53,784	68,290	921	122,074	898	100.00	67.48
2016-q4	4,613,065	49,737	66,007	3,282	115,744	3,262	100.00	97.46
2016-q3	4,614,788	53,031	75,036	14,015	128,067	13,996	100.00	19.23
2016-q2	4,784,566	52,568	66,620	1,265	119,188	1,258	100.00	48.57
2016-q1	4,698,346	54,118	64,690	4,690	118,808	4,687	99.62	9.47

Note: The difference between the Reversals and Net Reversals is due to the fact that some reversals are canceled by a cancellation or correction, and hence we don't need to find them a match.

4 Robustness checks

- Referring reports correcting referring reports:

  The outline of our error-filter splits, for each database, referring reports on the one hand and remaining reports on the other. However, we could hypothetically have a chain of referring reports. For example, a cancellation of a reversal.

  The cases in which a cancellation or correction refers to a reversal are accounted by our code [9]. We first match all cancellations and corrections with any remaining report, including reversals. This explains the difference between the columns Reversals and Net Reversals in Tables 1 and 2. Analyzing different data bases, we couldn't find any reversal calling a previous cancellation, correction or reversal. As a consequence, the sum of Cancellations and Corrections columns equals Net Can|Cor column in Tables 1 and 2. In this regard, the use of reversals to indicate that a previous cancellation, correction or reversal

[9] See R code

was erroneously reported would imply that the original report actually happened. Presumably because the same result can be obtain by uploading an asof_cd="A" report, we don't observe any of these cases.

- Small reversal matching rate in files 2016-01 and 2016-07:

  The reversals matching rate drops sharply for the first and third quarters of 2016, periods in which the number of reversals are several times higher than on average.

  About the first quarter of 2016, we find that a single dealer causes the big majority of such reduction. From the 4,687 reversals we need to match, the dealer with anonymous identifier $1082873b3e37ced5b81df37dc449e3c943effcdc$ uploaded 4,413 of them. All of these reports were uploaded on March 30, 2016. Their execution date is distributed between Feb of 2016 and November of 2015. Only 337 of these reversals are matched, explaining thus almost all the reduction in the matching rate. We try making a less restrictive algorithm, reducing the set of matching variables (price, volume, counter party identity, etc) but we cannot find the corresponding referred reports.

  Regarding the third quarter of 2016, we find a similar pattern. A single dealer causes almost all the reduction in the matching rate. From 13,996 reports we need to match, the dealer with anonymous identifier $d227d0fd448bbe76398183f61a9da7db4e7faf33$ uploaded 11,134 These reports were uploaded on Sept 20, 2016, all within 10 minutes: between 16 hs 19 min and 31sec and 16 hs, 29 min and 42 sec. Their execution date is distributed between June of 2016 and November of 2015. The curious thing is that, if we don't impose any uploading time restriction, we can match 10,022 out of the 11.134 targeted reports. Particularly, all the matched referred reports were uploaded in the same day of the reversals, but between 16 hs 30 min and 31 sec and 16 hs 37 min and 20 sec, i.e 10 minutes after (!) than the upload time of the reversals.

  In both cases, all these reversals are with non-member affiliates (cntra_party_id = cntra_party_gvp_id = A), not with customers or other dealers.

References

Dick-Nielsen, J. (2009), *"Liquidity biases in TRACE"*, Journal of Fixed Income, 19(2), 43-55.

Dick-Nielsen, J. (2014). *"How to Clean Enhanced TRACE Data"*, Technical report, Available at SSRN: http://ssrn.com/abstract=2337908 or http://dx.doi.org/10.2139/ssrn.2337908

Dick-Nielsen, J. and Poulsen, T. K. (2019). *"How to Clean Academic TRACE Data"*, Technical report, Available at SSRN: https://ssrn.com/abstract=3456082 or http://dx.doi.org/10.2139/ssrn.3456082