Personalized Statistics

Compsim White Paper

With advanced search engines, text analytics, semantic processing, with the ability to tie everything together with cloud computing, and with the internet, we have the ability to calculate probabilities and statistics for large populations of generalized items. The problem is that all "generalized items" are not the same. The concept of personalized medicine highlights this issue. The "holy grail" idea is that with the sequencing of the genome, along with testing the compatibility of different medicines with their genetic codes, healthcare providers will be able to match medications with individuals for the "perfect" treatment of a disease.

Perhaps, however, things are not quite so simple.

Stepping back to a simpler problem: the diagnosis of anemia[1]. To test for anemia, a blood sample is taken from an individual and run through a hematological analyzer. A series of tests are made to count the levels of a number of blood characteristics. The results are presented to a pathologist. The pathologist compares the results (along with age and gender of the patient) to statistical values to determine if the tested values are below normal, normal, or above normal. The pathologist then compares the results with patterns to determine the different types of anemia. When the pattern is matched to a certain type of anemia, the diagnosis is delivered to the healthcare provider who then suggests the appropriate treatment to the patient.

This process works if the patient's blood sample meets the "statistical norm" AND if there are no other influencing factors. For example (example only), perhaps some of the hematological tests might be influenced by other factors such as:

- the patient drank water before the test (amount, timing).
- the food the patient ate before the tests might influence the test results.
- there is a geographic impact or ethnic impact (not considered in the general statistics), or even the environment that the patient was exposed to in his/her work.
- the activities the patient performed prior to the test influenced the test results.
- the time of the day the tests were taken influenced the results.
- the level of stress the patient was under influenced the results.
- prior medications would influence the results.

What this suggests, is that the information collected and stored on the cloud could be more valuable if it was interpreted with an influencing filter. Using KEEL, we can use statistics to define the curves for the different blood tests. In the case of anemia, the medical industry has provided "some" characterization by age and gender. So there are lots of "somewhat" standardized curves. A patient might jump from one set to another as they age. With KEEL, we can bend the curves rather than jumping from dataset to dataset.

---

[1] Diagnosing Anemia Demonstration: http://www.compsim.com/demos/d60/anemia.htm

Also, with KEEL (given knowledge from research) one could adjust the standardized curves based on the influence of different factors.

It would be far easier to model the influence of different factors on the test results, than to attempt to gather statistics on all of the different combinations of influencing factors.

A user interface could be developed on the client side that would allow the user to enter, not only the age, gender and test results, but also the relative influencing factors. The influencing factors are likely to be relative (some amount of x). The information set would then be sent to the cloud for processing. Then a KEEL Engine would bend the generalized "curves" as defined for the influencing factors. Finally, the results would be developed for a personalized response.

To assist the healthcare provider, the response would show the personalized response next to the one that included the influencing factors. This would then be a "decision aid" for the clinicians.

Another benefit of using KEEL to generate the results is that an additional report could be generated "explaining" why other diagnoses might be incorrect. This would assist in reducing human error which is a major issue in the healthcare industry.

From a broader perspective, the benefit of using KEEL to create these models is an economic one. The conventional development process includes transferring an idea developed by a domain expert to a mathematician who develops a formula. The formula is then translated into code by the software engineer who has to debug the code and insert it into some test environment; all before the domain expert gets his or her first pass at reviewing the model. An iterative cycle takes place before the application can be completed. Using KEEL, the domain expert can avoid the iterative cycle of translating from one form to another before testing the model. The domain expert can test the model using the KEEL Dynamic Graphical Language as it is being refined and then give "working code" to the software engineer for insertion into the production application as soon as it is ready.

It is also likely that new influencing factors will be identified over time. There will be a need to quickly integrate new factors into the models. Time and money are always an issue, and the use of KEEL can save money by accelerating the development process.