A new (different) way of handling categorical variables in logistic regression

Abstract

By scaling categorical variables according to their proportions against the response levels in the training set and transforming them with the logistic function many difficulties of categorical variables are avoided. This is achieved by transforming the categorical variables into continuous variables. The transformation with the logistic function is a natural choice in many ways when performing logistic regression (LR) and in particular in conjunction with regularization techniques. The transformation opens up analysis without the need of choosing ‘leave out category levels’ and can be generalized to multinomial logistic regression. All this is the consequence of moving the proportion information of the categorical variable from the regression coefficients to the variable itself.

Explaining and justifying the procedure

When having many categorical covariates in LR it is an issue having to leave a category out in each covariate in order to avoid linear dependence in the regression matrix. True, in LR what is optimized is the likelihood function but a common numerical solution uses the regression matrix. Another problem is to assess the relative importance of categories depending on assigned observations. Some categories might have a lot of observations while others only a few. The problem becomes even more apparent when the logistic regression only has categorical variables. A solution that resolves all those problems is:

Consider for each category in each covariate the relative proportions to the response levels (assume for simplicity a binary response). This proportion is the best prediction probability for that specific category. So having a test example with that category set (for that covariate) immediately leads to the best prediction model simply by setting the response to the one with highest proportional probability. Here it is apparent that leaving a category out from a covariate can be a loss (at least for predictions). Say for a two category categorical variable one category level has largest proportion for ‘success’ and the other category level for ‘failure’. Both will be needed in general since leaving one out will ‘miss the opportunity’ of finding the best predictions for the related observations.

For more covariates and the rest of the category levels the same applies. The problem is then to find a formulation of the regression that can handle all the cases and that can sort out the cases which gives conflicting answers in the training set (to the response) in order to save the most informative for predictions.

When calculating the proportions one will use the training set, even for the test observations. This is the same as implementing ordinary LR with categorical variables since then the proportions are built into the calculated coefficients (from the training set). We will do this work *for* the regression. Now the other problem of having few examples in the training set (this is more apparent for many level interactions) or even no examples at all is overcome by a logistic transformation of the proportions after a centered scaling. The centered scaling is motivated by the fact that the proportions cannot have the values 0 or 1 in the logistic transformation since they translate to infinities and also the fact that a few training examples should, intuitively, have less importance than the same probability having many training examples. Say one proportion is 3/3=1 then a scaling towards 0.5 by max(3/3-1/(2*3), 0.5)=0.83 is reasonable to make. So for say 30/30=1 we get 1-1/60=0.98 for the centered proportion (max(m/n-1/2m) n=number of observations in category group and m is the number of observation belonging to a particular response. If the proportion is less than 0.5 it scales upwards by min(m/n+1/2m, 0.5)).

Why centered? For observations that end up at 0.5 after centering the logistic transform log(p/(1-p)) will be zero and the covariate will have no effect for that training example! Even the cases where the interaction (or category group) has no training examples can be handled easily (and justly!) simply by setting the transformed proportion to zero. Now the categorical levels of the categorical covariate has direct meaning in terms of that positive means promoting ‘success or 1’ as response outcome and negative the other way around.

This way all the categorical variables can be represented fully without having to leave a category level out and with their relative ‘commonness’ represented. Now if one has 5 categorical covariates, with say 3,2,8,5 levels of categories each, the above scheme will generate for the non-interacting case 3+2+8+5=18 separate transformed covariates and for the full all-interactions case 3*2*8*5=240 covariates. In this example (in analysis) I used only one of the response levels for the proportions (at a time) and they gave the same result, although both could be used at the same time in principle (in multinomial LR several response categories should be used and maybe n-1 is optimal? n=levels).

Although the size of the model grows exponentially at the same time the proportion group size shrinks exponentially effectively forcing a maximum interaction level (in my example that level seemed to be around 2 maybe at most 3). Using regularization such as the Lasso works well as far as I have tested. One added benefit of the logistic transform (other than the natural interpretation in logistic regression) is that the covariates will approximately scale as [-1,1] which is ideal in regularization.

As compared to the random forests approach in finding the best predictive model in pure categorical models the above need no parameter settings other than the choice of the categorical covariates and the levels of interactions. There is no difference between covariates having many or few category levels either in the sense that there is no bias towards covariates with many category. As compared to the ordinary case there are No worries about which category to leave out or even which categorical covariates to use, no intercept problems!!

What has really been done? Instead of having the information on the proportional frequencies of a category built into the regression coefficient it is transformed into the covariate itself . Doing so many of the problems of numerical nature due to similar numerical algorithms as in linear regression is overcome.

Are there any drawbacks? The need to transform the covariates by looking at the training data is one. This require programming. The initial choice in the covariates, their constitution and implementation in the form of interactions is a design issue. The amount of training data is certainly an issue in this analysis when considering many level interactions. But this is regular analysis issues anyway. In this case the problem of observation bias is another. Say the response is biased as 70% failure and 30% success. The ‘null model’ without any information would be to assert failure and get 70% prediction accuracy (remember that 50% is the random model). In general to get better accuracy one will have to find categorical partitions capturing the other 30% cases and this requires more specialized information found probably in different interactions (between categorical variables). So producing more categories (even artificial) should not affect the predictions and that is exactly what this method avoids. The main problem is to avoid selecting wrong categories to avoid introducing variance in the model. This is not entirely removed by this method but certainly it has much less effect. It is possible to over fit a model by choosing way too many interactions (as usual). So, yes, the problems are there but, yes they are handled!

I have read some posts on how to handle categorical covariates in Cox Regression and this method would be my advice to consider. I took some advice in some other post in looking at Kaggle competitions and this idea was worked out in the attempt to solve the ‘Titanic’ problem (I cannot however assess the quality of the result there since the answer to that competition implicitly can be found on the net and some have as high as 99%+ of accuracy which I doubt can be automatically found by an algorithm (there is a big gap from 82%, this method I think can get at least up to 85% maybe with work 90%…)

Having said all this, I am aware that especially those remarks I made that I have not checked can be the ‘turning point’ of my conclusions. Mixing the transformed categorical covariates with other continuous covariates is perhaps not as obvious at all since in the ordinary case the information of the continuous covariate is combined into the regression coefficients. However the benefit of resolving many technical analysis problems (and thereby automating the process) I believe is well worth the effort (applications to linear regression?).

Sincerely by

Jesse Burström