Using Manipula and CATI SubPriorities to Adjust and Fine-Tune Sample Delivery

Dave Dybicki, Joe Matuzak & James Wagner, The University of Michigan

Introduction

A constant concern of study managers is controlling sample delivery to maximize productivity. From a business perspective, the goals are to increase efficiency and thereby cost savings. Blaise 4.8 introduced two new instructions that allow easy adjustment of key fields within the current daybatch, (Daybatch_Transfer, and Daybatch_Mod) and combining these with the SubPriority field introduced in Blaise 4.81 provides the ability to shape delivery on a sample-line level.

Here we describe techniques we use to programmatically adjust delivery priorities at regular times during the day so as to ensure balanced delivery of sample from within two experimental groups while also ensuring the sample will appear in the desired sequence. These techniques can easily be extended to provide on-demand changes within a daybatch, allowing managers the ability to immediately adjust delivery priorities as needed.

Goals and Challenges

There were three main goals to our effort:

1. Maximizing delivery control for a propensity modeling experiment.
2. Giving more control to managers for sort orders
3. Providing the ability to make priority changes without interrupting production

The experiment was designed to test different protocols for delivering cases to interviewers for dialing. The experimental protocol used a statistical model to estimate the "call window" with the highest probability of contact for each case. A "call window" is a slot of time defined by the day of week and time of day. For example, Tuesday through Friday from 5pm to 9pm is a call window. Under the experimental condition, if a case had its highest probability of contact in the current window, then it would be sorted to the top of the case priority list for calling. This was done for a randomly selected half of the sample, and the remaining cases were prioritized by the study manager.

The experiment required three special programming tasks to be performed.

1. The experimental and control cases had to be sorted at the beginning of each call window in an interleaving fashion so that each condition (experiment and control) would receive approximately equal treatment.
2. Several groups of interviewers had to be included in the experiment.
3. The "manual" management of sample had to be tracked in order to detect any confounding of the experiment that may occur.

The first requirement was that the Blaise SMS sort the list of cases in a priority order that included both the experimental and control cases. It was decided that this could best be done by interleaving cases from each group. One difficulty was the sort priorities could be assigned for cases that were not part of the daybatch. This could occur if a case was prioritized, but was scheduled for an appointment on another day. In order to address this issue, Blaise SMS scripts were written that drew sampled cases from the top of a priority list that contained all non-finalized cases and determined if they were part of the daybatch. This process was repeated until it found 5 lines each from the experimental and control groups that were part of the daybatch. These groups of 5 were then interleaved into a priority list – 5 experimental cases, followed by 5 control cases, 5 experimental cases, and so on. The whole daybatch was prioritized in this way.

This procedure was repeated at the beginning of each call window. Since we generally call in 4 time zones, this meant the sample was sorted as many as 5 times per day. Fortunately, the most recent version of Blaise SMS allows the daybatch to be sorted without shutting down production.

Setting Priorities

A key goal was creating a sorting method that gave greater control than the internal Blaise algorithms so study managers could use this to prioritize sample according to changing study needs. Conversations with various study managers produced a list of areas they wanted to prioritize, including time zone, call number, release or replicate number, whether or not contact was made on a case, whether or not a household listing was completed, whether there was resistance on a case, and the sampletype of the case.

Managers can apply whatever numeric weight they desire to any of these criteria, and those weights can be changed at any time. This allows them to give huge priority to any particular area by giving it a large number, or to aggregate small weighting totals so cases meeting multiple criteria rise to the top of the list.

A timed script simply calculates the numeric total for each sample line, and then sorts them in descending order, so those with the highest total come first in the list and will thus be assigned the highest subpriority.

Program Flow

```
PROCESS ChangeDaybatchEntries
USES
  SMS
  DayBatchMeta

  DATAMODEL Daybatch
    FIELDS

      dJoinID          : STRING[8]
      dStatusCode      : STRING[20]
      dStartInterval   : STRING[20]
      dEndInterval     : STRING[20]
      dFutureStatus    : STRING[20]
      dNrOfDials       : INTEGER[2]
      dDialInterval    : STRING[20]
      dNrOfDialsBusy   : INTEGER[2]
      dQuotaIndex      : INTEGER[4]
      dGroup           : STRING[12]
      dInterviewer     : STRING[12]
      dTimeDifference  : STRING[8]
      dSliceID         : STRING[8]
      dSliceInfo       : STRING[20]

  ENDMODEL

...

TEMPORARYFILE MyDaybatch:DayBatchMeta

...

SMSData.DAYBATCH_TRANSFER(MyDaybatch)
...
WHILE More_Records = YES DO

        DISPLAY('Propensity Sampleid '+ xSampleid)

        IF (InputFile3.RESULTOK) THEN
            xJoinID := InputFile3.yJoinID

            DaybatchFile.GET(xJoinID)
            IF DaybatchFile.RESULTOK THEN
              IF (DaybatchFile.dStatusCode <> '4') and (DaybatchFile.dGroup <> 'z1')
                and (DaybatchFile.dGroup <> 'z2') and (DaybatchFile.dGroup <> 'z3') THEN

                SMSData.GET(INTERNALKEY,val(xJoinID))
                IF SMSData.RESULTOK THEN
                    IF (SMSData.Uservalue[19] = '1') OR (SMSData.Uservalue[19] = '2') THEN

                        SMSdata.DAYBATCH_MOD('SubPriority ', str(xSubPriority))
                        Counter := Counter + 1

                    ENDIF
                ENDIF
              ENDIF
            ENDIF
        ENDIF
...
xSubPriority := xSubPriority – 1
...
ENDDO
```

SMS refers to our sample management Blaise database

A Uservalue is a 100 character field in our Sample Management database that can be used for storing any type of data. There are 120 uservalues.
Dgroups are specific interviewers who have a special skill

In general, this script does the following:

Loads in the text files from experimental and control groups
Grabs the first 5 from the experiment file and assigns a 99 to subpriority
Grabs the next five from the manager's file and assigns 99 to subpriority
Subtracts 1 from subpriority and repeats the process
Repeats until there is no more sample or subpriority = 0

Transferring daybatch to a file

dStatusCode '4' equals a 'no need today' status in the daybatch

z1, z2, and z3 are our special interviewer groups for refusals

This checks the uservalue to determine whether the sample line is part of the experimental or control group

Setting the subpriority using DAYBATCH_MOD

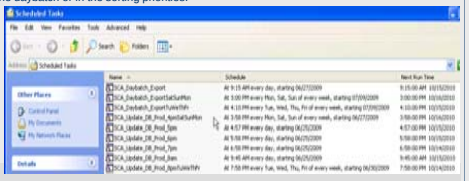
Subpriority Logic Flow

1. Manager must edit script to set priorities
2. Use DAYBATCH_TRANSFER to export the daybatch to a file.
3. Read in all cases from current daybatch.
4. Check to see if sample line is part of experiment
5. Check to see if sample line matches criteria for sort priority
6. Create variables for each of the following priority possibilities
7. Determine the value of each variable by matching as below
8. Continue till all priorities are checked
9. Determine subpriority for the sample line by adding the variables
10. Write Subpriority value
11. Continue to next sample line until all are completed

Automation

Using Windows Scheduled Tasks, we are able to have variations of these scripts update the daybatch at specific times of day, which can vary by day, so we can coordinate efficiency with variations in facility hours, with experimental priorities, etc. The script can also be run on demand, of course, so the daybatch can immediately be updated after creation or after any change a manager needs to make either in the daybatch or in the sorting priorities.

In this instance, there are two scripts: one to export priorities, one to rewrite the subpriorities. One time window covers daytime calling, the daybatch is updated each hour of weekday evenings, and weekends have their own schedule.



Outcomes

If this procedure is successful, we should see a relatively even balance of calls on experimental and control cases governed by this algorithm (excluding, appointments, for example) across the hours of the day. The following table shows that this balance was largely achieved. Imbalances in late hours are likely due to variances within sample size within the particular time zone.

Hour	Total Calls	Experimental Calls	Control Calls	Percent Experimental
10	2396	1157	1239	48.3%
11	1586	795	791	50.1%
12	3954	2056	1898	52.0%
13	2955	1542	1413	52.2%
14	4086	2198	1888	53.8%
15	4241	2173	2068	51.2%
16	4223	2428	1795	57.5%
17	9205	4425	4780	48.1%
18	13115	6951	6164	53.0%
19	13351	6411	6940	48.0%
20	11232	6092	5140	54.2%
21	7568	3902	3666	51.6%
22	4867	2225	2642	45.7%
23	2910	1266	1644	43.5%
Total	**85689**	**43621**	**42068**	**50.9%**

In addition to using this as part of an experiment on a study with a smaller amount of sample and short production time, it has also been implemented as part of a large-sample production project with a longer timeline.

This procedure allowed the managers of that project to have larger than usual daybatch sizes which could then be weighted to emphasize nightly production targets. In this situation, past waves had difficulty mixing priorities where, for example, older cases needed to be delivered a third of the time and newer cases the remaining times,

Future Directions

Our initial development effort was spurred by trying to get this experiment working successfully, but there are many other pieces that can be added. First, a control panel would allow project managers to make changes to priorities without needing to understand Manipula, and that panel could easily contain a toggle to turn on or off the interleave sort, so managers would control the entire daybatch. It could also allow immediate running of the scripts, so managers could make changes and have them take effect right away as opposed to waiting for a scheduled task.

For Further Information

Please email ddybicki@isr.umich.edu or jmatuzak@isr.umich.edu for more information.