An Analytic Method for Identifying Potential Associations between Drugs and Side Effects from Large-scale Clinical Databases

Ed Ramsden
October 2010

www.edscave.com
EdR@sensorlytics.com

Background:

In 2009-2010 the Foundation for the National Institutes of Health's Observational Medical Outcomes Partnership (OMOP) sponsored a competition (http://omopcup.orwik.com) to develop improved data analysis methods for identifying potential relationships between drugs and side-effects from clinical data. For the competition, OMOP provided a large-scale database of simulated data for ~10,000,000 patients, which included variables such as age, gender, drug treatment regimens, and diagnosed symptoms. The specific task set out for the competition was to discover which pairs of medications and symptoms were related and might be worthwhile candidates for more in-depth study. This paper describes the method I developed for performing this analysis.

My Method:

The methods I chose to explore were based on the frequency-count methods described in [1], and are most closely related to the Reporting Ratio (RR), Proportional Reporting Ratio (PRR), and Reporting Odds Ratio (ROR). All of these methods use rely on frequencies of the relation between a given drug and a given symptom:

	Symptom Present	Symptom Absent	Total
Drug Present	W_{11}	W_{01}	$W_{11} + W_{01}$
Drug Absent	W_{10}	W_{00}	$W_{10} + W_{00}$
Total	$W_{11} + W_{10}$	$W_{01} + W_{00}$	

There are also several possible options that can be used for counting – an important consideration. One scheme is to count patients – how many show the symptom with drug, how many don;t have the symptom with the drug, etc.. – to correspond with entries in the above frequency table.

One can also count by patient-days. Each day in the patient history is considered individually as to whether or not a symptom occurred or a drug was prescribed. This method of counting accounts for variations in patient exposure to a drug – relating total drug-days of exposure to symptom occurrence. Although this is somewhat more computationally intense than simply tallying by patient, it is computationally feasible on on even a modest personal computer. For example, the OMOP cup database contains approximately 10 million patient records, with a total of approximately 6 billion total patient observation days. Despite increasing the computational requirements nearly 3 orders of

magnitude, reading the patient data in from disk was still the most time-consuming part of the analysis. For my analyses, I chose this method of counting.

An alternate view of the proportional methods is as probability of a symptom occurring while taking a drug (P_A) versus the probability of the symptom occurring when not taking the drug (P_E).
One characteristic of all of the above frequency-count methods is that they only report the *strength* of the observed relation. In cases where there are a large number of relevant observations of symptoms both with and without drugs this is not a problem. It is possible, however, in cases with few observations that random sampling variation can result in extraordinarily high values of the metric, leading to a high proportion of false positives.

One solution is to use a metric based on statistical significance. The chi-square technique is a well known metric for interpreting the significance of frequency tables. When one reduce the number of categories of interest to two (one degree of freedom), chi-squared can be interpreted as a normal Z-score [2]:

$$Z = \frac{(P_A - P_E)}{\sqrt{\frac{P_E(1 - P_{E)}}{N}}} \approx (P_A - P_E)\sqrt{N/P_E}$$

While the values of Z obtained in this manner can be extremely high (>1000) and not meaningful in terms of significance as applied to a normal distribution, they can be effectively used as a way of ranking the 'significance' of drug-symptom relations in a way that accounts for the sample size (confidence of the measurement) as well as the strength of the measurement.

My Algorithm:

The first step is to pre-process the data so that symptom and drug histories are associated with each patient in the data-set. This will result in the following (or similar) data structure for each patient:

```
Record Patient
        Obs_Start            ' start date for patient observation period
        Obs_End              ' end date for patient observation period

        N_Symptoms           ' # of symptoms associated with patient
        Symptom_Type[1..N]   ' symptom ID
        Symptom_Date[1..N]   ' symptom report date

        M_Drugs              ' # of treatments associated with patient
        Drug_Type[1..M]            ' drug ID
        Drug_StartDate[1..M]       ' drug treatment start date
        Drug_EndDate[1..M] {date}  ' drug treatment end date

    EndRecord Patient
```

The other major problem beyond measurement of significance occurs when a symptom is observed

while two or more drugs are currently being prescribed – how do you identify the drug that is actually associated? For this reason, my algorithm consists of two separate phases; a first pass that is responsible for identifying the most likely 'offender' in cases where tow or more drugs are prescribed, and a second pass that allocates the count.

Some Definitions:

S – Symptom 'S'
D – Drug 'D'
D_P – Total # of patient-days in data set (~ 6 billion)
$N_S[S]$ - # of occurrences of symptom 'S' in entire data set
$N_{SD}[S,D]$ # of *credited* occurrences of symptom 'S' while exposed to drug 'D'
L[S,D] – 'Lift' factor describing relative increase in frequency of 'S' while
        exposed to 'D' versus frequency in entire data set.
$D_D[D]$ – Days exposure to drug 'D' in entire data set

Phase One – Initial Tally

```
For Each Patient Record

        D_P += Observation days for this patient

        For Today = Start_of_Observation_Period to End_of_Observation_Period

                Get List of Symptoms in use Today
                Get List of Drugs in use Today
                Get count of drugs in use Today

                Credit_Factor = 1/(count of drugs in use today)
                For each Symptom 'S' occurring Today
                        N_S[S] += 1   { total symptom count }
                        For each Drug 'D' in use Today
                                N_SD[S,D] += Credit_Factor
                        Next Drug
                Next Symptom

                For Each Drug 'D' in use Today
                        D_D[D] += 1  { total # of exposure days}
                next Drug

        Next Today

Next Patient Record
```

At the end of this phase, counts for the number of total patient days, number of overall occurrences of each symptom ($N_S[S]$), total # of prescription days for each drug ($D_D[D]$) and split occurrences of each symptom when accompanied by a drug ($N_{SD}[S,D]$) are available.

The next step is to determine apportioning factors for either splitting or crediting counts to drugs in cases where multiple drugs occur with a given symptom. While a relative likelihood metric such as

$N_{SD}[S,D]/D_D[D]$ could be used, I have found that this gives excessive weight to single symptom events that happen to occur in the course of a rarely-prescribed drug. One way to ameliorate this problem is to define a metric based on the actual ($N_{SD}[S,D]$) and expected counts ($D_D[D] * N_S[S] / D_P$) for the # of symptoms occurring. A 'lift ratio' can then be defined for each drug-symptom pair as:

$$L[S,D] = \frac{N_A + 0.5}{N_E + 0.5} = \frac{N_{SD}[S,D] + 0.5}{D_D[D] \times N_S[S]/D_P + 0.5}$$

The lift ratio is computed for each combination of drug and symptom (~22 million in the OMOP cup dataset) for use in phase two.

Phase Two – The Actual Tally

 The second phase of my algorithm uses the lift ratio calculated in phase one to decide how to apportion each drug-symptom event among the multiple drugs it may be truly associated with. The algorithm is similar to that of phase one:

```
Clear N_SD[,]
For Each Patient Record

        For Today = Start_of_Observation_Period to End_of_Observation_Period

                Get List of Symptoms in use Today
                Get List of Drugs in use Today
                Get count of drugs in use Today


                For each Symptom 'S' occuring Today
                        Factor_Total = 0
                        For Each Drug 'D' in use Today
                                Factor_Total += L[S,D]
                        Next Drug

                        For each Drug in use Today
                                N_SD[S,D] += Lift[S,D] / Factor_Total
                        Next Drug
                Next Symptom

        Next Today

Next Patient Record
```

Alternatively, the $N_{SD}[S,D]$ associated with the drug with the highest lift (for each symptom occurring that day) can be credited with a count of '1' in a winner-take-all variation of this algorithm. Although this approach seems like it might be a bit more defensible than splitting the credit up, it results in slightly lower performance.

Now with the final $N_{SD}[S,D]$ resulting from the second phase, one can calculate a 'Z-score' for each combination of symptom and drug. Counts are first converted to probabilities of a symptom occurring on a given day with the drug ($P_A[S,D]$) and overall across all data ($P_E[S]$).

$$P_A[S,D] = N_{SD}[S,D]/D_D[D]$$
$$P_E[S] = N_S[S]/D_P$$
$$Z[S,D] = (P_A[S,D] - P_E[S])\sqrt{D_D[D]/P_E[S]}$$

Positive values of Z-score can be interpreted as a positive association between the drug and the symptom (e.g. drug potentially causes symptom), negative Z-scores a negative association between the drug and symptom (drug potentially suppresses symptom) , and Z-scores near zero can be interpreted as a lack of observable relationship. To satisfy the requirements of the OMOP Cup scoring system, raw Z-scores were scaled and offset into a positive integer between 0 and 10000.

Results & Miscellaneous Implementation Details:

When the linear apportioning method (phase 2) was applied to the OMOP Cup data set, this resulted in a solution with a score of 0.2189, which compares favorably to more complex methods such as BCPNN and BLR. Calculating this result required roughly one hour of wall-clock time under the following conditions:

1) The data set was already preprocessed to a format similar to that described on page 2. A binary file format was used to minimize disk read time.
2) The processor was an Intel E5300 (dual core Pentium) with 4 GB of main memory
3) Only one processor core was used (the code was not parallelized)
4) The program was coded in Microsoft VB.NET, running under Microsoft Windows 7 (64 bit mode)

Because of the memory constraints of the computer I used, it was not convenient to load the entire data set into memory prior to processing. Instead, patient records were sequentially read and processed, which contributed substantially to the overall processing time. The algorithm's ability to handle the data in a one-at-a-time manner, does however offer the advantage of making readily scalable to databases containing even larger numbers of patients. Also, if larger data sets need to be handled, this method can be readily parallelized by tabulating counts from sub-groups of patients and combining the tallies at the end of each of the two phases.

References:

[1] DP Method Specification (part of 'Disproportionality Analysis Phase 3 Testing.zip'), *http://75.101.131.161/download/loadfile.php?docname=Disproportionality%20Analysis%20Phase%203*

[2] Jay L. Devore, *Probability and Statistics for Engineering and the Sciences, 7th Ed.*, Thomson Brooks/Cole 2008, Belmont CA, pp.568-572.