Communications for Statistical Applications and Methods
2020, Vol. 27, No. 1, 129–140

https://doi.org/10.29220/CSAM.2020.27.1.129
Print ISSN 2287-7843 / Online ISSN 2383-4757

An efficient algorithm for the non-convex penalized multinomial logistic regression

Sunghoon Kwon[a], Dongshin Kim[b], Sangin Lee[1,c]

[a]Department of Applied Statistics, Konkuk University, Korea;
[b]Graziadio School of Business and Management, Pepperdine University, USA
[c]Department of Information and Statistics, Chungnam National University, Korea

Abstract

In this paper, we introduce an efficient algorithm for the non-convex penalized multinomial logistic regression that can be uniformly applied to a class of non-convex penalties. The class includes most non-convex penalties such as the smoothly clipped absolute deviation, minimax concave and bridge penalties. The algorithm is developed based on the concave-convex procedure and modified local quadratic approximation algorithm. However, usual quadratic approximation may slow down computational speed since the dimension of the Hessian matrix depends on the number of categories of the output variable. For this issue, we use a uniform bound of the Hessian matrix in the quadratic approximation. The algorithm is available from the R package ncpen developed by the authors. Numerical studies via simulations and real data sets are provided for illustration.

Keywords: concave-convex procedure, modified local quadratic approximation algorithm, multinomial logistic regression, non-convex penalty

1. Introduction

In statistical learning, the multiclass classification is the problem of classifying samples into a specific category when there are more than two possible categories. There are various real filed applications of multiclass classification. For example, we can conduct cancer diagnosis from gene microarrays (Zhu and Hastie, 2004) or distinguish car types from various care images (Huttunen *et al.*, 2016). One of popular methods for multiclass classification is the multinomial logistic regression that assumes the multinomial distribution for the samples to be classified.

For years, the penalized multinomial logistic regression has been studied by many authors since there can be many noisy variables among the input variables. We can avoid unnecessary modeling biases by deleting the noisy input variables from the model, which often results in higher classification accuracy. For example, Krishnapuram *et al.* (2005) proposed to use the least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996) and ridge (Hoerl and Kennard, 1970). They developed a fast quadratic approximation algorithm for maximizing the penalized multinomial likelihood, where the Hessian matrix is uniformly bounded by a positive definite matrix (Böhning, 1992). Kim *et al.* (2006) proposed the sparse one-against-all logistic regression using the gradient LASSO algorithm developed by Kim *et al.* (2008). Cawley *et al.* (2007) proposed the Bayesian LASSO that significantly reduces computational expense by integrating out the usual tuning parameter in the LASSO. Simon

[1] Corresponding author: Department of Information and Statistics, Chungnam National University, 99 Daehak-ro, Yuseong-gu, Daejeon 34134, Korea. E-mail: sanginlee44@gmail.com

Published 31 January 2020 / journal homepage: http://csam.or.kr

et al. (2013) applied the group LASSO (Yuan and Lin, 2006) by treating the parameters in each class as grouped parameters in the group LASSO. Chen *et al.* (2014) adapted the elastic net (Zou and Hastie, 2005) for imposing group effects on the input variables which often serves to improve prediction accuracy. Tutz *et al.* (2015) developed a category-specific group LASSO for cases when a set of category-specific predictors are available (Tutz, 2011).

In general, convex penalties such as the LASSO and elastic net are known to select input variables more than necessary unless a certain condition on the design matrix (Zhao and Yu, 2006) is satisfied. On the other hand, non-convex penalties have been proven to have the oracle property for a wide range of statistical models, including the generalized linear models (Fan and Peng, 2004; Kwon and Kim, 2012), random effect models, (Bondell *et al.*, 2010; Kwon *et al.*, 2016) and non-parametric regression models (Xie and Huang, 2009; Huang *et al.*, 2010). However, up to the authors' knowledge, there are very few literatures that have concentrated on the multinomial logistic regression with non-convex penalties. One main reason comes from the lack of efficient computational algorithms that implement the penalized estimators. Although there are some unified algorithms studied before (Kwon and Kim, 2012; Lee *et al.*, 2016), data analysts still feel annoying or uncomfortable from working with non-convex penalties for multinomial logistic regression.

In this paper, we introduce an efficient algorithm for the non-convex penalized multinomial logistic regression that can be uniformly applied to a class of non-convex penalties. The class includes most non-convex penalties such as the smoothly clipped absolute deviation (SCAD) (Fan and Li, 2001), minimax concave (MC) (Zhang, 2010) and bridge (Huang *et al.*, 2008) penalties. The algorithm is developed based on the concave-convex procedure (CCCP) (Yuille and Rangarajan, 2002) and modified local quadratic approximation (MLQA) algorithm (Lee *et al.*, 2016). However, usual quadratic approximation may slow down computational speed since the dimension of the Hessian matrix depends on the number of categories of the output variable. For this issue, we use a uniform bound of the Hessian matrix introduced by Böhning (1992) when we apply the MLQA algorithm. The algorithm is available from R package ncpen that has been developed by the authors. Numerical studies via simulations and real data sets are provided.

The rest of the paper consists of the following. Section 2 introduces the non-convex penalized multinomial logistic regression. Section 3 presents some details on the algorithm. Numerical studies and concluding remarks follow in Sections 4 and 5.

2. Non-convex penalized multinomial logistic regression

2.1. Penalized multinomal likelihood

Let (y_i, x_i), $i \leq n$, be n output and input sample pairs, where $y_i \in \{1, \ldots, m+1\}$ is an output to be classified, $m+1$ is the number of distinct categories and $x_i = (x_{i1}, \ldots, x_{ip})^T$ is a p-dimensional input vector. The multinomial logistic regression assumes $\mathbf{P}(y_i = k|x_i) = p_{ik}$, $i \leq n$, $k \leq m+1$, where

$$p_{ik} = \frac{\exp\left(x_i^T \pi_k\right)}{\sum_{\ell=1}^{m+1} \exp\left(x_i^T \pi_\ell\right)} \tag{2.1}$$

and $\pi_k = (\pi_{k1}, \ldots, \pi_{kp})^T$. Without loss of generality, we set $\pi_{m+1} = 0_p$ for a reference level, where 0_p is the zero vector of length p, which makes the model (2.1) identifiable. Then the negative log-likelihood to be minimized becomes

$$\ell(\pi) = -\sum_{i=1}^{n}\sum_{k=1}^{m+1} y_{ik}\log(p_{ik}) = -\sum_{i=1}^{n}\sum_{k=1}^{m}\left\{y_{ik}x_i^T\pi_k - \log\left(1+\sum_{\ell=1}^{m}\exp\left(x_i^T\pi_\ell\right)\right)\right\}, \tag{2.2}$$

where $\pi = (\pi_1^T, \ldots, \pi_m^T)^T$ and $y_{ik} = I(y_i = k)$.

Let ψ_λ be a penalty then the penalized estimator with respect to ψ_λ is defined as a local or global minimizer of the penalized negative log-likelihood:

$$\ell_\lambda(\pi) = \ell(\pi) + \sum_{k=1}^{m}\sum_{j=1}^{p}\psi_\lambda(|\pi_{kj}|), \tag{2.3}$$

where $\lambda > 0$ is an extra parameter that controls the model complexity, which is often called the tuning parameter. For example, the LASSO is equivalent to $\psi_\lambda(t) = \lambda t$, $t \geq 0$.

2.2. Non-convex penalties

We consider a class of non-convex penalties that satisfy:

(C1) $\psi_\lambda(|t|) = \int_0^{|t|}\nabla\psi_\lambda(s)ds$, $t \in \mathbb{R}$ for some non-decreasing function $\nabla\psi_\lambda$.

(C2) $\xi_\lambda(|t|) = \psi_\lambda(|t|) - \kappa_\lambda|t|$, $t \in \mathbb{R}$ is concave and continuously differentiable, where $\kappa_\lambda = \lim_{t\to 0+}\nabla\psi_\lambda(t)$.

There is a number of non-convex penalties that satisfy (C1) and (C2). Examples include flat-tailed non-convex penalties such as the SCAD penalty (Fan and Lv, 2011),

$$\nabla\psi_\lambda(t) = \lambda I(0 < t < \lambda) + \frac{a\lambda - t}{a-1}I(\lambda \leq t < a\lambda),$$

for $a > 2$, MC penalty (Zhang, 2010),

$$\nabla\psi_\lambda(t) = \left(\lambda - \frac{t}{a}\right)I(0 < t < a\lambda),$$

for $a > 1$ and capped or truncated ℓ_1 penalty (Zhang and Zhang, 2012; Shen *et al.*, 2012),

$$\nabla\psi_\lambda(t) = \lambda I(0 < t < a),$$

for $a > 0$. The class also includes some hybrid versions of existing convex and non-convex penalties. Let ψ_λ^M, ψ_λ^L, and ψ_λ^R be the MC, LASSO and ridge penalties, respectively. Then the class includes the sparse ridge (Kwon *et al.*, 2013),

$$\nabla\psi_\lambda(t) = \nabla\psi_\lambda^M(t)I\left(0 \leq t < \frac{a\lambda}{1+a\gamma}\right) + \nabla\psi_\gamma^R(t)I\left(t \geq \frac{a\lambda}{1+a\gamma}\right),$$

for $a > 2$ and $\gamma \geq 0$, moderately clipped LASSO (Kwon *et al.*, 2015),

$$\nabla\psi_\lambda(t) = \nabla\psi_\lambda^M(t)I(0 < t < a(\lambda-\gamma)) + \nabla\psi_\gamma^L(t)I(t \geq a(\lambda-\gamma)),$$

for $a > 1$ and $0 \leq \gamma \leq \lambda$, and mnet penalty (Huang *et al.*, 2016),

$$\nabla\psi_\lambda(t) = \nabla\psi_\lambda^M(t) + \nabla\psi_\gamma^R(t),$$

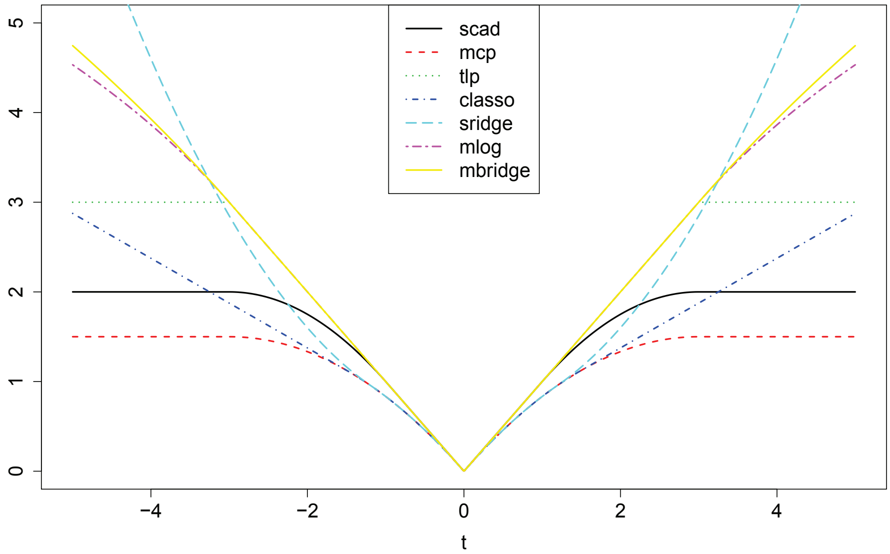

Figure 1: *Plot of various penalties with* $\lambda = 1, \gamma = 0.5$, *and* $a = 3$.

for $a > 2$ and $\gamma \geq 0$. Some non-convex penalties have infinite derivatives near the origin, that is, $\kappa_\lambda = \infty$ so that it cannot be cast into the class. Examples are the log (Zou and Li, 2008), bridge (Huang *et al.*, 2008) and h-likelihood (Lee and Oh, 2014) penalties defined as $\psi_\lambda(|t|) = \lambda \log|t|$, $\psi_\lambda(|t|) = \lambda\sqrt{|t|}$ and $\psi_\lambda(|t|) = \lambda \log|t| + \gamma|t|$, for $\gamma > 0$, respectively. For these penalties, Um *et al.* (2019) introduced a linear approximation near the origin

$$\nabla\psi_\lambda^a(t) = \nabla\psi_\lambda(a)I(0 < t < a) + \nabla\psi_\lambda(t)I(t \geq a),$$

for $a > 0$, so that ψ_λ^a satisfies (C1) and (C2).

Note that the h-likelihood penalty has more complex form than the one defined in this paper. However, it is sufficient to understand the h-likelihood penalty as a weighted sum of the log and LASSO penalties as described in Kwon *et al.* (2016). We put some plots in Figure 1 for graphical comparison of the penalties introduced.

3. Computational algorithm

In this section, we introduce an efficient algorithm for minimizing the penalized negative log-likelihood in (2.3). Since the objective function is non-convex, we first introduce the CCCP (Yuille and Rangarajan, 2002) and then apply local quadratic approximation (LQA) (Lee *et al.*, 2016), where the Hessian matrix is replaced with a fixed positive definite matrix (Böhning, 1992).

3.1. Concave-convex procedure

The CCCP is one of powerful optimization algorithms for minimizing non-convex functions that can be decomposed as a sum of convex and concave functions. Assume that $f = v + c$, where v is convex and c is concave and continuously differentiable. Given a current solution $\hat{x}$, the CCCP first defines a function $f^u(\cdot|\hat{x})$ that is an upper tight convex function of f by using local linear approximation of c

around $\hat{x}$:

$$f^u(x|\hat{x}) = v(x) + c(\hat{x}) + \nabla c(\hat{x})^T(x - \hat{x}),$$

where $\nabla c(x) = \partial c(x)/\partial x$. Then the iterative algorithm below is known to converge to a local minimizer of f (Yuille and Rangarajan, 2002) under some regularity conditions:

$$x^{s+1} = \arg\min_x f^u(x|x^s), \quad s \geq 1.$$

From (C2), we can see that $\psi_\lambda(|t|) = \kappa_\lambda|t| + \xi_\lambda(|t|)$, $t \in \mathbb{R}$. Hence, the penalized negative log-likelihood in (2.3) can be written as

$$\ell_\lambda(\pi) = \ell(\pi) + \sum_{k=1}^m \sum_{j=1}^p \kappa_\lambda|\pi_{kj}| + \sum_{k=1}^m \sum_{j=1}^p \xi_\lambda(|\pi_{kj}|),$$

where the first two terms are convex and the third term is concave and continuously differentiable. Hence the upper tight convex function, ignoring the constant, to be minimized becomes

$$\ell_\lambda^u(\pi|\hat{\pi}) = \ell(\pi) + \sum_{k=1}^m \sum_{j=1}^p \kappa_\lambda|\pi_{kj}| + \sum_{k=1}^m \sum_{j=1}^p \nabla\xi_\lambda(|\hat{\pi}_{kj}|)\pi_{kj},$$

given an initial solution $\hat{\pi}$. To sum up, we can obtain an iterative algorithm that converges to a local minimizer of ℓ_λ as follows:

$$\pi^{s+1} = \arg\min_{pi} \ell_\lambda^u(\pi|\pi^s), \quad s \geq 1. \tag{3.1}$$

Note that the algorithm in (3.1) iteratively solves LASSO penalized convex objective functions which is an important advantage from the CCCP.

3.2. Local quadratic approximation

The algorithm (3.1) includes minimizing ℓ_λ^u, which can be done by using the LQA algorithm (Lee *et al.*, 2016). The LQA first defines a function that locally majorizes ℓ around an initial solution $\tilde{\pi}$:

$$\ell^q(\pi|\tilde{\pi}) = \ell(\tilde{\pi}) + \nabla\ell(\tilde{\pi})^T(\pi - \tilde{\pi}) + \frac{(\pi - \tilde{\pi})^T\nabla^2\ell(\tilde{\pi})(\pi - \tilde{\pi})}{2},$$

where $\nabla\ell(\pi) = \partial\ell(\pi)/\partial\pi$ and $\nabla^2\ell(\pi) = \partial^2\ell(\pi)/\partial\pi^2$. Then we have an iterative algorithm for minimizing ℓ_λ^u in (3.1), using $\ell^q(\cdot|\tilde{\pi})$ instead of ℓ for given π^s:

$$\pi^{t+1,s} = \arg\min_\pi \ell_\lambda^{u,q}\left(\pi|\pi^{t,s}, \pi^s\right), \quad t \geq 1, \tag{3.2}$$

where

$$\ell_\lambda^{u,q}\left(\pi|\pi^{t,s}, \pi^s\right) = \ell^q\left(\pi|\pi^{t,s}\right) + \sum_{k=1}^m \sum_{j=1}^p \kappa_\lambda|\pi_{kj}| + \sum_{k=1}^m \sum_{j=1}^p \nabla\xi_\lambda\left(\left|\pi_{kj}^s\right|\right)\pi_{kj}.$$

Note that the objective function $\ell_\lambda^{u,q}$ in (3.2) is a LASSO penalized quadratic function with tuning parameter κ_λ so that we may use many existing algorithms for the LASSO such as the coordinate descent (CD) algorithm developed by Friedman *et al.* (2010).

3.3. Uniform bound of the Hessian

The computational time of the algorithm in (3.2) can be significantly slow since we repeatedly calculate $mp \times mp$ dimensional Hessian matrix $\nabla^2 \ell(\pi)$ for the multinomial logistic regression:

$$\nabla^2 \ell(\pi) = \sum_{i=1}^{n} \left(\Lambda_i - p_i p_i^T\right) \otimes \left(x_i x_i^T\right),$$

where $p_i = (p_{i1}, \ldots, p_{im})^T$, Λ_i is the diagonal matrix with elements in p_i and $\otimes$ is Kronecker matrix product. Note that the Hessian matrix is uniformly bounded (Böhning, 1992) as follows:

$$\nabla^2 \ell(\pi) \le Q, \quad \forall \pi,$$

where $A \le B$ implies that $B - A$ is positive definite, $Q = \sum_{i=1}^{n} \{I_m - 1_m 1_m^T/(m+1)\} \otimes x_i x_i^T/2$, I_m is $m \times m$ identity matrix and 1_m is the vector of length m whose elements are all 1. Hence we can save the computational time by using Q instead of the Hessian matrix for all the iteration steps:

$$\pi^{t+1,s} = \arg\min_{\pi} \ell_{\lambda}^{u,q}\left(\pi | \pi^{t,s}, \pi^{s}\right), \quad t \ge 1, \tag{3.3}$$

where

$$\ell_{\lambda}^{u,q}\left(\pi | \pi^{t,s}, \pi^{s}\right) = \frac{\pi^T Q \pi}{2} + L_{t,s}^T \pi + \sum_{k=1}^{m} \sum_{j=1}^{p} \kappa_{\lambda} |\pi_{kj}| + \sum_{k=1}^{m} \sum_{j=1}^{p} \nabla \xi_{\lambda}\left(\left|\pi_{kj}^{s}\right|\right) \pi_{kj},$$

and $L_{t,s} = \nabla \ell(\pi^{t,s}) - Q\pi^{t,s}$.

3.4. CCCP-UBQA-CD algorithm

The two core algorithms in (3.1) and (3.3) for minimizing ℓ_λ as follows:

CCCP-UBQA algorithm for minimizing ℓ_λ

- (CCCP) Set an initial $\hat{\pi}$ and update $\hat{\pi}$ with $\tilde{\pi}$ obtained by UBQA until $\hat{\pi}$ converges.
- (UBQA) Set an initial $\tilde{\pi}$ and update $\tilde{\pi}$ with $\check{\pi}$ below until $\tilde{\pi}$ converges:

$$\check{\pi} = \arg\min_{\pi} \ell_{\lambda}^{u,q}(\pi | \hat{\pi}, \tilde{\pi}).$$

We finish the section giving the solution $\check{\pi}$ in UBQA step explicitly by applying the CD algorithm in Friedman *et al.* (2010), which is used for ncpen. Let $\alpha_{kj} = (k-1)p + j$, $k \le m$, $j \le p$ be the parameter index of π. Let $Q_{kj,}$ be the α_{kj}th column vector of Q, $Q_{kj,kj}$ the α_{kj}^{th} entry of Q_{kj} and $Q_{kj,-kj}$ be the vector obtained by deleting $Q_{kj,kj}$ from $Q_{kj,}$. Similarly let π_{kj} and $\nabla_{kj}\ell(\hat{\pi})$ be the α_{kj}^{th} entry of π and $\nabla\ell(\hat{\pi})$, respectively and π_{-kj} be the vector obtained by deleting π_{kj} from π. The CD algorithm sequentially minimizes coordinate functions of $\ell_{\lambda}^{u,q}(\pi|\hat{\pi}, \tilde{\pi})$, where the α_{kj}th coordinate function becomes

$$\ell_{\lambda,kj}^{u,q}(\pi_{kj}) = \left(\frac{Q_{kj,kj}}{2}\right) \pi_{kj}^2 + \left\{Q_{kj,-kj}^T \pi_{-kj} + \nabla_{kj}\ell(\tilde{\pi}) - Q_{kj}^T \tilde{\pi} + \nabla\psi_{\lambda}(|\hat{\pi}_{kj}|)\right\} \pi_{kj} + \kappa_{\lambda} |\pi_{kj}|,$$

as a function of π_{kj} only. Then the minimizer of $\ell^{u,q}_{\lambda,kj}(\pi_{kj})$ becomes (Friedman *et al.*, 2010)

$$\breve{\pi}^{\kappa_\lambda}_{kj} = \operatorname{sign}\left(\breve{\pi}^{o}_{kj}\right)\left(\left|\breve{\pi}^{o}_{kj}\right| - \frac{\kappa_\lambda}{Q_{kj,kj}}\right)_+, \tag{3.4}$$

where $x_+ = xI(x > 0)$ and

$$\breve{\pi}^{o}_{kj} = -\frac{Q^T_{kj,-kj}\pi_{-kj} + \nabla_{kj}\ell(\tilde{\pi}) - Q^T_{kj}\tilde{\pi} + \nabla\psi_\lambda\left(\left|\hat{\pi}_{kj}\right|\right)}{Q_{kj,kj}}.$$

Now, the CCCP-UBQA algorithm applied with the CD algorithm becomes as follows:

CCCP-UBQA-CD algorithm for minimizing ℓ_λ

- (CCCP) Set an initial $\hat{\pi}$ and update $\hat{\pi}$ with $\tilde{\pi}$ obtained by UBQA until $\hat{\pi}$ converges.
- (UBQA) Set an initial $\tilde{\pi}$ and update $\tilde{\pi}$ with $\breve{\pi}$ obtained by CD until $\tilde{\pi}$ converges.
- (CD) Set an initial $\breve{\pi}$ and update coordinates of $\breve{\pi}$ as below until $\breve{\pi}$ converges:

$$\tilde{\pi}_{kj} = \breve{\pi}^{\kappa_\lambda}_{kj}, \quad k \le m,\ j \le p.$$

Note that an immediate and reasonable initial solution for the UBQA and CD steps are $\tilde{\pi} = \hat{\pi}$ and $\breve{\pi} = \tilde{\pi}$, respectively. Based on our empirical experience, we found that the choice greatly enhances the computational time and makes the algorithm more stable compared with the trivial cases when $\tilde{\pi} = \breve{\pi} = 0$.

4. Numerical studies

In this section, we present results from numerical studies via simulations and data analysis. We investigate the finite sample performance of the penalized multinomial logistic regression. We compare the SCAD, moderately clipped LASSO and modified bridge penalties with the LASSO penalty for illustration, which are denoted by lasso, scad, classo and mbridge in the tables. The non-convex penalized estimators are obtained by R package ncpen and the LASSO is obtained by R package glmnet. Tuning parameters are obtained by using the Bayesian information criterion (BIC) or generalized information criterion (GIC).

4.1. Simulation studies for finite sample performance

We generate n simulated samples from model (2.1), where $x_i \sim N(0_p, \Sigma)$, $i \le n$ with $\Sigma_{jj'} = \rho^{|j-j'|}$, $j, j' \le p$. We set $\pi_{kj} = 2/\sqrt{j}I(k \le m, j \le q)$ for the true regression coefficients. We consider $n \in \{200, 400, 800\}$, $m \in \{3, 5\}$, $p \in \{10, 100\}$, $q = 5$, and $\rho = 0.5$. We measure selection performance by using the sensitivity, specificity and accuracy defined by $|\hat{\mathcal{S}} \cap \mathcal{S}|/|\mathcal{S}|$, $|\hat{\mathcal{S}}^c \cap \mathcal{S}^c|/|\mathcal{S}^c|$, and $I(\mathcal{S} = \hat{\mathcal{S}})$, where $\hat{\mathcal{S}} = \{(k, j) : \hat{\pi}_{kj} \ne 0\}$ and $\mathcal{S} = \{(k, j) : \pi_{kj} \ne 0\}$, and the prediction error based on 1,000 independent test samples.

We repeat the simulation 100 times and present the averages of the measures in Tables 1 and 2. For comparison we also consider the oracle estimator obtained by using signal variables only as well as the ordinary non-penalized estimator available only when $n \ge mp$. We summarize some observations from the simulations. The LASSO is the best for the sensitivity but the worst for the specificity in

Table 1: Simulation results for the selection

k	p	n	Sensitivity					
			oracle	ordinary	lasso	scad	classo	mbridge
3	10	200	1	1	0.96	0.758	0.718	0.754
		400	1	1	0.99	0.896	0.904	0.902
		800	1	1	1	0.996	0.99	0.996
		1600	1	1	1	1	1	1
	100	200	1	0	0.552	0.522	0.566	0.568
		400	1	1	0.936	0.79	0.832	0.85
		800	1	1	0.996	0.932	0.972	0.932
		1600	1	1	1	0.992	1	1
5	10	200	1	1	0.828	0.613	0.583	0.617
		400	1	1	0.962	0.76	0.711	0.788
		800	1	1	0.998	0.951	0.926	0.943
		1600	1	1	1	1	0.998	0.997
	100	200	1	0	0.002	0.278	0.274	0.305
		400	1	0	0.063	0.452	0.479	0.499
		800	1	1	0.878	0.723	0.764	0.766
		1600	1	1	0.983	0.864	0.941	0.903
k	p	n	Specificity					
			oracle	ordinary	lasso	scad	classo	mbridge
3	10	200	1	0	0.726	0.928	0.962	0.946
		400	1	0	0.726	0.928	0.952	0.95
		800	1	0	0.816	0.968	0.984	0.974
		1600	1	0	0.9	0.988	0.996	0.99
	100	200	1	1	0.996	0.99	0.992	0.989
		400	1	0	0.974	0.969	0.975	0.968
		800	1	0	0.966	0.975	0.986	0.981
		1600	1	0	0.962	0.992	0.996	0.992
5	10	200	1	0	0.755	0.913	0.935	0.92
		400	1	0	0.705	0.921	0.947	0.921
		800	1	0	0.718	0.906	0.948	0.926
		1600	1	0	0.757	0.94	0.963	0.951
	100	200	1	1	1	0.994	0.995	0.995
		400	1	1	1	0.995	0.995	0.995
		800	1	0	0.983	0.984	0.983	0.983
		1600	1	0	0.98	0.981	0.985	0.979
k	p	n	Accuracy					
			oracle	ordinary	lasso	scad	classo	mbridge
3	10	200	1	0	0	0.02	0.04	0.06
		400	1	0	0.02	0.12	0.16	0.18
		800	1	0	0.16	0.72	0.8	0.78
		1600	1	0	0.42	0.88	0.96	0.90
	100	200	1	0	0	0	0	0
		400	1	0	0	0	0	0
		800	1	0	0	0.02	0.06	0.04
		1600	1	0	0	0.30	0.48	0.26
5	10	200	1	0	0	0	0	0
		400	1	0	0	0	0	0
		800	1	0	0	0.06	0.16	0.10
		1600	1	0	0	0.32	0.52	0.40
	100	200	1	0	0	0	0	0
		400	1	0	0	0	0	0
		800	1	0	0	0	0	0
		1600	1	0	0	0	0.02	0

Table 2: Simulation results for the prediction

k	p	n	Prediction error					
			oracle	ordinary	lasso	scad	classo	mbridge
3	10	200	0.369	0.376	0.376	0.383	0.385	0.382
		400	0.365	0.370	0.368	0.369	0.370	0.368
		800	0.362	0.363	0.364	0.362	0.363	0.362
		1600	0.362	0.363	0.365	0.362	0.362	0.362
	100	200	0.366	0.717	0.441	0.406	0.397	0.399
		400	0.367	0.491	0.381	0.382	0.377	0.378
		800	0.361	0.390	0.367	0.366	0.364	0.364
		1600	0.364	0.380	0.367	0.364	0.363	0.363
5	10	200	0.523	0.529	0.535	0.539	0.535	0.537
		400	0.518	0.522	0.527	0.529	0.53	0.528
		800	0.514	0.516	0.522	0.516	0.517	0.516
		1600	0.515	0.516	0.52	0.515	0.515	0.516
	100	200	0.520	0.845	0.615	0.571	0.567	0.565
		400	0.520	0.847	0.615	0.549	0.542	0.544
		800	0.514	0.550	0.551	0.528	0.524	0.523
		1600	0.517	0.531	0.539	0.518	0.517	0.520

Table 3: Simulation results for the computation time in seconds

k	p	n	scad	classo	mbridge	k	p	n	scad	classo	mbridge
3	10	200	0.290	0.228	0.292	5	10	200	2.567	2.248	2.474
		400	1.221	1.059	1.329			400	9.904	8.927	9.968
		800	4.321	3.909	5.107			800	35.505	32.863	38.414
		1600	15.500	14.219	19.486			1600	112.720	116.600	119.740
	100	200	0.656	0.559	0.574		100	200	2.204	2.201	1.938
		400	2.682	2.362	2.320			400	8.601	8.359	7.988
		800	9.814	8.873	9.265			800	33.290	34.680	32.099
		1600	26.414	25.002	29.977			1600	135.88	187.800	142.320

most cases, which empirically shows that the LASSO tends to overfit the true model. The sensitivity and specificity for the non-convex penalties are increasing to 1 which empirically supports the oracle property studied in other papers (Fan and Li, 2001; Fan and Peng, 2004; Kwon and Kim, 2012). The accuracy for the non-convex penalties is increasing although it is very small when m and p are large. We believe that the accuracy will approach to 1 in this case also if we increase the sample size to be enough. The prediction accuracy of the non-convex penalized estimators become better than that of the LASSO as the sample size increases. The GIC may not guarantee the best prediction performance of the LASSO so that the prediction results in this simulation should be interpreted carefully. For the readers, we report the averages of the computational times for the simulations in Table 3.

4.2. Data examples

We apply the penalized multinomial regression for the 'zoo' sample that is available from the UCI machine learning repository. The sample includes $n = 101$ observations with 16 covariates (hair, feathers, eggs, milk, airborne, aquatic, predator, toothed, backbone, breathes, venomous, fins, legs*, tail, domestic, and catsize) and the type of animals are labeled from 1 to 7. All the covariates are Boolean except for legs that is ranged from 0 to 8. Visit 'https://archive.ics.uci.edu/ml/datasets/zoo' for a detailed description of the variables.

The estimated regression coefficients are listed in Table 4, where the variables with zero coefficients for all methods are deleted. All the penalized estimators share the same variables with non-zero

Table 4: Estimated coefficients of zoo sample

	Class1				Class2			
	lasso	scad	classo	mbridge	lasso	scad	classo	mbridge
intercept	−1.0925	−15.4251	−1.0926	−5.2049	−1.4835	−14.4535	−1.4835	−5.3949
feathers	0	0	0	0	5.0542	31.5816	5.0542	15.7573
milk	4.9497	37.6148	4.9498	15.336	0	0	0	0
airborne	0	0	0	0	0	0	0	0
fins	0	0	0	0	0	0	0	0
	Class3				Class4			
	lasso	scad	classo	mbridge	lasso	scad	classo	mbridge
intercept	−0.6931	−0.6931	−0.6931	−0.6931	−1.5134	−7.5558	−1.5134	−5.2502
feathers	0	0	0	0	0	0	0	0
milk	0	0	0	0	0	0	0	0
airborne	0	0	0	0	0	0	0	0
fins	0	0	0	0	3.9357	16.6094	3.9357	10.8944
	Class5				Class6			
	lasso	scad	classo	mbridge	lasso	scad	classo	mbridge
intercept	−0.9163	−0.9163	−0.9163	−0.9163	−0.6472	−0.2231	−0.6472	−1.5649
feathers	0	0	0	0	0	0	0	0
milk	0	0	0	0	0	0	0	0
airborne	0	0	0	0	1.4255	0	1.4255	6.0378
fins	0	0	0	0	0	0	0	0

Table 5: Number of wrong classifications of zoo sample

ordinary	lasso	scad	classo	mbridge
5	11	18	11	11

regression coefficients for each class but the effect size is different. We calculate the leave-one-out errors and summarize the results in Table 5. The ordinary non-penalized estimator performs the best and the SCAD is the worst. However we note that the number of variables used for the classification is only 4, which can be an advantage from penalized estimation.

5. Concluding remarks

We introduced the CCCP-UBQA algorithm for the non-convex penalized multinomial logistic regression which can cover most non-convex penalties. The algorithm implemented in R package ncpen is stable and fast enough to be used for academic purposes. However, we also found that the algorithm rapidly becomes slow when m and p are large. For this issue, we have two strategies for enhancing the computational speed, which can be a future study regarding the algorithm. The CCCP-UBQA includes two iterative algorithms which cause a computational burden. Based on the authors' experience, we can collapse these two iterative algorithms into one. The idea is approximating the penalty and likelihood simultaneously at the current solution. Further, we may not fully iterate the UBQA steps for the convergence which often reduces the computational time. We did not use these two methods in ncpen since the convergence of the methods should be carefully investigated.

Acknowledgements

This paper was written as part of Konkuk University's research support program for its faculty on sabbatical leave in 2018 and Chungnam National University fund.

References

Böhning D (1992). Multinomial logistic regression algorithm, *Annals of the Institute of Statistical Mathematics*, **44**, 197–200.

Bondell HD, Krishna A, and Ghosh SK (2010). Joint variable selection for fixed and random effects in linear mixed-effects models, *Biometrics*, **66**, 1069–1077.

Cawley GC, Talbot NL, and Girolami M (2007). Sparse multinomial logistic regression via Bayesian l1 regularisation. In *Advances in Neural Information Processing Systems*, 209–216.

Chen L, Yang J, Li J, and Wang X (2014). Multinomial regression with elastic net penalty and its grouping effect in gene selection. In *Abstract and Applied Analysis*, **2014**, Hindawi.

Fan J and Li R (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American statistical Association*, **96**, 1348–1360.

Fan J and Lv J (2011). Nonconcave penalized likelihood with np-dimensionality, *IEEE Transactions on Information Theory*, **57**, 5467–5484.

Fan J and Peng H (2004). Nonconcave penalized likelihood with a diverging number of parameters, *The Annals of Statistics*, **32**, 928–961.

Friedman J, Hastie T, and Tibshirani R (2010). Regularization paths for generalized linear models via coordinate descent, *Journal of Statistical Software*, **33**, 1.

Hoerl AE and Kennard RW (1970). Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics*, **12**, 55–67.

Huang J, Breheny P, Lee S, Ma S, and Zhang CH (2016). The Mnet method for variable selection, *Statistica Sinica*, **26**, 903–923.

Huang J, Horowitz JL, and Ma S (2008). Asymptotic properties of bridge estimators in sparse high-dimensional regression models, *The Annals of Statistics*, **36**, 587–613.

Huang J, Horowitz JL, and Wei F (2010). Variable selection in nonparametric additive models, *The Annals of Statistics*, **38**, 2282–2313.

Huttunen H, Yancheshmeh FS, and Chen K (2016). Car type recognition with deep neural networks. *In 2016 IEEE Intelligent Vehicles Symposium (IV)*, 1115–1120, IEEE.

Kim J, Kim Y, and Kim Y (2008). A gradient-based optimization algorithm for lasso, *Journal of Computational and Graphical Statistics*, **17**, 994–1009.

Kim Y, Kwon S, and Song SH (2006). Multiclass sparse logistic regression for classification of multiple cancer types using gene expression data, *Computational Statistics & Data Analysis*, **51**, 1643–1655.

Krishnapuram B, Carin L, Figueiredo MA, and Hartemink AJ (2005). Sparse multinomial logistic regression: Fast algorithms and generalization bounds, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 957–968.

Kwon S and Kim Y (2012). Large sample properties of the SCAD-penalized maximum likelihood estimation on high dimensions, *Statistica Sinica*, **12**, 629–653.

Kwon S, Kim Y, and Choi H (2013). Sparse bridge estimation with a diverging number of parameters, *Statistics and Its Interface*, **6**, 231–242.

Kwon S, Lee S, and Kim Y (2015). Moderately clipped lasso, *Computational Statistics & Data Analysis*, **92**, 53–67.

Kwon S, Oh S, and Lee Y (2016). The use of random-effect models for high-dimensional variable selection problems, *Computational Statistics & Data Analysis*, **103**, 401–412.

Lee S, Kwon S, and Kim Y (2016). A modified local quadratic approximation algorithm for penalized optimization problems, *Computational Statistics & Data Analysis*, **94(C)**, 275–286.

Lee Y and Oh HS (2014). A new sparse variable selection via random-effect model, *Journal of Multivariate Analysis*, **125**, 89–99.

Shen X, Pan W, and Zhu Y (2012). Likelihood-based selection and sharp parameter estimation, *Journal of the American Statistical Association*, **107**, 223–232.

Simon N, Friedman J, and Hastie T (2013). A blockwise descent algorithm for group-penalized multiresponse and multinomial regression. arXiv preprint arXiv:1311.6529.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society: Series B (Methodological)*, **58**, 267–288.

Tutz G (2011). *Regression for categorical data*, volume 34. Cambridge University Press.

Tutz G, Pößnecker W, and Uhlmann L (2015). Variable selection in general multinomial logit models, *Computational Statistics & Data Analysis*, **82**, 207–222.

Um S, Kim D, Lee S, and Kwon S (2019). On the strong oracle property of concave penalized estimators with infinite penalty derivative at the origin, *The Korean Journal of Statistics*, Under review.

Xie H and Huang J (2009). SCAD-penalized regression in high-dimensional partially linear models, *The Annals of Statistics*, **37**, 673–696.

Yuan M and Lin Y (2006). Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**, 49–67.

Yuille AL and Rangarajan A (2002). The concave-convex procedure (CCCP). In *Advances in Neural Information Processing Systems*, 1033–1040.

Zhang CH (2010). Nearly unbiased variable selection under minimax concave penalty, *The Annals of Statistics*, **38**, 894–942.

Zhang CH and Zhang T (2012). A general theory of concave regularization for high-dimensional sparse estimation problems, *Statistical Science*, **27**, 576–593.

Zhao P and Yu B (2006). On model selection consistency of lasso, *Journal of Machine Learning Research*, **7**, 2541–2563.

Zhu J and Hastie T (2004). Classification of gene microarrays by penalized logistic regression, *Biostatistics*, **5**, 427–443.

Zou H and Hastie T (2005). Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67**, 301–320.

Zou H and Li R (2008). One-step sparse estimates in nonconcave penalized likelihood models, *Annals of Statistics*, **36**, 1509–1533.

Received October 17, 2019; Revised November 26, 2019; Accepted November 26, 2019