Sawtooth Software

RESEARCH PAPER SERIES

Analysis Options for Patient Chart Studies

Keith Chrzan
Sawtooth Software, Inc.

3210 N. Canyon Rd., Provo, Utah
+1 801 477 4700
www.sawtoothsoftware.com

Analysis Options for Patient Chart Studies

Keith Chrzan, Sawtooth Software

Introduction

Not all choice models we build are experiments[1]. One non-experimental choice model I run for my pharmaceutical clients involves modeling physicians' prescribing decisions using data taken from patients' medical charts. Patient charts provide rich data about individual patients, including their demographics and "diseaseographics" (facts about their disease, its treatment and progression, and about concomitant conditions). Add in data about each patient's insurance coverage and about the physicians and their "practiceographics," and we have potentially a wealth of information to use to understand prescribing decisions. Moreover, using patient charts may ameliorate one of the serious handicaps of pharmaceutical marketing research, severely limited sample sizes: often we can collect three or five or even 10 patient charts per physician (or office nurse) respondent.

You can find below descriptions of three choice analysis options for patient chart databases, plus a suggestion for combining this kind of revealed preference (RP) data with experimentally designed stated preference (SP) data.

Analyses for Patient Chart Data

Polytomous multinomial logit (MNL)

This model is a special case of the conditional multinomial logit model we use to estimate utilities for choice-based conjoint analysis and for MaxDiff scaling.

We can apply the polytomous MNL model, described in the earlier white paper as a way to analyze patient type experiments, to non-experimental patient chart data. In a nutshell, the model uses any variables collected for the patients and any variables that are specific to the physician-prescriber to predict a multiple choice dependent variable about the prescriber's therapy choice.

Let's take a simple example and imagine a chart study where we collect patient's age and gender, as well as whether the doctor prescribed (1) over-the-counter (OTC) medication or (2) a prescription painkiller or (3) physical therapy for a patient's headache. The model resulting from the polytomous logit analysis might look something like this (of course in an actual study we'll usually have many more variables than two):

Variable	OTC	Prescription	PT
Constant	0.48	0.33	0
Age 18-44	1.45	-1.95	0
Age 45-64	-0.62	-0.78	0
Age 65+	0	0	0
Male	-1.22	-0.54	0
Female	0	0	0

Usually, the last prescribing choice becomes the anchor or reference level, with all other utilities scaled relative to it. From these utilities we could predict therapy choice for a patient, given the patient's age and gender, using the standard logit choice rule. For example, the utility for each therapy for a 63-year old male patient would be

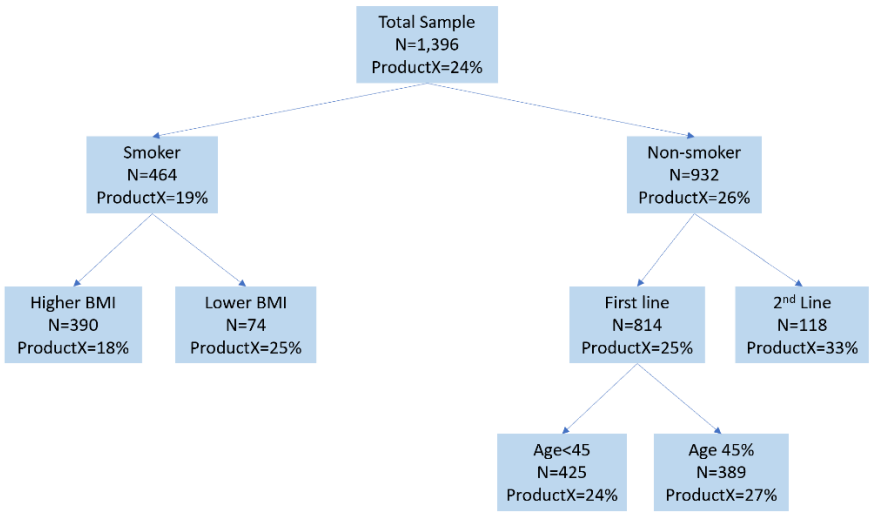
OTC: 0.48 - 0.62 - 1.22 = -1.36
Prescription: 0.33 -0.78 -0.54 = -0.99
PT: 0 + 0 + 0 = 0

Taking the ratio of the exponentials of these utilities yields predicted shares of 15.8% for OTC pain reliever, 22.8% for a prescription pain reliever and 61.4% for physical therapy. We could thus build a simulator allowing us to predict therapy shares for any patient whose age and gender we've collected. The polytomous MNL model provides confidence intervals and stat tests as you'd expect from a statistical model. It is available in Sawtooth Software's Menu Based Choice (MBC) software. There are also ways to run this model in our CBC/HB and CBC/Latent Class software packages, but it's easier to do in MBC.

The next two analysis options both use machine learning methods rather than statistical models.

Decision trees

In addition to modeling the choices, a decision tree also allows us to visualize the decision process (it's our go-to method when clients ask for a "decision hierarchy"). A decision tree analysis repeatedly and sequentially splits the sample of observations (patient charts). To start off, the analysis finds the single most predictive variable and splits the sample based on it into two subgroups. It then looks at each subgroup in turn to see if they can be significantly split. For example, in a recent patient chart study we looked at a large number of possible predictors of therapy choice, and the client was most interested in seeing the predictions of choice of their new product, Product X:



The analysis shows that the most significant split of the patient charts to predict Product X choice is whether the patient is a current smoker (19% of whom were prescribed Product X) or not (26% prescribed Product X). Among the smokers, another significant result splits the sample further based on

whether the patient has a higher or lower BMI. On the non-smoker side of the tree the sample splits twice more, once for whether the treatment is first line or second line (second line means the patient had some other therapy that failed and is now looking for a "plan B") and again on age. You can see that the tree identifies segments of patients with Product X shares as high as 33% and as low as 18%.

There are several decision tree algorithms available which operate on different kinds of fit metrics and splitting criteria. Some of them tend to produce slightly branchier trees and some tend to produce less branchy trees.

As noted, the tree provides a nice visualization that appeals to some clients (it's certainly more intuitive to read than the polytomous MNL utilities in the pain relief example above). And if the client wants a decision hierarchy, the tree sure looks like one. In addition, trees have a simple if-then structure which is easily built into an Excel simulator. A final benefit of a decision tree analysis is that we can put a large number of variables into the tree and let it tell us which ones are important: in the example above, a total of over 200 candidate predictors were submitted to analysis, though the tree only ended up including four of them. That means we as researchers can treat the analysis as more exploratory and let the model tell us which variables drive decisions and which do not.

Trees usually end up with a limited number of variables/splits and it's well known that when two predictors are correlated, a tree might take in one of the variables and exclude the other. For example, imagine a case where choice of therapy depends on income and education (maybe it's a complex therapy that requires a great deal of patient compliance). Because income and education are correlated, it may be that one of the two enters the tree as a predictor and the other is left out. For this reason we sometimes use a second machine learning method for analyzing patient chart data: the random forest.

Random forests (RF)

As the name suggests, instead of a single tree we might run 500 or 1,000 trees. Without going into great depth, a random forest contains a pair of randomizing steps in the composition of these trees that "decorrelates" the forest as a protection from harm due to corelated variables. The result isn't a diagram, but a set of derived importances we can use to quantify how much influence each predictor variable has on the physician's therapy decision. For example, we might get a set of importances like this:

Variable	Importance
Patient age	3.22%
Patient income	14.52%
Patient education	15.47%
. . .	. . .
Physician specialty	1.89%

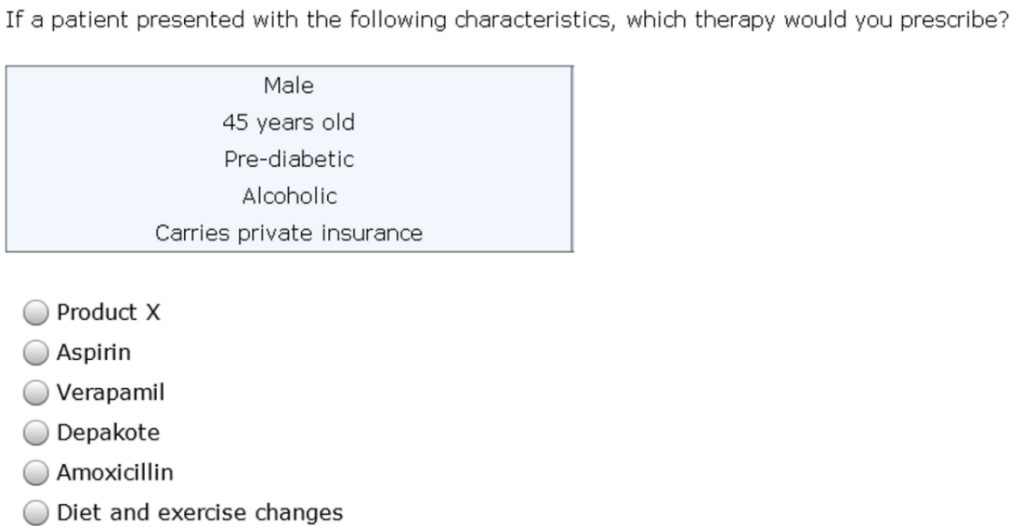
From these importances, it's clear that income and education are both important predictors.

Like decision tees, RF allows us to include large numbers of predictors so that we can explore the nature of variable relations rather than assuming we know which variables do the best job of predicting. Much as I dislike "garbage can" models, if you're going to run one, you could do a lot worse than using a random forest.

Unlike decision trees or polytomous MNL, however, a RF simulator will be harder to deliver to your client; a forest made from 1,000 trees, for example, would need to use all 1,000 trees to make predictions (this is easy enough to do in R, where the program stores the forest but it would be a tall order to make in Excel, where you might need a separate sheet for each tree).

Fusing RP and SP Patient Data

On occasion I've even combined a patient chart study with data from a designed experiment. For example, we might from a physician's office assistant collect records from 10 patients and then of the physician ask a set of eight questions like this one:



We build the patients in these 8 questions according to an experimental design, like in conjoint analysis, using the same attributes we intend to collect from the patient records.

Now we have 18 observations from each physician: 10 from his patients' charts and eight from the experimentally designed patients in the questions above. Even if we started with a limited sample of 150 physicians, we now have a database of 2,700 observations of patient therapy decisions we can model, using polytomous MNL, a decision tree or a random forest.

Summary

After introducing the idea of patient chart data, we reviewed a statistical model and two machine learning approaches we can apply to patient chart data. We also covered a way to combine patient chart data with a conjoint-like patient type experiment to get the benefits of data fusion.

Footnotes

[1] For a review of choice experiments appropriate in pharmaceutical markets see https://sawtoothsoftware.com/resources/technical-papers/choice-experiments-for-pharmaceutical-market-research