Logistic Regression with the Nonnegative Garrote

Enes Makalic and Daniel F. Schmidt

Centre for MEGA Epidemiology, The University of Melbourne
Carlton VIC 3053, Australia
{emakalic,dschmidt}@unimelb.edu.au

Abstract. Logistic regression is one of the most commonly applied statistical methods for binary classification problems. This paper considers the nonnegative garrote regularization penalty in logistic models and derives an optimization algorithm for minimizing the resultant penalty function. The search algorithm is computationally efficient and can be used even when the number of regressors is much larger than the number of samples. As the nonnegative garrote requires an initial estimate of the parameters, a number of possible estimators are compared and contrasted. Logistic regression with the nonnegative garrote is then compared with several popular regularization methods in a set of comprehensive numerical simulations. The proposed method attained excellent performance in terms of prediction rate and variable selection accuracy on both real and artificially generated data.

1 Introduction

Logistic regression is one of the most commonly applied statistical methods for binary classification problems. Here, one observes n data samples

$$\{(y_i, x_{i1}, \ldots, x_{ip}), i = 1, \ldots, n\}$$

comprising p predictor variables and a binary class indicator $y \in \{-1, +1\}$ which denotes the class membership of the observed predictors. The conditional probability that a vector of covariates $\mathbf{x} = (x_1, \ldots, x_p)'$ is assigned to class y is

$$p(y = \pm 1 | \mathbf{x}, \boldsymbol{\beta}) = \frac{1}{1 + \exp(-y\mathbf{x}'\boldsymbol{\beta})} \tag{1}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)' \in \mathbb{R}^p$ are the regression coefficients. The log-likelihood of a logistic regression is then given by

$$l(\boldsymbol{\beta}) = -\sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{x}_i' \boldsymbol{\beta})\right) \tag{2}$$

which is a function of the regression parameters $\boldsymbol{\beta}$. The regression coefficients determine the probability of the target variable. That is, a positive regression coefficient for a predictor implies that the predictor is associated with an increased

D. Wang and M. Reynolds (Eds.): AI 2011, LNAI 7106, pp. 82–91, 2011.

probability of the response ($y = +1$), while a negative parameter coefficient reduces the response probability. A regression coefficient of zero has no effect on the conditional class probability and should ideally be excluded from the final model. The main task in inference of logistic models is to estimate the parameter coefficients $\boldsymbol{\beta}$ and select which of the p observed covariate vectors, if any, are useful in explaining the target variable.

This paper applies the nonnegative garrote to logistic regression models and examines the performance of the resulting procedure under various settings of sample size, number of predictors and regressor correlation. There are three main contributions in the paper: (1) an efficient algorithm for the implementation of NNG in logistic regression models, (2) empirical evaluation of several initial estimators for the NNG, and (3) extensive performance comparison in terms of prediction and variable selection of the NNG procedure with several popular regularization algorithms.

2 Nonnegative Garrote

Let $\boldsymbol{\beta}^* \in \mathbb{R}^p$ be an initial estimate of the logistic regression parameters, for example, the maximum likelihood estimate or a ridge regression estimate. Denote a shrunken estimate of $\boldsymbol{\beta}^*$ as $\tilde{\boldsymbol{\beta}}(\mathbf{c}) = (\mathbf{c} \odot \boldsymbol{\beta}^*)$ where $\mathbf{c} = (c_1, \ldots, c_p)'$ and the operator $\odot$ is the Hadamard (element-wise) product. The nonnegative garrote estimate [1] is defined as the solution to

$$\boldsymbol{\beta}_\lambda = \underset{\tilde{\boldsymbol{\beta}}(\mathbf{c})}{\arg\max} \left\{ l(\tilde{\boldsymbol{\beta}}) \right\} = \underset{\tilde{\boldsymbol{\beta}}(\mathbf{c})}{\arg\min} \left\{ \sum_{i=1}^{n} \log \left(1 + \exp(-y_i \mathbf{x}_i' \tilde{\boldsymbol{\beta}}) \right) + \lambda \sum_{j=1}^{p} c_j \right\} \quad (3)$$

subject to the constraints

$$c_j \geq 0, \quad (j = 1, 2, \ldots, p) \quad (4)$$

and assuming that the initial parameter estimate $\boldsymbol{\beta}^*$ is kept fixed. The NNG shrinks the initial parameter estimates by varying the multiplier $\mathbf{c}$. The regularization parameter $\lambda > 0$ controls the amount of shrinkage that is applied to the initial parameter estimates. Increasing λ (tightening the garrote) results in more of the initial parameters being set to zero and greater shrinkage of the non-zero components of $\boldsymbol{\beta}^*$. In contrast, decreasing the regularization parameter induces less shrinkage leading to a final solution that is closer to the starting parameter estimates. In this way, the NNG allows for both parameter shrinkage and variable selection automatically. In practice, the regularization parameter may be selected using a model selection criterion such as the Bayesian information criterion (BIC) [2].

There is no clear consensus as to which initial estimator should be used with the NNG. Breiman [1] originally advocated the maximum likelihood estimator to be used as the initial estimate in linear models. There are three disadvantages of this approach in logistic regression: (1) the maximum likelihood estimator cannot

Algorithm 1. Cyclic coordinate descent for nonnegative garrote (nng)

input : data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, target vector $\mathbf{y} \in \{-1, +1\}^n$, initial estimate $\boldsymbol{\beta}^* \in \mathbb{R}^p$, regularization parameter $\lambda > 0$
output: NNG estimate $\boldsymbol{\beta} \in \mathbb{R}^p$

initialize $\Delta_j \leftarrow 1$ for $j = 1, \ldots, p$, $\quad \Delta r_i \leftarrow 0$ for $i = 1, \ldots, n$
$r \leftarrow y \odot X\boldsymbol{\beta}^*$ (⊙ denotes element-wise product)
$\mathbf{x}_i \leftarrow \mathbf{x}_i \odot \boldsymbol{\beta}^* \quad (i = 1, \ldots, n)$ (rescale data)
$\beta \leftarrow (1, \ldots, 1)'$ (start search from $\boldsymbol{\beta}^*$)
for $t \leftarrow 1, 2, \ldots$ **to** convergence **do**
  for $j \leftarrow 1, 2, \ldots$ **to** p **do**
    $F_i \leftarrow$
    $\min(0.25, 1/(2\exp(-\Delta_j|x_{ij}|) + \exp(r_i - \Delta_j|x_{ij}|) + \exp(\Delta_j|x_{ij}| - r_i))$
    $(i = 1, \ldots, n)$
    $\Delta v_j \leftarrow \left(\sum_{i=1}^n x_{ij} y_i/(1 + \exp(r_i)) - \lambda\right) / (\sum_{i=1}^n x_{ij}^2 F_i)$
    (Newton--Raphson update)
    if $\beta_j = 0$ **then**
      if $\Delta v_j \leq 0$ **then**
        $\Delta v_j = 0$
      end
    else
      if $\beta_j + \Delta v_j < 0$ **then**
        $\Delta v_j = -\beta_j$ (if sign change, set β_j to zero)
      end
    end
    $\Delta\beta_j \leftarrow \min(\max(\Delta v_j, -\Delta_j), \Delta_j)$ (limit step size to trust region)
    $\Delta r_i \leftarrow \Delta\beta_j X_{ij} y_i, \quad r_i \leftarrow r_i + \Delta r_i \quad (i = 1, \ldots, n)$
    $\beta_j \leftarrow \beta_j + \Delta\beta_j$
    $\Delta_j \leftarrow \max(2|\Delta\beta_j|, \Delta_j/2)$ (update trust region size)
  end
end
$\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} \odot \boldsymbol{\beta}^*$ (use original scale)

be used if the number of predictors is greater than the number of samples ($p > n$ setting) or the covariates are highly correlated, (2) the maximum likelihood estimator performs poorly when the sample size is small, and (3) the maximum likelihood estimator does not exist if the data is quasicompletely or completely separable [3]. In this paper, following [4], we compare and contrast a number of alternative initial estimators.

A software implementation of NNG logistic regression requires some thought since the standard convex programming solution to (3)-(4) is not feasible when the number of covariates is large. The NNG solution was originally implemented using constrained least squares minimization in the linear regression setting. This approach is however not possible in logistic regression models and subsequently a number of alternative optimization routines have been proposed [5,6,7]. This

paper employs a numerical optimization routine based on the cyclic coordinate descent method detailed in [5]. The cyclic coordinate descent method was chosen because of the low computational complexity and the fact that the algorithm can be used when the number of predictors is large, potentially much larger than the sample size. The pseudo-code for the proposed optimization routine is shown in in Algorithm 1.

Our modified algorithm, henceforth `NNG_OPT`, begins by transforming the data matrix $\mathbf{X}$ and the initial parameter vector $\boldsymbol{\beta}^*$ in such a way that the transformed regression parameters are restricted to be positive (lines 3–4). Contrary to the LASSO where a variable can change signs during the optimization, the NNG multiplier factor $\mathbf{c}$ is strictly positive. Subsequently, our algorithm checks for sign changes in variables (lines 9–17) and does not allow negative multiplier parameters. A Newton–Raphson update is then performed for each regression parameter, while all the remaining parameters are kept fixed (lines 19–20). During the optimization, the variable r is used to keep track of the product $y \odot X\boldsymbol{\beta}$ for speed purposes. The optimization steps are performed until convergence is reached; `NNG_OPT` uses the convergence criterion recommended in [5]. Note, when implementing `NNG_OPT`, special care needs to be taken for the constant regressor which should not be subject to shrinkage. A MATLAB$^{\text{TM}}$ implementation of `NNG_OPT` is available from the authors upon request.

3 Simulation

This section examines finite sample performance of the NNG estimator using artificially generated data (see Section 3.1) as well as real data (see Section 3.4). Since the performance of the NNG estimator depends on the initial estimate [4], four different initial estimates are considered: (1) stepwise forward selection (`fwd`), (2) ridge regression (`rr`) [8], (3) the least angle shrinkage and selection operators (`lasso`) [9], and (4) the elastic net (`enet`) [10]. For completeness, a method that uses several possible ridge regression estimates, denoted `nng`, is also considered. The resulting NNG estimates are compared against the standard stepwise forward selection (`fwd`), ridge regression (`rr`), LASSO (`lasso`) and elastic net (`enet`) estimates. Furthermore, we have also included the iterated LASSO estimate (`ilasso`) [11] in all our comparisons, though it is not used as an initial estimate for an NNG solution. The performance of each method is measured using a variety of metrics including classification accuracy, size of the final model, the mean number of false positive regressors and the mean number of false negative regressors. Note that the constant regressor was included in all subsequent simulation runs but was not used when tabulating results.

3.1 Simulated Data

3.2 Path Consistency

The first simulation examined how often the NNG estimator and the popular LASSO estimator select only the true regression coefficients from artificially

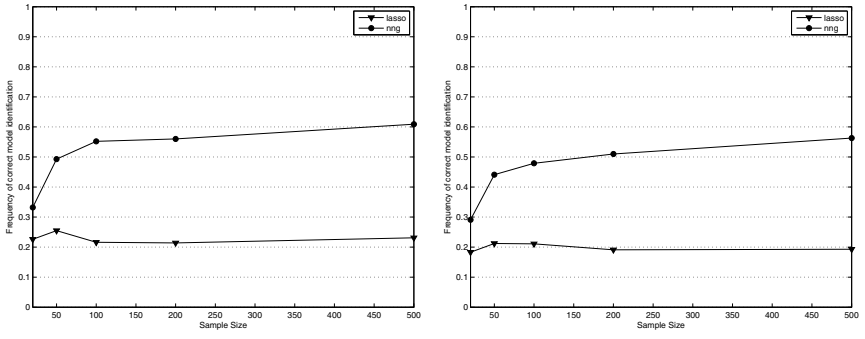
generated data. This property is known as path consistency. The simulation closely followed the setup in ([4], Example 1) for linear regression models. Here, the regressor matrix $\mathbf{X}$ consists of four regressors ($p = 4$) generated from:

$$\mathbf{x}_i = (1, X_1, X_2, X_3)' \qquad X_1, X_2 \sim N(0,1), \quad X_3 \sim N(\alpha(X_1+X_2), 1-2\alpha^2) \tag{5}$$

where $N(\cdot,\cdot)$ denotes the univariate normal distribution, $\alpha \in \{0.35, 0.55\}$ and $i = (1,\ldots,n)$. The true regression coefficients were set to

$$\boldsymbol{\beta} = (0,1,1,0)' \tag{6}$$

In all simulations, the class indicators $\mathbf{y} \in \{-1,+1\}^n$ were independently generated with probability given by (1). For each value of α, we generated training data with the following sample sizes $n = \{20, 50, 100, 200, 500\}$. The regularization parameter for both the LASSO and NNG algorithms was selected using the log-likelihood of an independently generated validation data set. The validation data set was of the same size as the corresponding training data set. The range of regularization parameters considered was chosen to comprise 1000 values of λ uniformly spaced between 10^{-5} and 10^2. The simulation comprised 100 training and validation data sets generated for each (n, α) pair. For each run, we recorded the number of times the LASSO and NNG correctly identified the true regression coefficients and excluded the noise variables. The NNG estimator was selected as follows: (1) train a number of initial models using ridge regression estimates, (2) obtain a NNG solution for each ridge regression model, and (3) use the validation data set to select the NNG model with the largest log-likelihood. Figure 1 depicts the frequency of true model identification by both LASSO and NNG.



(a) $\alpha = 0.35$ (b) $\alpha = 0.55$

Fig. 1. Path consistency of the LASSO and NNG estimators

When $\alpha = 0.35$, the NNG and LASSO select the true model with frequencies of 30% and 20% respectively for smaller samples ($n < 50$). As the sample size is

increased, the frequency of true model detection dramatically increases for the NNG and increases only slightly for the LASSO. At $n = 500$, for example, the NNG correctly identifies the true model 60% of the time, compared to about 25% for the LASSO. As noted in [4], increasing the value of α increases the difficulty of true model identification. At $\alpha = 0.55$, the LASSO is no longer path consistent and selects the true model approximately 20% of the time irrespective of the sample size. In contrast, the NNG remains path consistent and selects the true model with increasing frequency for larger sample sizes.

3.3 Initial Estimates for the NNG

The simulation involved generating 1000 training and validation data sets of $n \in \{20, 50, 100\}$ samples. The regularization parameters for each run were selected based on the log-likelihood of an independently generated validation set. The best subset for the stepwise forward selection method was selected using the same approach. In all simulations, the class indicators $\mathbf{y} \in \{-1, +1\}^n$ were generated independently with probability given by (1). Performance metrics recorded in each test run were: (1) negative log-likelihood, (2) model size, (3) the number of false positive regressors, and (4) the number of false negative regressors included in the best model. A regressor that was inferred to be zero is deemed to be a false positive if the data generating model has the corresponding coefficient set to a non-zero value, and similarly for false negative regressors. The following two simulation models were considered:

1. The true regression coefficients were set to $\boldsymbol{\beta} = (3, 1{\cdot}5, 0, 0, 2, 0, 0, 0)'$. The pairwise correlation between predictors i and j was $\text{corr}(i, j) = 0.5^{|i-j|}$ [9].
2. Same as Example (1), except $\beta_i = 0.85$ for all i [9].

The constant regressor was included in all simulations but was not used when tabulating results.

Example 1. The data generating model in this scenario is of medium sparsity with five out of eight regressors being noise. It is expected that ridge regression will not perform as well as the alternative methods given the level of sparsity. In contrast, the LASSO should do quite well as the data generating model does not contain highly correlated regressors. Simulation results for the test are shown in Table 1. The stepwise forward regression method (`fwd`) achieved the worst median negative log-likelihood from all the methods tested, with the performance being clearly inferior to other methods when the sample size was small ($n = 20$). From the four initial estimators tested, the `grr` method (NNG initialized with ridge regression) achieved the best median log-likelihood for the smallest sample size. However, as the sample size was increased, all four methods performed approximately equally well. In comparison to the starting estimates, the corresponding NNG method exhibited slightly more false positive regressors and significantly less false negative regressors. Additionally, the models selected by the NNG methods were generally smaller than any of the initial models.

Table 1. Simulation results for Example 1; median negative log-likelihood (NLL), mean model size (Size), mean number of false positive regressors (FP) and mean number of false negative regressors (FN) included in the selected model. Tests are based on 1000 iterations with standard errors included in parentheses.

Methods	$n = 20$				$n = 50$				$n = 100$			
	NLL	Size	FP	FN	NLL	Size	FP	FN	NLL	Size	FP	FN
fwd	30·70	1·24	1·96	0·20	7·12	2·79	0·63	0·42	2·92	3·46	0·11	0·57
	(0·80)	(0·04)	(0·03)	(0·02)	(0·07)	(0·04)	(0·03)	(0·03)	(0·01)	(0·04)	(0·01)	(0·04)
gfwd	26·64	1·18	1·97	0·15	6·74	2·75	0·64	0·38	2·84	3·36	0·12	0·47
	(0·41)	(0·04)	(0·03)	(0·02)	(0·05)	(0·04)	(0·02)	(0·03)	(0·01)	(0·04)	(0·01)	(0·03)
lasso	21·13	4·53	0·50	2·03	6·50	5·78	0·05	2·83	2·91	6·20	0·01	3·20
	(0·15)	(0·05)	(0·02)	(0·04)	(0·04)	(0·04)	(0·01)	(0·04)	(0·01)	(0·04)	(0·00)	(0·04)
glasso	22·40	2·89	0·93	0·82	6·44	3·97	0·23	1·20	2·85	4·33	0·02	1·35
	(0·18)	(0·04)	(0·03)	(0·03)	(0·05)	(0·04)	(0·01)	(0·04)	(0·01)	(0·04)	(0·00)	(0·04)
rr	21·13	8·00	0·00	5·00	6·76	8·00	0·00	5·00	3·00	8·00	0·00	5·00
	(0·14)	(0·00)	(0·00)	(0·00)	(0·03)	(0·00)	(0·00)	(0·00)	(0·01)	(0·00)	(0·00)	(0·00)
grr	21·86	3·37	0·77	1·14	6·45	4·32	0·17	1·49	2·86	4·51	0·02	1·53
	(0·21)	(0·05)	(0·02)	(0·03)	(0·03)	(0·04)	(0·01)	(0·04)	(0·01)	(0·04)	(0·00)	(0·04)
enet	20·64	6·10	0·21	3·31	6·50	6·40	0·02	3·42	2·92	6·50	0·00	3·51
	(0·12)	(0·05)	(0·01)	(0·05)	(0·03)	(0·04)	(0·00)	(0·04)	(0·01)	(0·04)	(0·00)	(0·04)
genet	22·20	3·07	0·85	0·92	6·42	4·06	0·20	1·26	2·85	4·34	0·02	1·36
	(0·22)	(0·04)	(0·02)	(0·03)	(0·04)	(0·04)	(0·01)	(0·04)	(0·01)	(0·04)	(0·00)	(0·04)
ilasso	22·41	2·90	0·93	0·83	6·42	3·98	0·22	1·20	2·85	4·33	0·02	1·35
	(0·19)	(0·04)	(0·02)	(0·03)	(0·05)	(0·04)	(0·01)	(0·04)	(0·01)	(0·04)	(0·00)	(0·04)
nng	21·34	3·50	0·66	1·16	6·34	4·35	0·11	1·46	2·84	4·51	0·01	1·52
	(0·25)	(0·04)	(0·02)	(0·03)	(0·03)	(0·04)	(0·01)	(0·04)	(0·01)	(0·04)	(0·00)	(0·04)

As the sample size was increased to $n = 100$, all methods performed equally well in terms of log-likelihood. Interestingly, the `nng` solution appeared to perform the same as using only the single best ridge regression estimate (`grr`) in this example.

Example 2. The true model is now dense and does not include any noise regressors which would make ridge regression the ideal solution for this type of problem. Table 2 depicts the corresponding simulation results. As in Example 1, all methods performed equally well given enough data. It is therefore of interest to examine regularization performance under small sample sizes ($n = 20$). Stepwise forward selection (`fwd`) attained the highest negative log-likelihood of all the regularization methods tested. The performance of `fwd` was especially poor when $n = 20$, obtaining the highest negative log-likelihood and largest number of false positives. Ridge regression achieved the smallest negative log-likelihood (`rr`) of all the methods tested for all sample sizes. This is not surprising give that the generating model is dense and ridge regression cannot zero out individual regressors. In contrast, using the NNG with LASSO, ridge regression and elastic net resulted in a somewhat worse log-likelihood and a significantly sparser solution, compared to the original model. This indicates that the NNG is producing models that are too sparse which agrees with the findings in [4]. Models superior to the corresponding initial estimates were obtained only when the NNG was

Table 2. Simulation results for Example 2; median negative log-likelihood (NLL), mean model size (Size), mean number of false positive regressors (FP) and mean number of false negative regressors (FN) included in the selected model. Tests are based on 1000 iterations with standard errors included in parentheses.

Methods	$n = 20$				$n = 50$				$n = 100$			
	NLL	Size	FP	FN	NLL	Size	FP	FN	NLL	Size	FP	FN
fwd	35·10	1·17	6·83	0·00	10·20	4·27	3·73	0·00	3·73	6·94	1·06	0·00
	(0·08)	(0·05)	(0·05)	(0·00)	(0·09)	(0·07)	(0·07)	(0·00)	(0·02)	(0·05)	(0·05)	(0·00)
gfwd	34·66	1·11	6·89	0·00	9·19	4·05	3·94	0·00	3·68	6·69	1·31	0·00
	(0·03)	(0·04)	(0·04)	(0·00)	(0·08)	(0·07)	(0·07)	(0·00)	(0·02)	(0·05)	(0·05)	(0·00)
lasso	24·83	4·95	3·06	0·00	7·76	6·94	1·06	0·00	3·50	7·76	0·23	0·00
	(0·19)	(0·05)	(0·05)	(0·00)	(0·04)	(0·03)	(0·03)	(0·00)	(0·01)	(0·01)	(0·01)	(0·00)
glasso	28·24	3·14	4·86	0·00	8·44	5·66	2·34	0·00	3·62	7·24	0·76	0·00
	(0·19)	(0·05)	(0·05)	(0·00)	(0·05)	(0·05)	(0·05)	(0·00)	(0·02)	(0·03)	(0·03)	(0·00)
rr	21·23	8·00	0·00	0·00	7·18	8·00	0·00	0·00	3·38	8·00	0·00	0·00
	(0·11)	(0·00)	(0·00)	(0·00)	(0·03)	(0·00)	(0·00)	(0·00)	(0·01)	(0·00)	(0·00)	(0·00)
grr	26·97	3·80	4·20	0·00	8·23	6·10	1·90	0·00	3·59	7·42	0·58	0·00
	(0·19)	(0·05)	(0·05)	(0·00)	(0·04)	(0·04)	(0·04)	(0·00)	(0·02)	(0·03)	(0·03)	(0·00)
enet	21·59	7·40	0·60	0·00	7·24	7·88	0·12	0·00	3·38	7·98	0·02	0·00
	(0·12)	(0·04)	(0·04)	(0·00)	(0·02)	(0·01)	(0·01)	(0·00)	(0·01)	(0·00)	(0·00)	(0·00)
genet	27·20	3·65	4·35	0·00	8·24	6·06	1·94	0·00	3·59	7·42	0·58	0·00
	(0·22)	(0·05)	(0·05)	(0·00)	(0·04)	(0·04)	(0·04)	(0·00)	(0·02)	(0·03)	(0·03)	(0·00)
ilasso	28·23	3·15	4·85	0·00	8·44	5·67	2·33	0·00	3·62	7·24	0·76	0·00
	(0·21)	(0·05)	(0·05)	(0·00)	(0·05)	(0·05)	(0·05)	(0·00)	(0·02)	(0·03)	(0·03)	(0·00)
nng	26·49	4·08	3·92	0·00	8·05	6·33	1·67	0·00	3·54	7·57	0·43	0·00
	(0·23)	(0·05)	(0·05)	(0·00)	(0·04)	(0·04)	(0·04)	(0·00)	(0·01)	(0·02)	(0·02)	(0·00)

used with stepwise forward selection. The `nng` approach attained the best negative log-likelihood and somewhat larger models in contrast to alternative NNG strategies.

3.4 Real Data

This section examines the performance of logistic regression regularization solutions using six real data sets from the UCI Machine Learning repository. The number of regressors, excluding the constant regressor, ranged from small ($p = 4$ in "transfusion") to moderate ($p = 60$ in "sonar"). All data sets were standardized to have $||\mathbf{x}_j|| = 1$ $(j = 1, \ldots, p)$, where $||\cdot||$ denotes the Euclidean norm. For each data set, we randomly split the available data into a training, a validation and a test subset. The training data set was used to infer the parameter estimates, while the validation data set was used for selecting the regularization parameters. All tabulated results are based only on the test set. There were 100 simulation runs for each data set. For each iteration, two performance metrics were recorded: (1) classification accuracy, and (2) model size in terms of the number of regressors remaining. Stepwise forward selection was not included in this test due to its poor performance in previous experiments as well as the relatively high computational complexity of the method. The simulation results are shown in Table 3.

On the pima data set, the best classification accuracy of all the methods tested was attained by `nng`, closely followed by `grr` and `genet`. Although `rr` resulted

Table 3. Simulation results for real data. Median classification accuracy (in percent) is shown along with bootstrap estimates of standard error. Mean model size is included in parentheses. Tests are based on 100 iterations.

Methods	Datasets					
	pima	wdbc	spambase	ionosphere	transfusion	sonar
lasso	74.82 ± 0.30 (6.52)	95.53 ± 0.18 (7.78)	91.62 ± 0.09 (48.09)	79.68 ± 0.69 (7.73)	78.46 ± 0.29 (3.68)	71.02 ± 0.29 (10.81)
glasso	74.82 ± 0.28 (4.61)	95.12 ± 0.20 (4.64)	91.79 ± 0.09 (35.22)	79.68 ± 0.58 (3.67)	78.35 ± 0.29 (3.42)	69.32 ± 0.29 (4.57)
rr	74.30 ± 0.32 (8.00)	95.93 ± 0.16 (30.00)	91.45 ± 0.12 (57.00)	81.27 ± 0.53 (32.00)	78.57 ± 0.20 (4.00)	75.00 ± 0.20 (60.00)
grr	75.18 ± 0.29 (4.77)	95.39 ± 0.15 (6.21)	91.75 ± 0.11 (39.04)	79.88 ± 0.32 (5.53)	78.79 ± 0.33 (3.53)	69.89 ± 0.33 (7.54)
enet	74.65 ± 0.31 (7.05)	96.21 ± 0.16 (23.03)	91.48 ± 0.11 (51.61)	81.27 ± 0.40 (22.27)	78.57 ± 0.19 (3.92)	75.00 ± 0.19 (53.06)
genet	75.00 ± 0.23 (4.70)	95.12 ± 0.16 (5.82)	91.77 ± 0.09 (37.47)	80.28 ± 0.41 (5.10)	78.79 ± 0.35 (3.52)	68.75 ± 0.35 (7.20)
ilasso	74.82 ± 0.27 (4.61)	95.12 ± 0.20 (4.82)	91.77 ± 0.08 (35.28)	79.88 ± 0.55 (3.79)	78.35 ± 0.29 (3.44)	69.32 ± 0.29 (4.75)
nng	75.35 ± 0.21 (4.80)	95.66 ± 0.19 (6.63)	91.77 ± 0.08 (40.49)	80.48 ± 0.36 (6.53)	78.35 ± 0.37 (3.49)	71.59 ± 0.37 (7.94)

in the lowest classification accuracy on this data set, the difference between `rr` and `nng` was only about 1%. It is clear that applying the NNG to any initial estimate has again resulted in a more parsimonous model, which is still highly predictive. For example, `lasso` models have on average 6.5 regressors compared to 4.6 for the `glasso` for about the same classification accuracy. Although the iterated LASSO did not improve on the LASSO, it generally inferred sparser models. The elastic net obtained the best classification accuracy on the wdbc data set out of all the methods tested. However, the average model inferred by `enet` was approximately four times the size of the average `nng` model, while the classification accuracy of `nng` was only slightly smaller (95.6% for `nng` versus 96.2% for `enet`).

All methods performed equally well on the spambase dataset in terms of classification accuracy. In terms of model complexity, the LASSO and the elastic net resulted in the largest models, while the NNG based methods as well as the iterative LASSO inferred models with about 10 regressors less, on average. Similar findings can be noted for the ionosphere, transfusion and sonar data sets. We did observe an interesting anomaly on the sonar dataset. The elastic net and ridge regression obtained significantly higher prediction accuracy and larger average model size, in contrast to all other methods considered. For example, the `nng` inferred models were significantly simpler and resulted in reduced classification accuracy of about 5%. Given the size of the test data, an increase in accuracy of 5% equates to four extra samples being correctly classified by `enet` and `rr`, which is not a highly significant improvement.

3.5 Discussion and Recommendations

Simulations in Section 3.3 clearly show that stepwise forward selection commonly resulted in models which generalize poorly, especially given small to medium

sample sizes. Predictive performance of stepwise methods remained poor irrespective of the sparsity of the data generating model. The nonnegative garrote, or the iterated LASSO, is recommended if the data generating model is expected to be (highly) sparse. While most of the considered regularization strategies showed promising performance in the sparse setting, the consistency of NNG and the iterated LASSO (see Section 3.2) make the techniques highly suitable. Models inferred by the NNG were consistently simpler and attained significantly smaller numbers of false negatives in contrast to most other methods considered. Interestingly, the iterated LASSO has outperformed the original LASSO in terms of prediction and model size and is thus recommended for logistic regression if a LASSO-type penalty is desired. Although the elastic net achieved similar classification performance to the NNG, the models inferred by the elastic net consisted of significantly more regressors.

A ridge regression estimate is recommended as the starting point for NNG over maximum likelihood or LASSO-type solutions. Ridge regression allows the NNG to be applied to collinear models which is otherwise not possible with the maximum likelihood approach. Unlike ridge regression, LASSO and the elastic net generate sparse models which implies that some coefficients will be set to zero prior to running the NNG. Due to the form of the NNG penalty, regression coefficients are not altered once set to zero. Thus, if a sparse solution, like the LASSO, is used for the initial estimates, coefficients that the NNG would normally retain may be rendered insignificant. Our recommendation of ridge regression as an initial solution to NNG is in agreement with the findings published in [4].

References

1. Breiman, L.: Better subset regression using the nonnegative garrote. Technometrics 37, 373–384 (1995)
2. Xiong, S.: Some notes on the nonnegative garrote. Technometrics 52(3), 349–361 (2010)
3. Albert, A., Anderson, J.A.: On the existence of maximum likelihood estimates in logistic regression models. Biometrika 71(1), 1–10 (1984)
4. Yuan, M., Lin, Y.: On the non-negative garrotte estimator. Journal of the Royal Statistical Society (Series B) 69(2), 143–161 (2007)
5. Genkin, A., Lewis, D.D., Madigan, D.: Large-scale Bayesian logistic regression for text categorization. Technometrics 49(3), 291–304 (2007)
6. Park, M.Y., Hastie, T.: L_1-regularization path algorithm for generalized linear models. Journal of the Royal Statistical Society (Series B) 69(4), 659–677 (2007)
7. Friedman, J., Hastie, T., Tibshirani, R.: Regularized paths for generalized linear models via coordinate descent. Journal of Statistical Software 33(1) (2010)
8. Cessie, S.L., Houwelingen, J.C.V.: Ridge estimators in logistic regression. Journal of the Royal Statistical Society (Series C) 41(1), 191–201 (1992)
9. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (Series B) 58(1), 267–288 (1996)
10. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society (Series B) 67(2), 301–320 (2005)
11. Huang, J., Ma, S., hui Zhang, C.: The iterated lasso for high-dimensional logistic regression. Technical Report 392, The University of Iowa (2008)