Webgression:
Automatic Inference of Claim Propensity from Web Search Results

Tyche Analytics Company

May 2, 2017

Executive Summary

Casualty underwriters make decisions on the basis of limited information, so they must make the best use of the data available to them at the time of underwriting. One major overlooked source of data is the principal's name itself. As we will see, names turn out to be a highly predictive indicator of claims experience when properly analyzed. Tyche has developed a method for automating a common procedure that an underwriter might informally apply when evaluating a potential submission: *typing the name of the establishment into a search engine and evaluating the results.*

Using machine learning and natural language processing, we formalize and automate the process of predicting claims experience from hypertext in a technique we call *Webgression.* Webgression is highly predictive of claim outcomes. In our first pilot study, for example, Webgression scores were a better predictor of claims experience than the commonly-used insurance industry association internal risk classification scheme. Webgression scores are available via API and can yield claim probability predictions, or augment the feature set of an existing model.

Introduction

To understand Webgression, imagine what you might do if you, a casualty lines underwriter, had to evaluate a new submission for a business owner's, or other small business coverage, policy using nothing but the principal name of the applicant. You might try typing the name into a search engine and seeing what comes up in the first page of results. You might skim over the page and take a general impression. Certain words or phrases like “`five stars`”, “`exceeded expectations`”, or “`would recommend`” suggest that the establishment in question is managed well, and hence less likely to generate claims. Conversely, “`lawsuit`”, “`Better Business Bureau`”, or “`passersby were amazed by the unusually large amounts of blood`” might signal that the submission represents a conspicuous exposure.

An underwriter performing this process manually faces several challenges. First, the process is time-consuming. Second and more importantly, the underwriter can bring only their intuition to the task of evaluation, and that intuition can only be honed over a relatively small number of cases. An underwriter who searched one name per minute for an entire 40 year career would not be able to process the workload that a consumer laptop could handle in an hour.

Through formal statistical analysis of such datasets, we have found that good and bad risks tend to differ not only in rare, spectacular,and obvious language (*e.g.* “`wonderful`”vs. “`life-threatening`”, but in their common, mundane and subtle results (*e.g.* “`LLC`” vs. “`INC.`”) as well. Webgression allows underwriters to interpret the rich stream of internet data not only more quickly, but also more rigorously and effectively, than they could do by hand.

Methods

Given a set of training data consisting of establishment names and binary class labels (1 for **claim**, 0 for **no-claim**), we wish to construct a classifier to predict outcomes for novel names. To begin, we run each name in the training data through a search engine and record the first page of search results. We then simply count the number of times that each word appears in each set of results, yielding the following matrix:

$$\begin{array}{rcc} & M \text{ vocabulary items} & \text{outcomes} \\ N \text{ names} & \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1M} \\ c_{21} & c_{22} & \cdots & c_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & \cdots & c_{NM} \end{bmatrix} & \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \end{array}$$

where c_{ij} is the count of the jth item in the vocabulary in the search results for the ith name. Here, the vocabulary is simply the set of all space-delimited tokens that appear in the search results after stopping, stemming and converting to lowercase.[1]

Now, since we are training the model to predict *historical* claims outcomes with *current* search results, we also restrict the vocabulary to terms that appear at least three times in the positive results and three times in the negative results. We take this step in order to forestall the possibility of contaminating the training data for a positive instance with information about the claim itself, which would necessarily enter the public record only after the time of underwriting. In other words, we restrict the vocabulary in this way in order to focus on the generic features of applicants, rather than the specific properties of individual claims or principals.

We can then express the probability that a novel page of search results corresponds to a claim-generating bond by Bayes' theorem as follows,

$$\mathrm{P}(y=1|c_1,\ldots,c_M) = \frac{\mathrm{P}(c_1,\ldots,c_M|y=1)\ \mathrm{P}(y=1)}{\mathrm{P}(c_1,\ldots,c_M)}$$

[1] Stopping removes common words like “`is`” and “`are`”, stemming unifies tokens such as “`run`” and “`running`”, and conversion to lowercase unifies tokens like “`Business`” and “`business`”.

where we assume a multinomial Naive Bayes[2] model of the likelihood,

$$\mathrm{P}(c_1, \ldots, c_M | y = 1) = \prod_{j=1}^{M} c_j p_j,$$

with p_j being the frequency of word j in the search results conditional on a claim, or formally,

$$p_j = \frac{\left(\sum_{i=1}^{N} c_{ij} y_i \right) + \alpha}{\left(\sum_{i=1, j'=1}^{N,M} c_{ij'} y_i \right) + M\alpha},$$

where α is a pseudo-count (here we choose $\alpha = 1$).

Results

To measure the predictive strength of Webgression scores, we evaluated it on a randomly sampled test set of approximately 1,400 positive (claim-generating) instances (as many as contained in a surety bonding dataset submitted to us by a pilot client) and a matched number of randomly selected negative (non-claim-generating) instances. The performance of the classifier on the test set of a 70/30 train/test split is summarized by an ROC curve in Figure 1 below. For reference, we also include a single feature model built from a commonly-used insurance industry association code (SFAA), as well as the full-fledged claims avoidance model, consisting of a random forest classifier built from approximately 40 given and generated features.

We find that Webgression is quite comparable to SFAA code in predictive strength and in fact slightly outperforms it, with an AUC of 0.716, vs. 0.693 for SFAA code. Naturally the full claims avoidance model does best, yielding a test AUC of approximately 0.836.

[2] So-called because of the simplicity of its statistical assumptions—not for its performance, which is often quite good in spite of its methodological innocence!

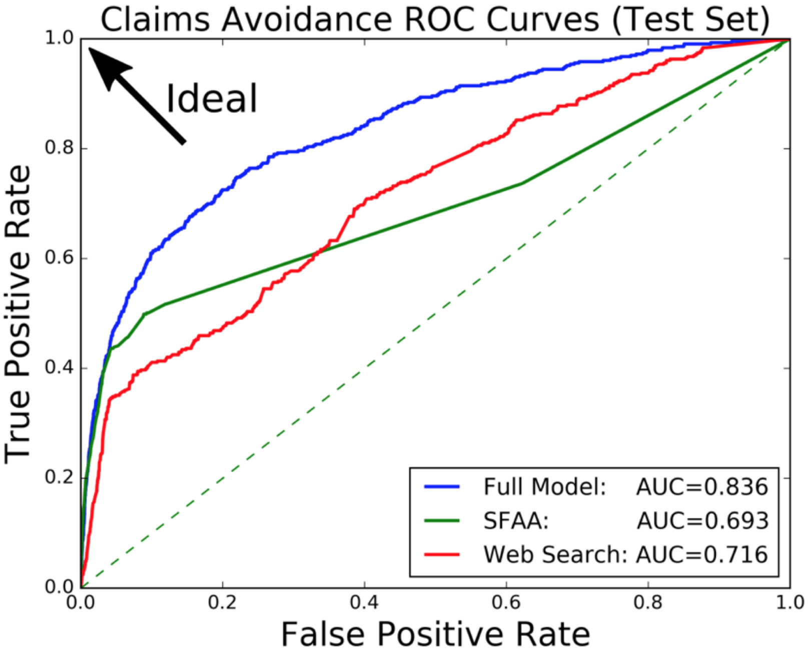

Figure 1: **Comparison of feature strength.** ROC curves for Webgression (red), SFAA code (green) and Tyche's full claims avoidance model (blue) are shown for a hold-out test set. An ROC curve summarizes the behavior of a binary classifier with a tunable threshold by displaying its false positive vs. true positive rates over all values of the threshold. A perfect classifier achieves a true positive rate of 1 at a false positive rate of 0 (corresponding to the upper left corner of the graph) whereas a perfectly random classifier achieves a true positive rate equal to its false positive rate (corresponding to the diagonal).

For the sake of example, we list the top ten terms in search results most strongly associated with claims experience in Table 1.

Token	Negative hits	Positive hits
consumer	3	48
bbb	4	50
equipment	3	32
contractor	8	82
www.bbb.org	4	35
contract	3	25
revenue	5	40
leverage	4	32
trucking	3	23
supply	3	23

Table 1: **Ten terms most strongly associated with claims**. For each token we list the number of occurrences in search results for negative instances, as well as for positive instances.

Intriguingly, at least three of the top ten terms(viz. "`consumer`", "`bbb`" and"`www.bbb.org`", the latter two counted separately by the algorithm) speak to evidence of complaints lodged against the business. Other terms such as "`contract`" and"`revenue`" perhaps paint a picture of litigiousness, and"`trucking`" and "`supply`" seem to pick out road transport. We emphasize that these terms were discovered *de novo* by the algorithm without any explicit guidance from its developers.

Conclusions and Directions for Future Work

Webgression is a simple and robust means of extracting and summarizing data from the open web in order to predict claims experience. Technically speaking, it is a form of naive Bayes classification whose output is an estimated probability generating one or more claims. As such, it can be used as a standalone claims avoidance model. In fact, as we have seen, Webgression is a more predictive feature than SFAA code itself. We consider this result quite striking in light of the fact that the SFAA code system embodies the collective intelligence of the industry in grouping similar risks together, whereas Webgression is practically industry-agnostic, “knowing” only what terms tend to appear in good and bad search results. Although Webgression scores are already proper probabilities, we note that, as with most complex modeling tasks, they are best used as one feature among many in an ensemble model.