Generalized Reduced Error Logistic Regression Machine

Daniel M. Rice
Rice Analytics, 10805 Sunset Office Drive, Suite 300, St. Louis, MO, 63127

Abstract

Reduced Error Logistic Regression (RELR) is a new form of regression that significantly reduces error in logit coefficients and outcome predictions. RELR leads to reliable variable shortlisting and stable parameter reduction processes that overcome the dual curses of dimensionality and complexity. RELR is fundamentally different from arbitrary regression coefficient smoothing methods like Lasso and Penalized Logistic Regression, as it models non-arbitrary error estimates. This paper will review this new supervised learning machine and show that it can perform decisively better than standard methods including Penalized Logistic Regression. RELR does not use validation sample information in training, so its reduced error compared to Penalized Logistic Regression is especially significant.

Key Words: Reduced Error Logistic Regression, RELR, Data Mining, Machine Learning, Penalized Logistic Regression, High Dimensional Data.

1. Statistical Formulation

Given the equivalence between the maximum entropy and maximum likelihood solutions in logistic regression, RELR can be understood as resulting from the following maximum entropy formulation related to that of Golan, Judge and Perloff (1996). The one fundamental departure from Golan et al. (1996) is that RELR contains constraints and parameters to model error probability in a way consistent with the Extreme Value Type I error in logistic regression (Luce and Suppes, 1965; McFadden, 1974). That is, we seek to maximize:

$$(1) \qquad H(p,w) = -\sum_{i=1}^{N}\sum_{j=1}^{C} p_{ij}\ln(p_{ij}) - \sum_{l=1}^{2}\sum_{r=1}^{M}\sum_{j=1}^{C} w_{jlr}\ln(w_{jlr})$$

subject to constraints that include:

$$(2) \quad \sum_{i=1}^{N}\sum_{j=1}^{C}(x_{ijr}\,y_{ij}) = \sum_{i=1}^{N}\sum_{j=1}^{C}(x_{ijr}\,p_{ij}) + (u_r\,w_{j1r} - u_r\,w_{j2r}) \quad \text{for } r = 1 \text{ to } M,$$

$$(3) \quad \sum_{j=1}^{C} p_{ij} = 1 \text{ for } i = 1 \text{ to } N,$$

$$(4) \quad \sum_{j=1}^{C} w_{jlr} = 1 \text{ for } l = 1 \text{ to } 2 \text{ and } r = 1 \text{ to } M,$$

$$(5) \quad \sum_{i=1}^{N} y_{ij} = \sum_{i=1}^{N} p_{ij} \quad \text{for } j = 1 \text{ to } C\text{-}1,$$

where C is the number of outcome or choice alternatives, N is the number of observations and M is the number of data moment constraints. In this formulation, y_{ij} =1 if the ith observation yields an outcome/choice that is the jth possible alternative and 0 otherwise. Also, x_{ijr} is the rth characteristic or attribute associated with the ith observation and the jth alternative, so $\mathbf{x}_{ij}$ is a vector of attributes specific to the jth alternative associated with the ith observation and $\mathbf{x}_i$ is a vector of characteristics of the ith observation. In addition to representing non-interactive features of the observation or the choice/outcome, an individual $\boldsymbol{x}_r$ vector also may represent an interaction where each interaction vector is formed in the usual way as a product of characteristics and/or attributes. The p_{ij} term represents the probability that the ith observation yields the jth alternative as an outcome/choice and w_{jlr} represents the probability of error across observations corresponding to the jth alternative and rth moment and lth sign condition. When l=1, w_{jlr} represents the probability of positive error. When l=2, w_{jlr} represents the probability of negative error.

The u_r term is a measure that estimates the expected extreme error for the rth moment. The fact that the error term is an extreme error value is consistent with Extreme Value Theory. It is defined as:

(5a) $u_r = \Omega \sqrt{1 - r_r^2} / (r_r \sqrt{N_r' - 2})$ for r=1 to *M*, where Ω is defined as:

(5b) $\Omega = 2 \sum_{r=1}^{M} \sqrt{1 - r_r^2} / (|r_r| \sqrt{N_r' - 2})$ for r=1 to *M*, and where $N'_r > 2$, $-1 < r_r < 1$, and where $r_r \neq 0$.

The part of Equation (5a) that is multiplied by Ω is analogous to the inverse of a t-value that arises from the standard t-test that is performed to determine whether a correlation with the target variable is significantly different from zero across the number of non-missing and independent observations in an independent variable. When this t-value is small, then this extreme error value estimate defined by u_r is large. When this t-value is relatively large, then this u_r extreme error value estimate is relatively small. N_r reflects the number of non-missing observations across all x_{ir}, i=1 to N observations whether or not these are independent observations, whereas N'_r is a count of the number of such non-missing independent observations for each of the r=1 to M moments. Unless the observational training design is a repeated measures or multi-level design or a similar design that involves non-independent measures, then $N_r = N'_r$. With a nominal or binary dependent variable, r_r is analogous to the Pearson Product Moment Correlation between the N'_r non-missing independent values of the rth moment and the corresponding binary coded representation of reference vs. non-reference membership of the target variable, where this binary representation is 1 if the class is the reference class and 0 otherwise. With ordinal logistic regression, r_r is analogous to the Pearson Product Moment Correlation between the N'_r non-missing independent values of the rth moment and the corresponding values of the ordinal dependent variable. Ω is a positively valued scale factor that is the sum of the magnitude of the denominator of these t-values across all r moments. This sum is multiplied by 2 because there are two t-values with identical magnitudes but opposite signs proportional to the inverse of positive and negative expected extreme error for each moment. Note that we are not using t-values in a truly inferential manner in the RELR formulation, but are instead using them as an estimate that is inversely proportional to the expected extreme error value corresponding to each moment.

The constraints given as Equation (5) are intercept constraints and they can determine the threshold level for the output decisions. These intercept constraints are presented for sake of generality, but in RELR applications these constraints may be dropped from the model. This is because RELR does not attempt to reduce error in intercept weights, as unlike the input moment constraints, there are no error probability terms w_{jlr} corresponding to these intercept moments, so the inclusion of such intercept constraints may increase multicollinearity error. For example, in applications involving a binary target variable, the one intercept constraint may be dropped from the model and instead we use an alternative manner to arrive at the threshold level for output decisions. As examples of alternatives, the threshold level for each class may be determined based upon probability thresholds that yield minimal expected response bias, or such threshold levels may be determined by the user based upon the relative cost of a wrong decision in one class or another.

In many RELR applications in the case of a nominal target with more than two alternatives C, it is useful to choose the reference choice condition $j=C$ to reflect the category with the largest number of responses. This ensures that the t-value described above is the most stable possible value. The reference choice does not matter with binary targets.

1.1 Moment Definitions

The first $M/2$ set of data moments in Equation (2) are from r=1 to $M/2$ and reflect the linear or cubic components that are expected to have the largest magnitude logit coefficients as given in a full model or as defined below through variable shortlisting. The second $M/2$ set of data moments are from r=M/2+1 to M and reflect the quadratic or quartic components that are expected to have the largest magnitude logit coefficients as defined similarly. When only linear components are requested, all nonlinear variables can be are excluded. Nominal input variables are recoded into binary variables for each category condition to be accepted as inputs within this structure. In addition, input variables that are perfectly correlated with a previously admitted input variable are not allowed as input variables. Input variables with zero or a perfect positive or negative correlation to the target variable are also not allowed as input variables, but the problem would be solved if there were a perfect correlation to the target variable.

1.1.1 *linear constraints:* The linear constraints are formed from the original input variables. These constraints are standardized so that each vector $\mathbf{x}_r$ has a mean of 0 and a standard deviation of 1 across all C choice conditions and N observations that gave appropriate non-missing data. When missing data are present, imputation is performed after this standardization by setting all missing data to 0. Interaction input variables are formed by multiplying these standardized variables together to produce the desired interactions. When such interactions are formed, the vector $\mathbf{x}_r$ corresponding to each resulting interaction variable is also standardized and imputed in the same way. Finally, in order to model missing vs. non-missing patterns of observations in input variables that are correlated to the target variable,

new input variables are also formed that are dummy coded to reflect whether observations were missing or not for each previously defined linear component. Through these missing status code variables, structural relationships between missing data and the target variable can also be modeled for each of the components. These missing code variables are also standardized to a mean of 0 and a standard deviation of 1.

1.1.2 *quadratic, cubic, and quartic constraints:* These constraints are formed by taking elements in each standardized vector $\mathbf{x}_r$ described in 1.1.1 to the 2nd, 3rd, and 4th power with the exception of the missing code variables. If the original input variable that formed the linear variable was a binary variable, these components will not be independent from linear components and are dropped. When missing data are present, imputation is performed as in 1.1.1.

1.2 Symmetrical Error Probability Constraints on Cross Product Sums

With these moment definitions, two additional sets of linear constraints are now imposed in this maximum entropy formulation. These are constraints on **w**:

$$(6) \qquad \sum_{j=1}^{C} \sum_{r=1}^{M} s_r w_{j1r} - \sum_{r=1}^{M} s_r w_{j2r} = 0$$

$$(7) \qquad \sum_{j=1}^{C} \sum_{r=1}^{M} w_{j1r} - \sum_{r=1}^{M} w_{j2r} = 0$$

where s_r is equal to 1 for the linear and cubic group of data constraints and -1 for the quadratic and quartic group of data constraints. Equation (6) forces the sum of the probabilities of error across the linear and cubic components to equal the sum of the probabilities of error across all the quadratic and quartic components. Equation (6) groups together the linear and cubic constraints that tend to correlate and matches them to quadratic and quartic components in likelihood of error. Equation (6) is akin to assuming that there is no inherent bias in the likelihood of error in the linear and cubic components vs. the quadratic and quartic components. Equation (7) forces the sum of the probabilities of positive error across all *M* moments to equal the sum of the probabilities of negative error across these same moments.

Practical experience suggests that without constraints given as Equation (6) and (7), there can be significant bias in error across these components. Such bias will often manifest as spurious, larger magnitude β coefficients for nonlinear variables that do not replicate with an independent sample of observations, so these symmetrical error probability constraints are absolutely essential to obtain relatively reliable regression coefficients across independent samples.

1.3 Form of Solutions

The probability components that are of interest in the solutions have the form:

8) $p_{ij} = \exp(\beta_{j0} + \sum_{r=1}^{M} \beta_{jr} x_{ijr}) / (1 + \sum_{j=1}^{C-1} \exp(\beta_{j0} + \sum_{r=1}^{M} \beta_{jr} x_{ijr}))$ for *i*=1 to *N* and *j*=1 to *C*,

9) $w_{j1r} = \exp(\beta_{jr} u_r + \lambda_j + s_r \tau_j) / (1 + \sum_{j=1}^{C-1} \exp(\beta_{jr} u_r + \lambda_j + s_r \tau_j))$ for *r*=1 to *M* and j=1 to *C*-1,

10) $w_{j2r} = \exp(-\beta_{jr} u_r - \lambda_j - s_r \tau_j) / (1 + \sum_{j=1}^{C-1} \exp(-\beta_{jr} u_r - \lambda_j - s_r \tau_j))$ for *r*=1 to *M* and j=1 to *C*-1.

However, for the reference condition where *j*=*C*, the solutions have the normalization conditions described by Golan et al. (1996) where in this case the vectors $\boldsymbol{\beta}_c$ and $\boldsymbol{\lambda}_c$ and $\boldsymbol{\tau}_c$ are zero for all elements *r*=1 to *M*. Hence, the solution at *j*=*C* has the numerator equal to 1 in Equations (8), (9) and (10). Also, in the case of ordinal regression, the logit coefficients β_r and λ and τ corresponding to each of the *r*=1 to *M* moments and two symmetrical error probability constraints only exist at *j*=1 due to the linear ordering assumptions.

1.4 Full RELR Models and Variable Shortlisting

With high dimensional data, the purpose of variable shortlisting is to reduce the dimensionality of the problem into a manageable number of variables. The basic idea is to capture the subset of variables that would have the largest magnitude logit coefficients in the full model, while excluding those variables with smaller magnitude logit coefficients. In this way, we can build a model with a much smaller set of variables than with the entire set and achieve a model that still includes those variables with the largest weight in the solution if all variables had been included. This is important, as it is very easy to get models with tens of thousands of variables when higher level interactions are

included, even when the original number of variables was less than 100. This variable shortlisting process arises from a set of relationships in the preceding equations. For example, from Equations (9) and (10) and the corresponding equations for the reference condition with the numerator set to 1, we have:

11) $\ln(w_{j1r}/w_{j2r}) - \ln(w_{C1r}/w_{C2r}) = 2\beta_{jr}u_r + 2\lambda_j + 2s_r\tau_j$ for j=*C-1* and r=1 to M.

Equation (11) reflects the log of the probability ratio of the positive and negative error for the rth moment and jth outcome relative to this same ratio for the reference condition. The right hand side of Equation (11) is interpreted as the difference in positive and negative error for the rth moment and jth outcome relative to the reference condition. This quantity would estimate aggregate utility error if these were choice data, as unlike utility it reflects an aggregate across observations for each moment. This error ε_{jr} is directly related to the extreme expected moment error u_r:

12) $\varepsilon_{jr} = 2\beta_{jr}u_r + 2\lambda_j + 2s_r\tau_j$ for j=*C-1* and r=1 to M

and is assumed to be Extreme Value Type I distributed. Writing u_r in terms of Ω/t_r and rearranging terms gives:

13) $(t_r/\Omega)(\varepsilon_{jr}/2 - \lambda_j - s_r\tau_j) = \beta_{jr}$.

Therefore, we know that the following relationship will hold when each ε_{jr} can be assumed to have a negligible value:

14) $(t_r)(-\lambda_j - s_r\tau_j)/\Omega \approx \beta_{jr}$ for j=*C-1* and r=1 to M.

Hence, we know that the value of each logit coefficient β_{jr} will be approximately proportional to t_r across all linear and cubic moments, but with a different proportionality across all quadratic and quartic moments, when in all cases ε_{jr} is close to zero. This is because we also know that the expression $-\lambda_j - s_r\tau_j$ corresponding to all linear and cubic variables will be equal across all r=1 to M/2 components and that corresponding to all quadratic and quartic variables also will be equal across all r=M/2+1 to M moments for each of the jth conditions. This follows from the definition of s_r in section 1.2. Therefore, we use this relationship to select the linear and/or cubic moments with the largest expected logit coefficient magnitudes simply in terms of the magnitude of t_r. Likewise, we select the largest expected logit coefficient magnitudes for all quadratic and/or quartic moments simply in terms of this same magnitude.

In fact, we expect that this relationship in Equation (14) to be a very good approximation when Ω gets large in comparison to the magnitude of t_r. This is due to the linear relationship in Equation (2) which suggests that if we substitute $u_r = \Omega/t_r$, then as Ω gets larger in comparison to the magnitude of t_r, w_{j1r} and w_{j2r} must become closer to being equal. At very large values of $\Omega/|t_r|$, w_{j1r} and w_{j2r} would be substantially equal, but this would require that ε_{jr} is also close to zero which is exactly what we assume. Then again, this relationship in Equation (14) could be a poor approximation for datasets which have few variables with significantly large t-value magnitudes. This is because there would be small values of the ratio $\Omega/|t_r|$ in which case the error ε_{jr} is not necessarily close to zero. However, in such datasets, the variables with the strongest relationship to the target variable would not be an issue as these are the few variables with significantly large t-value magnitudes. If these were causal variables, this would be akin to assuming that cause is not possible without some degree of correlation. Therefore, in general, we use the magnitude of t_r to select the variables with the largest expected logit coefficient magnitudes whether or not the actual estimated error ε_{jr} turns out to be close to zero. With many variables with relatively large t-values, empirical results do consistently verify this reliably strong relationship between t_r and β_{jr} that allows shortlisting to get highest magnitude logit coefficients simply on the basis of t_r magnitudes. Yet, because this correspondence between t_r and β_{jr} tends to deviate from exact proportionality more and more as there are fewer variables with relatively large magnitude t-values, it is not seen in Parsed RELR solutions that result from parameter deletion. At the end, we can still miss less important variables that fail to make it on the shortlist. This is a small price for shaking the wrath of this curse of dimensionality.

RELR's lack of rigid proportionality between t_r and β_{jr} differentiates it from Naïve Bayes. For example, given standardized variables with no missing data, the Gaussian Naïve Bayes formulation (Mitchell, 2005) is rigidly tied to exact equality between a measure of t_r and β_{jr} that does not allow for interactions. In contrast, RELR is well suited for high dimensional data and interactions, as RELR's t-value screening is easily batched in serial or parallel.

Full RELR models will be useful by themselves in some applications if a user is only concerned with a good fit and not concerned with the number of parameters or the meaning of parameters in the model. In cases where a user is also not

interested in a full model across a set number of variables, we build successively smaller Full RELR models by successively dropping variables based upon t-values and then choose the model that has the maximal log likelihood value across the training observations. Practical experience has found that Full RELR models can overfit slightly more if there are too many error probability terms in the model and not enough observations. For example, with binary dependent variables, we have sometimes seen an uptick in overfitting when we included more variables than roughly half the smaller of the target vs. non-target observation count. This can be controlled by limiting the starting number of variables in the shortlist, but this limitation is only practically relevant with an extremely small sample size.

1.5 Parsed RELR Models through Parameter Reduction

In many applications, we would like a model that fits the data reasonably well with a reliable and small set of most important parameters. The goal is to arrive at the same small set of parameters that would also be selected with a completely independent sample. This selection of these most important parameters should not be biased by univariate effects related to the magnitude of t values, as there are many more interaction variables, so if there is a bias toward univariate effects, there will likely be many more spurious interactions. We employ a backward selection process that drops the least important parameter at each step until we arrive at a solution that meets our objective with the smallest number of parameters that yield a reasonably good fit in the training sample. Like Full RELR, Parsed RELR does not use any validation sample information to arrive at its solution.

2. RELR Evaluation with Comparison of Results to Standard Methods

Data were obtained from the Pew Research Center's 2004 Election Weekend survey using observations that indicated Bush or Kerry as their vote. In a first "smaller training sample" model, the training sample consisted of a segmented sample of 8% or 188 observations. The remainder of the overall sample defined the validation sample. In a second "larger training sample" model, the training sample consisted of a segmented sample of roughly 50% or 1180; the remainder of this sample defined the validation sample. This segmented sampling method is the default sampling method in the SAS Enterprise Miner 5.2 software that was also used to build the models. The target variable was Presidential Election Choice (Bush vs. Kerry). Kerry was the target condition. The 2004 election was very close, as roughly 50% of the respondents opted for Kerry in all of these sub-samples. Hence, these are extremely balanced samples. Target condition response numbers are indicated in Figures 1 and 2.

There were 11 interval variables and 44 nominal input variables originally, but the nominal variables were recoded into binary variables for input into RELR. RELR also produced new input variables from these original input variables corresponding to two-way interactions, polynomial terms, and missing data. Over 2500 variables resulted in total. Both Full and Parsed models were run. Full RELR employs a backward selection process based upon the magnitude of the t values. Full RELR chose as its best model the model corresponding to the variable set associated with the maximal log likelihood across the training observations. 176 variables were selected in the Full RELR best larger training sample model, whereas 24 variables were selected in the Full RELR best smaller training sample model. These identical variables were input into Penalized Logistic Regression (PLR) to have an "apples-to-apples" comparison to Full RELR. For this same "apples-to-apples" reason, the same intercept estimation procedure was used with RELR models and PLR- including the use of thresholds that minimized bias in classification with corresponding intercept correction. The λ values in these PLR models were systematically varied between .5 and 150; values above 150 had floating point error. In both samples, the λ=1 was either associated with the best validation misclassification rate as in the small sample, or very close to the best, as a λ of 150 was slightly better in the larger sample. However, we report the results based upon λ=1 for both samples here because this is in the range that is normally employed for λ for PLR.

Bush vs. Kerry models were also run within Enterprise Miner 5.2 using the Support Vector Machine, Partial Least Squares, Decision Tree, Logistic Regression and Neural Network methods. SVM and PLS were beta versions in EM 5.2; reported error measures were computed independently from EM for these methods and RELR. Defaults were employed in all cases except Logistic Regression where two way interactions and polynomial terms up to the 3rd degree were specified and Support Vector Machines where the polynomial kernel function was requested. Also, Stepwise Selection was performed with the Logistic Regression in the Pew "smaller sample", but no selection was performed with the larger sample due to the long time that it took to run the stepwise process in this sample. In addition, the Imputation Node within Enterprise Miner was employed with the Regression and Neural Network methods and was run using its default parameters. The identical variables and samples were employed as inputs in all cases. Like most political polling datasets, there was a wide range in the correlation between the original input variables that went from roughly -.6 to about 0.81. These correlation magnitudes were over .9 for many of the interactions and nonlinear

variables produced by RELR, so this dataset clearly exhibited multicollinearity. In addition, there was significant correlation to the target variable in a number of variables; the largest correlations were in the .7-.8 range.

Figure 1: "Smaller Training Sample" Models.

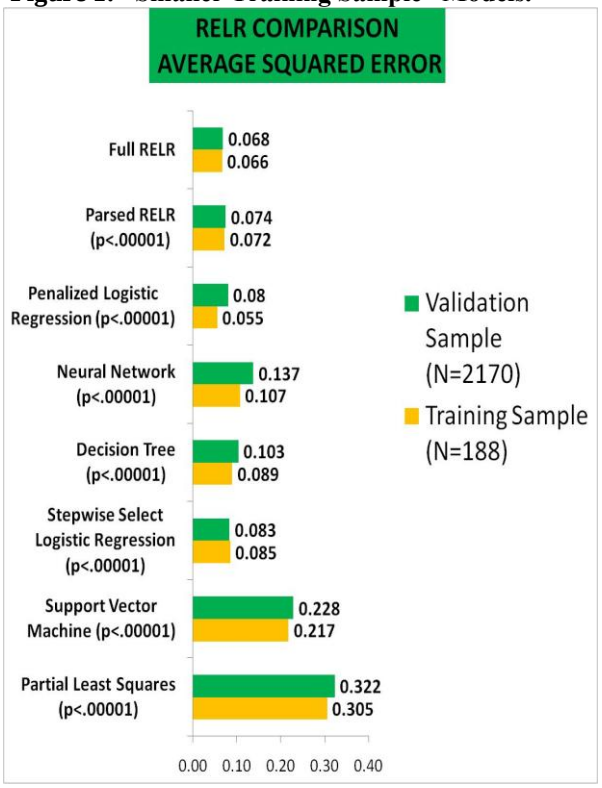
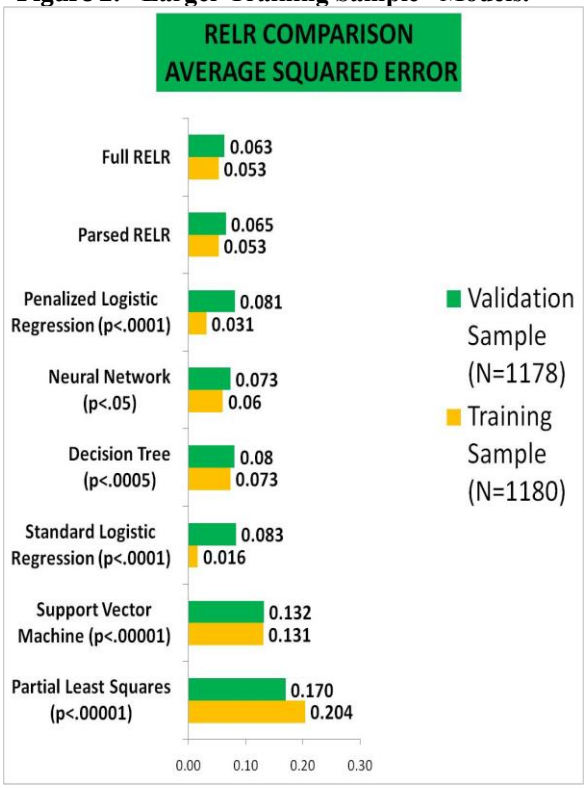


Figure 2: "Larger Training Sample" Models.



Figures 1 and 2 show the average squared error from models based upon the "smaller training sample" and "larger training sample" datasets. The probability levels compared the significance of the validation sample error to that of Full RELR using a one-tailed paired t-test. Clearly, Full RELR had significantly reduced error compared to all methods, except Parsed RELR in the larger training sample. Overfitting probability is not shown in the figures, but slight overfitting (p<.05) was seen in the larger training sample condition for Full and Parsed RELR. Overfitting was also observed in all other methods except Decision Tree in the larger training sample which still had a non-significant trend, SVM which had almost no trend in either training sample condition, and Partial Least Squares. Overfitting was most pronounced with Penalized Logistic Regression in both conditions.

Table 1: Misclassification Rate from Smaller (left) and Larger (right) Training Sample Conditions

Method	*Training*	*Validation*	*Method*	*Training*	*Validation*
Full RELR	0.122	0.106	Full RELR	0.074	0.089
Parsed RELR (p<.05)	0.138	0.115	Parsed RELR	0.094	0.097
Penalized Logistic Regression	0.117	0.104	PLR	0.049	0.099
Neural Network (p<.001)	0.122	0.135	NN	0.084	0.103
Decision Tree (p<.001)	0.149	0.146	DT	0.122	0.106
Stepwise Logistic Regression (p<.001)	0.100	0.150	Standard LR	0.024	0.097
Partial Least Squares (p<.0001)	0.329	0.373	PLS (p<.001)	0.206	0.179
Support Vector Machine (p<.0001)	0.324	0.376	SVM	0.109	0.098

Table 1 shows the rate of misclassification error across these methods. The probability levels reflect the differences in the validation misclassification rates in each method compared to Full RELR's; these p values are from McNemar's test for dependent proportions. In the smaller training sample, Full RELR had better validation sample classification accuracy than all other methods except Penalized Logistic Regression. However, there was no significant advantage to Full RELR in the larger sample, except in comparison to Partial Least Squares. Overfitting, was not significant with either Full or Parsed RELR in either sample – although there was a strong non-significant trend (p<.10) for Full RELR

to overfit in the larger training sample model. On the other hand, Penalized Logistic Regression had significant overfitting in the larger training sample; this also would have been observed with a value of λ=150 in the larger training sample. Significant overfitting was also observed in a number of the other methods as is obvious in this table.

Table 2 compares the top seven logit coefficients in magnitude from Full RELR and Penalized Logistic Regression in the larger training sample models. Note that with standardized variables, nonlinear interactions are possible and interpretable with binary variables, as they measure patterns of differential effects across the subgroups involved in the interaction. All seven RELR logit coefficients were highly significant with a Wald Chi-Square statistic, whereas PLR coefficients had much greater error such that most of these effects were at best marginally significant and some (NOTPARTYLINExDEMOCRAT and NOTPARTYLINE) were not significant. The ability of Full RELR to return logit coefficients with much smaller error was also seen in the correlations between training and validation logit coefficients across all 176 variables (r=.976) for Full RELR vs. (r=.238) for PLR.

Table 2: Full RELR (left) and PLR (right) Largest Magnitude Logit Coefficients from Larger Training Sample Model

Parameter	*β*	*Stderr*	*Parameter*	*β*	*Stderr*
IRAQWRONG	*0.36*	0.05	(REPUBLICANxREGISTERED)^3	-7.40	4.44
LOSINGWARONTERROR	0.27	0.03	(DEMOCRATxPARENT)^3	-6.13	2.87
LOWRISKINCHANGE	0.26	0.03	(REPLUBICANxBORNAGAIN)^2	3.82	2.01
REPUBLICAN	-0.25	0.03	(REGISTEREDxIRAQWRONG)^3	3.63	1.93
BUSHWILLLOSE	0.24	0.03	NOTPARTYLINExDEMOCRAT	-3.21	4.02
REGISTEREDxIRAQRIGHT	-0.24	0.03	NOTPARTYLINE	-3.21	4.01
DEMOCRAT	0.22	0.03	(RISKINCHANGExGOCHURCH)^3	-3.17	1.15

Table 3 compares the Parsed RELR solution obtained with the larger sample training condition (n=1178) to that obtained in the corresponding independent validation sample (n=1180). The identical 9 variables and intercept were selected in each case and had highly correlated logit coefficients (r=.99). These same models were selected in both samples even when three way interactions were included. Yet, when three way interactions were included, there were several other variables that had univariate t-values that were greater in magnitude than the t-values for these variables in one sample or the other. However, these other variables were likely spurious, as their t-values did not rank reliably high in magnitude across both independent samples. Parsed RELR does not use any out-of-sample information. Still it was able to avoid the selection of apparently spurious three way interactions and instead selected the same variables from both independent samples. Note that the same variables were not selected in independent samples with the smaller training sample size (n=188).

Table 3: Stability of Parsed RELR Solutions from Independent Samples (Larger Sample Training and Validation)

Parameter	*β*	*Stderr*	*Parameter*	*β*	*Stderr*
INTERCEPT	-0.040	0.000	INTERCEPT	-0.040	0.000
DEMOCRAT	1.049	0.112	DEMOCRAT	0.937	0.093
NOTPARTYLINE	-0.947	0.105	NOTPARTYLINE	-0.937	0.102
NOTPARTYLINExDEMOCRAT	-0.929	0.103	NOTPARTYLINExDEMOCRAT	-0.908	0.100
IRAQWRONG	1.653	0.176	IRAQWRONG	1.255	0.121
LOSINGWARONTERROR	1.223	0.139	LOSINGWARONTERROR	0.759	0.087
LOWRISKINCHANGE	1.281	0.136	LOWRISKINCHANGE	0.856	0.091
HIGHRISKINCHANGE	-0.978	0.114	HIGHRISKINCHANGE	-0.748	0.083
BUSHWILLLOSE	1.153	0.126	BUSHWILLLOSE	0.788	0.084
REPUBLICAN	-1.207	0.128	REPUBLICAN	-0.951	0.094

3. General Discussion

RELR has been developed as a means to model and subtract out estimates of error in logistic regression. This is fundamentally different from the arbitrary regression coefficient smoothing functions in Lasso and Penalized Logistic Regression. The RELR error probability function is not arbitrary and is instead based upon a model of extreme value error. If RELR is successful, then its models should have lower error. In fact, RELR can return models with substantially less error. At the time of the JSM Conference in August, 2008, we had not yet seen these average squared error results, so we only reported the classification results. It is now clear that the most reliable reduced error effects will be seen with a measure like average squared error. Average squared error is less dependent upon a threshold than misclassification rate. Average squared error also reflects the entire probability range. Of particular interest was that

RELR had substantially lower average squared error than Penalized Logistic Regression in the validation samples. Penalized Logistic Regression's validation sample classification performance matched Full RELR's, but it picked the λ that gave the best validation sample classification, whereas RELR does not use the validation sample to arrive at its scaling constant Ω. Because both RELR models chose intercepts that minimized bias in classification decisions, their accurate classification performance cannot be due to bias in hit vs. correct rejection accuracy. Overall, the only fit measure where both Full and Parsed RELR performed significantly less well than another method was the overfitting shown in Figure 2, as SVM did not have much trend for overfitting here and Partial Least Squares had none. However, both SVM and Partial Least Squares had extremely poor average squared error performance, so their lack of overfitting should mostly reflect the fact that they did not cover the full 0-1 range in predicted scores.

These results show that Parsed RELR can have comparable fit performance to Full RELR with a large enough training sample. This concurs with the solution stability results, as Parsed RELR did not return the same variables from independent training samples when there was a smaller training sample. With a larger training sample, Parsed RELR selected the same variables across independent samples with a similar pattern of logit coefficients as shown in Table 3. Such stability would appear to be beyond the reach of all other known methods. Another striking feature is the face validity of these Parsed RELR solutions. Party affiliation and party line voting, attitudes toward a risk in change and the war on terror and Iraq, and whether one believes that one's candidate will win were all arguably important variables in the 2004 U.S. presidential election. More complicated variables involving interactions and higher order polynomials such as in the Penalized Logistic Regression highest magnitude logit coefficients in Table 2 would not seem to have as much face validity. We want a method that will find nonlinear and interaction effects if they are actually there. RELR is well tuned to do this across very high dimensional datasets. Yet, we do not want a method that is biased to favour these effects. When bias is present, such effects may appear as apparently spurious effects across independent samples as in this Penalized Logistic Regression case. Parsed RELR does not appear to show this bias.

It is impossible to verify causality on the basis of correlation data. Yet, given a representative sample, data mining might be able to provide a small set of the most probable and important causal relationships and/or their proxies if necessary conditions are met. These include 1) independent out-of-sample solution replication to prove stability and to rule out spurious variables, 2) face validity including the logical possibility that all independent variables reflect a direct or indirect causal effect on the dependent variable, and 3) relatively accurate out-of-sample fit. These necessary conditions were met in the Parsed RELR solution of Table 3. These conditions are not by themselves sufficient for proving causality, as other solutions are possible that could meet these conditions. Also, not all important causal variables or their proxies may have been specified in the original input variable set. Nevertheless, it would appear unlikely that these three necessary conditions would happen by chance without any causal association in the data. Therefore, at the very least, a highly stable and accurate Parsed RELR solution might be useful as a starting point for experimental verification of a possible causal mechanism. Yet, in social science, medicine and business where experimentation is not always possible, a regression model that meets conditions 1, 2 and 3 might be the only available window into the most probable causal associations in the data. The most striking aspect of these results is that Parsed RELR may return solutions that meet all three conditions at a sample size within the reach of most applications.

Acknowledgements

We thank the Pew Research Center for the availability of their data for academic publications, but the Pew Research Center for the People and the Press bears no responsibility for the analyses or interpretations of the data presented here. Rice Analytics is a member of the SAS Alliance Partnership; we thank SAS Institute for this partnership.

References

Golan, A., Judge, G. and Perloff, J.M. (1996), A maximum entropy approach to recovering information from multinomial response data. *Journal of the American Statistical Association*, 91: 841-853.

Luce, R.D. and Suppes, P. (1965). Preference, utility and subjective probability, in R.D. Luce, R.R. Bush and E. Galanter (eds), Handbook of Mathematical Psychology, Vol. 3, Wiley and Sons, New York, NY, pp. 249-410.

McFadden, D. (1974). Conditional Logit Analysis of Qualitative Choice Behavior. In P. Zarembka (ed) Frontiers in Econometrics, New York, Academic Press, pp. 105-142.

Mitchell, T. (2005), Generative and discriminative classifiers: Naïve Bayes and Logistic Regression. Online draft.

Patents Pending

Generalized Reduced Error Logistic Regression Method is currently pending patent.