A Logistic Regression Model for Small Sample Classification Problems with Hidden Variables and Non-Linear Relationships: An Application in Business Analytics

Aleksandra Mojsilović

IBM T. J. Watson Research Center
Yorktown Heights, NY 10538

ABSTRACT

Logistic regression is one of the frequently used models in pattern recognition, especially in binary classification tasks. We focus on a class of small-sample classification problems where logistic regression seems to be a "natural" choice for the classifier, yet its direct application yields sub-optimal results. Specifically, we consider cases when: 1) input-output relationships are non-linear, 2) there is a need to estimate hidden states or auxiliary variables in the model, and 3) the training set is small preventing the use of more sophisticated techniques. We first describe an approach to compute the parameters of the regression, which addresses the issue of estimating hidden variables. We then describe a recursive adaptation procedure that identifies the most significant non-linear relationships in the data and adapts the model by introducing corresponding higher-order terms. The performance of the method is tested in a business modeling application, demonstrating significant improvements over the traditional classifiers.

1. INTRODUCTION

Logistic regression model is one of the most frequently used approaches in pattern classification. It is used either as a stand alone technique to model input-output relationships in a wide range of applications [1]-[3], or as an underlying kernel function in the *support vector machine classifiers* (SVM) [4],[5]. In this work we focus on a class of problems where logistic regression seems to be a "natural" model, yet its direct application yields sub-optimal results. In particular, we focus on problems where the following statements apply: 1) the input-output relationships are complex, 2) in addition to the input-output relationships there is a need to estimate hidden states or auxiliary variables in the model, and 3) the training set is very small. Problems of this type occur frequently in business analytics applications, financial forecasting, drug discovery and pharmaceutical research, where the outcome is typically driven by a complex set of often mutually related factors, and where historical examples of certain behavior needed to train the model are limited. Such problems include: 1) business process modeling and forecasting (e.g. customer targeting and estimating the likelihood of buying a new product, deciding if a company is "risky" customer, deciding weather to pursue an investment into a project, evaluating a business action of a company, etc.), 2) evaluating the quality of software engineering projects, 3) bankruptcy prediction, and 4) monitoring the effect of a combinational therapy on a patient. This work describes a new methodology for designing and training logistic regression classifiers, suitable for the problems of this kind. We also demonstrate the use of the method in a business analytics application.

The first aspect of the problem can be more formally described as "*given a set of inputs and observed outputs, estimate both the parameters of the model and several hidden states if only partial a priori information about the states is given.*" An example of such a problem is developing a dashboard to track a portfolio of large IT services customers and identifying those who might decide to terminate their contract, or renegotiate it to achieve lower price, both leading to a significant loss of revenue to the service providing company. In this case, the inputs to the model are numerous variables that describe the following five risk factors: 1) financial health of client companies, 2) previous relationships with the service provider, 3) price and competitiveness of the offered service, 4) significant events in the client company that could have a potential impact on the decision to cancel the service (e.g. change of CEO, merger, restructuring, etc.), and 5) previous history of contract terminations or renegotiations. The output variable is the likelihood that a customer will terminate its contract (or a part of it). Conventional classification methods are limited to estimating the likelihood of termination, without providing insights into which factor is most influential in the decision. Yet, knowing the impact of different factors to the client's decision can help the service providing company influence the outcome. For example, if the decision to terminate is based on the limited cash availability, the service providing company might architect different ways of financing for the customers with lower liquidity. On the other hand, if the decision is formed based on the low satisfaction with the service, the service providing company can still influence the outcome by improving the service and mobilizing its sales and marketing teams to save the relationship with the customer. These "risk factors" are typically not known *a priori*; what are known are only the variables that influence them. Therefore, these "risk factors"

can be seen as hidden states in the model. (Note an important difference between this problem and many signal processing and control problems where the objective is to estimate states or auxiliary variables in the presence of noise [2]). In traditional classification methods, such as logistic regression, after the parameters of the model have been estimated, the values of these states are computed as a bi-product of the model. However, in many applications, at least some of the relationships among these factors (i.e. hidden variables) are know. For example, in the aforementioned problem of the dashboard design, it is often possible to provide additional information in the form of "Company A has been more satisfied than Company B" or "Company C has better financial health than Company D". Since the traditional parameter estimation techniques, such as MAP or iteratively reweighted least squares, do not account for these relationships, the estimation of hidden variables obtained with the standard parameter estimation procedures is not optimal. Hence, it is of interest to develop training procedures that will capture such relationships in the data.

The second aspect of the problem can be more formally described as "*given a small set of input-output examples, estimate the parameters of a simple model, so as to capture complex non-linear input-output relationships in the data*." While conventional learning algorithms produce sufficiently accurate methods for many applications, when working with small data sets (and especially when there are non-linear relationships among the variables) they suffer from many limitations, which if overcome, could greatly improve the performance of the data classification and regression systems that employ such models. The small size of the training set severely limits the selection of the classifier to the simplest models, which typically do not account for non-linear relationships in the data, which are to be discovered in the training phase. For example, the tree-based classifiers that effectively capture complex relationships in the data [8] cannot be applied at all if the training set is small. The small size of the training set also limits the number of input variables. Increasing the number of input variables in the model increases the number of free-parameters. This results in a deteriorating performance, the "curse of dimensionality", which is due to the mismatch between the size of the training set and the number of free parameters. This can be overcome with a new class of "SVM-like" models that operate in sparsely populated feature spaces [10]. Such models rely on the observed relationships between the number of training samples *m*, number of features *k*, and the generalization error of the classifier. Namely, for many traditional classifiers trained by *m* objects, the generalization error $e(k)$ increases with the increase in feature size, and reaches the maximum at about $k = m$ (the "peaking phenomenon"). However, it has been observed that after the maximum is reached, in cases when the sample size is significantly smaller that the feature size ($m < k$), it is possible to obtain classification performances that are much better than those obtained with "sound" feature sizes [10]. However, in many applications it is not possible to select a large number of features as required by such approach. Therefore there is a need for simple models, which can be constructed from the small training samples to capture non-linear input-output relationships.

2. OVERVIEW OF THE STANDARD LOGISTIC REGRESSION MODEL

Let us consider a simple classification problem of a data labeled by a random variable, ω, which takes its values from a discrete set $\omega \in \{\omega_o, \omega_1\}$. The input data is in the form of a *L*-dimensional random vector, $\mathbf{u} = [u_1, u_2, ..., u_L]$. A natural choice for the model in this case is the logistic regression

$$y_i = P(\omega_i = \omega_1 \mid \mathbf{u}_i) = \frac{1}{1+\exp(-a_o - \mathbf{a}^T \mathbf{u}_i)} \quad (1)$$

The logistic model is also known to be more robust against violations of multivariate normality assumption than the naïve Bayesian classifiers [6]. Parameters of the model, **a**, are usually determined by maximizing the log-likelihood function

$$\log P(S \mid \mathbf{a}) = \sum_{i=1}^{N} \log P(\mathbf{u}_i, \omega_i) = \sum_{i=1}^{N} [z_i \log y_i + (1 - z_i) \log(1 - y_i)] \quad (2)$$

where $S = \{\mathbf{u}_i, \omega_i\}, i = 1, ..., N$ is the training set and z_i is an indicator random variable, which has the value one when $\omega_i = \omega_o$ and zero otherwise. There exist no closed form solution for (2) and **a** is often computed by taking the second derivative of (2), which yields the Newton-Raphson update

$$\mathbf{a}(k+1) = \mathbf{a}(k) + (\mathbf{U}^T \mathbf{V}_r \mathbf{U})^{-1} \mathbf{U}^T \mathbf{V}(k) \mathbf{z}^*(k) \quad (3)$$

where *k* is the iteration number, **U** is a matrix whose rows are the vectors $\mathbf{u}_i$, $\mathbf{V}(k)$ is a diagonal matrix whose elements are $y_i(k)(1 - y_i(k))$, and $\mathbf{z}^*(k)$ is a vector with elements $(z_i - y_i(k))/(y_i(k)(1 - y_i(k)))$. This update rule is also referred to as *Iteratively Reweighted Least Squares* (IRLS). Newton-Raphson takes this form not only for logistic regression problems, but for a family of statistical models known as *Generalized Linear Models* [7].

3. PARAMETER RE-ESTIMATION IN CASE OF PARTIALLY KNOWN HIDDEN VARIABLES

We now extend the logistic regression method to address the case of partially know hidden states. As an illustration let us use the customer targeting application described in Introduction. In building a model that estimates the likelihood of terminating a services contract, in addition to estimating the overall risk of termination, we also need to capture which one (or which combination) of the five risk factors (i.e. financial health, client satisfaction, price, significant corporate changes and history of termination) drives the overall risk. Back to our logistic model, this calls for introducing five hidden variables, $x_1 - x_5$, to measure the contribution of the five risk factors. Then, the straightforward solution to estimate $x_1 - x_5$ is to use linear combinations of explanatory variables that are related to the corresponding risk factors, as

$$y_i = \frac{1}{1+\exp(-a_o - \sum_{j=1}^{M} x_{ji})}, \; x_{ji} = \mathbf{a}_j{}^T \mathbf{u}_{ji} \quad (4)$$

where: *M* is the number of hidden variables (in our case $M = 5$), **u** is the vector of measured inputs rewritten as $\mathbf{u} = [\mathbf{u}_1 \, \mathbf{u}_2 \, ... \, \mathbf{u}_M]$ and $\mathbf{u}_i$ is a vector of inputs that contribute to the state *i*. The regression above assumes that the hidden variables (states) are entirely unknown and will be computed

as a bi-product of the model. This is an acceptable solution; however, it is not an optimal one when some information about either the value of some of the hidden states or their relationship is available. Let us assume that this information is given *a priori*, or can be entered into the model by modifying the initial estimates to reflect the known relationships. (For example, the known relationship "company k had better financial health than company l" could be enforced by increasing the estimated value $\mathbf{a}_1{}^T\mathbf{u}_{1k}$ to reflect $x_{1k} > x_{1l}$, or "company p had higher customer satisfaction than company q" could be entered into the model by increasing the value of $\mathbf{a}_2{}^T\mathbf{u}_{2p}$ to reflect $x_{2p} > x_{2q}$.) Given the matrix $X = \{x_{ji}\}$, $j = 1,...,M$, $i = 1,...,N$ whose entries are known, estimated, or modified values of the states in the model, we can rewrite the objective function as

$$\text{maximize} \sum_{i=1}^{N}[z_i \log y_i + (1-z_i)\log(1-y_i)] \qquad (5)$$

$$\text{subject to} \sum_{j=1}^{M}\sum_{i=1}^{N}(x_{ji} - \mathbf{a}_j{}^T\mathbf{u}_{ji})^2 \le \varepsilon \qquad (6)$$

where ε is a predefined threshold. Solving (5)-(6) using Lagrange multipliers gives a new update rule for $\mathbf{a}$:

$$\begin{bmatrix}\mathbf{a}(k+1)\\ \lambda(k+1)\end{bmatrix} = \begin{bmatrix}\mathbf{a}(k)\\ \lambda(k)\end{bmatrix} + \mu\begin{bmatrix}\sum_{i=1}^{N}(z_i - y_i)\mathbf{u}_i - 2\lambda(k)\sum_{j=1}^{M}\sum_{i=1}^{N}(x_{ji} - \mathbf{a}(k)^T\mathbf{v}_{ji})\mathbf{v}_{ji}\\ \sum_{j=1}^{M}\sum_{i=1}^{N}(x_{ji} - \mathbf{a}(k)^T\mathbf{v}_{ji})^2 - \varepsilon\end{bmatrix} \qquad (7)$$

where λ is the Lagrange multiplier, μ is the step size, $\mathbf{u}_i = [\mathbf{u}_{1i}\ \mathbf{u}_{2i}\ ...\ \mathbf{u}_{Mi}]$ is $1\times L$ vector of inputs, $\mathbf{v}_{ji}$ is a $1\times L$ vector defined as $\mathbf{v}_{ji} = [\mathbf{o}_{1i}\ \mathbf{o}_{2i}\ ...\ \mathbf{u}_{ji}\ ...\ \mathbf{o}_{Mi}]$, and $\mathbf{o}_{ki}$ is a zero vector of the same length as the sub-vector $\mathbf{u}_{ki}$.

4. MODEL ADAPTATION TO CAPTURE NON-LINEAR RELATIONSHIPS

Note that in general, equation (4) assumes that any input variable can contribute to any state. Eq. (4) can be also generalized to account for quadratic terms, higher-order nonlinearities or any other relationship $x = f(\mathbf{a},\mathbf{u})$. In many applications the performance of the model depends entirely on the ability of the classifier to capture such relationships. For example, in the customer targeting application described above, variables related to customer satisfaction and business efficiency are much more important to the clients who are paying high price for the service than to the clients who have negotiated lower price. The later are less likely to terminate their contract unless something happens to their business, implying that in such examples, variables related to financial health are far more important. Parametric models such as (1) fail to adequately approximate these relationships, unless the relevant quadratic terms are already included into the training set, or a non-linear form of (4) is used. An alternative approach is to use treed models, consider a partition of the data and then fit a separate logistic model within each subset of the partition [8]. This is illustrated in Fig. 1. In practice, when dealing with small training sets, the small number of samples does not allow for any partitioning of the data, thereby severely limiting the model selection to the simplest structures with few explanatory variables.

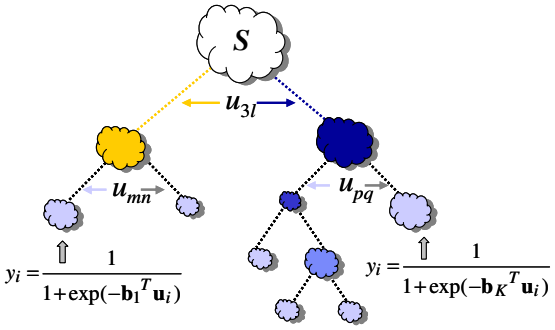

Fig.1: Traditional tree-based regression.

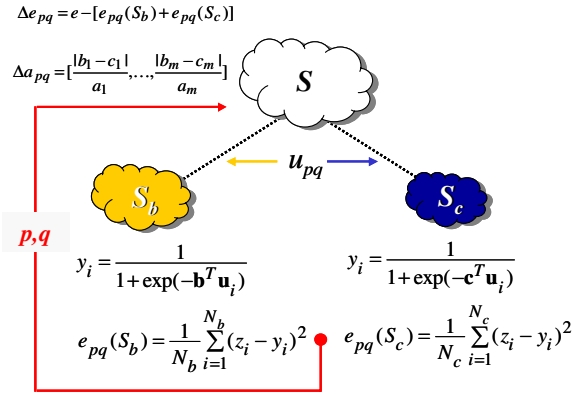

Fig. 2: Model adaptation used to identify significant non-linear relationships in the data.

To construct a logistic model that will capture at least some of the complex input-output relationships, we propose a recursive approach to estimate significant nonlinear relationships and include them in the set of explanatory variables $\mathbf{u}$. We will use a modification of the tree-structured approach. We start by estimating the parameters, $\mathbf{a}$, of logistic model (1) via (7), and compute the classification error as:

$$e = \frac{1}{N}\sum_{i=1}^{N}(z_i - \hat{z}_i)^2 \qquad (8)$$

In each iteration, we will use a different input variable, u_{pq}, to split the data into two subsets, S_a and S_b, and fit a separate regression model in each subset. We then compute the relative difference between the parameter values, $\Delta\mathbf{a}_{pq}$, and change in model error, Δe_{pq}, as:

$$\Delta\mathbf{a}_{pq} = [\frac{|b_1 - c_1|}{a_1},...,\frac{|b_m - c_m|}{a_m}],\ \Delta e_{pq} = e - e_{pq}(S_b) - e_{pq}(S_c)$$

where $\mathbf{b}$ and $\mathbf{c}$ are the parameters computed from the subsets S_a and S_b, respectively. This procedure is illustrated in Fig. 2. The iteration continues until all variables of interest are explored. The values $\Delta\mathbf{a}_{pq}$ and Δe_{pq}, where $p = 1,...,M$ and $q = 1,...,N$ are used to identify combinations of variables that results in a significant change in parameter values (and a large decrease in the classification error) and expand the logistic model by adding the corresponding quadratic terms, e.g.

$$(a_{11} + d_{11,pq} u_{pq}) u_{11} \ldots - (a_{21} + d_{21,st} u_{st}) u_{21} - \ldots$$

Once these new terms are added to the model, the final parameters, **a** and **d**, are computed via (7).

5. RESULTS AND CONCLUSIONS

We have applied the proposed approach in the previously described dashboard design problem, to track the portfolio of large IT services clients in the IBM Global Services division and identify those who are likely to terminate or reduce the scope of their engagement. The training set consists of 84 IT services clients over the period of three years. Some clients have been "measured" at different time periods yielding a dataset with 148 samples (79 examples of companies that terminated their services engagement and 69 examples of companies that had no significant changes to their services contract). As input to the model we have selected 18 explanatory variables, grouped according to the five risk factors, $\mathbf{u} = [\mathbf{u}_1 \ldots \mathbf{u}_5]$. We used: 1) nine financial metrics (revenue growth, earnings volatility, return on assets, expense growth, etc.), $\mathbf{u}_1 = [u_{11} \ldots u_{19}]$, reflecting the financial health of the client, (x_1), 2) three inputs from customer surveys, $\mathbf{u}_2 = [u_{21}\ u_{22}\ u_{23}]$, reflecting client satisfaction, (x_2), 3) two inputs, $\mathbf{u}_3 = [u_{31}\ u_{32}]$, that capture the price of the service engagement, (x_3), 4) three inputs representing the number of officer changes, share repurchases and restructurings in the trailing 12-month period, $\mathbf{u}_4 = [u_{41}\ u_{42}\ u_{43}]$, reflecting significant changes in client company, (x_4), and 5) one variable $\mathbf{u}_5 = [u_{51}]$ indicating the previous history of contract changes, (x_5).

We have first compared the model errors for the standard maximum a posteriori classifier (MAP), logistic regression model (LRM) trained via (3) and the proposed adaptive logistic regression classifier (ALRM). The ALRM model has been trained by computing the initial values for hidden variables $x_1 - x_5$ via IRLS, modifying some of the initial estimates to reflect known relationships in the data, and re-estimating the parameters of the model via (7). Furthermore, the adaptation procedure (described in Section 4) has identified three significant non-linear relationships and the model has been expanded with the corresponding quadratic terms, resulting in total of 21 parameters for the ALRM model. The comparison between the model errors is given in Table 1. Note that both LRM and ALRM1 have the same training error (8), however ALRM1 captures the specified relationships better - the value of state error (6) is significantly smaller for ALRM than for the LRM model.

The small size of the training data does not allow for the use of independent sets to train and test the model. Therefore, we have assessed the performance of the classifier by using the standard leave-one-out cross validation approach [9]. Table 2 compares the classification error between the MAP classifier, LRM and ALRM models. As it can be seen introducing the additional variables identified in the adaptation procedure significantly improves the classification accuracy. Fig. 3 illustrates an example from a dashboard application developed for IBM Global Services division (using the new method) to monitor their largest IT services customers. Our results indicate that in small-sample classification problems with "non-linear" relationships the proposed approach has advantages over the traditional methods.

TABLE 1: THE COMPARISON BETWEEN THE TRADITIONAL LOGISTIC REGRESSION MODEL AND THE NEW APPROACH. TRAINING ERROR IS COMPUTED VIA (8), STATE ERROR IS COMPUTED VIA (6).

Model type	Training error	State error	# inputs
MAP	0.297	0.561	18
LRM	0.229	0.365	18
ALRM1 (without adaptation)	0.229	0.029	18
ALRM2 (with adaptation)	0.149	0.017	21

TABLE 2: THE COMPARISON BETWEEN THE CLASSIFICATION ACCURACY OF TRADITIONAL MODELS AND THE NEW APPROACH.

Model type	Classification error	# inputs
MAP	0.301	18
LRM	0.249	18
ALRM2	0.158	21

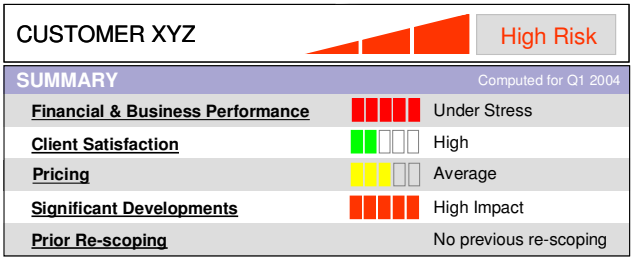

Fig. 3: An example of a client scorecard from the business analytics application that uses the described approach.

6. REFERENCES

[1] D. S. Rosario, "Highly effective logistic regression model for signal (anomaly) detection", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Montreal, Canada, May 2004.

[2] V. Solo, "Adaptive algorithms and Markov Chain Monte Carlo methods", *Proc. of the IEEE Conference on Decision and Control*, Phoenix, Arizona, December 1999.

[3] D. Brown, J. Stile, L. Gunderson, T. C. Giras, "Mining human failure dynamics from accident data using logistic regression and decision trees", *Proc. IEEE Int. conf. on Systems, Man and Cybernetics*, Hammamet, Tunisia, October 2002.

[4] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine", *Proc. of Neural Information Processing Systems*, Vancouver, Canada, December 2001.

[5] B. Scholkopf, and A. J. Smola, *Learning with kernels*, MIT Press, Cambridge, MA, 2002.

[6] B. D. Ripley, *Pattern classification and neural networks*, Cambridge University Press, 1996

[7] M. Jordan, R. Jacobs, "Hierachical mixtures of experts and the EM algorithm", *Neural Computation*, vol. 6, pp. 181-214, 1994.

[8] H. Chipman, E. I. George, and R. E. McCulloch, "Bayesian treed models", *Machine Learning*, vol. 48, pp. 299-320, 2002.

[9] L. Breiman, J. H Friedman, R. A. Olshen and C. J. Stone. *Classification and Regression Trees*. Wadsworth & Brooks, Pacific Grove, CA, 1984.

[10] R. Duin, "Classifiers in almost empty spaces", *International Conference on Pattern Recognition*, Barcelona, Spain, Sept. 2000.