Entropy weighted regularisation, a general way to debias regularisation penalties.

Olof Zetterqvist

October 2021

Setting

- We will linear models of the form $Y = X\beta + \epsilon$ where $\epsilon \in N(0, \sigma^2 \mathrm{I})$.

Setting

- We will linear models of the form $Y = X\beta + \epsilon$ where $\epsilon \in N(0, \sigma^2 \mathrm{I})$.
- To find an estimation of the parameters β we will solve the minimisation problem

$$\tilde{\beta} = \arg\min_{\beta} \frac{1}{2} || Y - X\beta ||_2^2$$

Setting

- We will linear models of the form $Y = X\beta + \epsilon$ where $\epsilon \in N(0, \sigma^2 \mathrm{I})$.
- To find an estimation of the parameters β we will solve the minimisation problem

$$\tilde{\beta} = \arg\min_{\beta} \frac{1}{2} || Y - X\beta ||_2^2 + g_\lambda(\beta, \phi)$$

- $g_\lambda(\beta; \phi)$ is a regularisation penalty that reduces overfitting by reducing the amount of variance in the model.

Setting

- We will linear models of the form $Y = X\beta + \epsilon$ where $\epsilon \in N(0, \sigma^2 I)$.
- To find an estimation of the parameters β we will solve the minimisation problem

$$\tilde{\beta} = \arg\min_{\beta} \frac{1}{2} ||Y - X\beta||_2^2 + g_\lambda(\beta, \phi)$$

- $g_\lambda(\beta; \phi)$ is a regularisation penalty that reduces overfitting by reducing the amount of variance in the model.

$$E_\beta\left[(\tilde{\beta} - \beta^*)^2\right] = E_\beta[(\tilde{\beta} - E_\beta[\tilde{\beta}])^2] + (E_\beta[\tilde{\beta}] - \beta^*)^2 = Var(\tilde{\beta}) + Bias(\tilde{\beta})^2.$$

Properties of a good regularisation technique

- **Unbiasedness.** Parameter estimates should be unbiased or very close to unbiased, in particular for parameters that are far from zero.

Properties of a good regularisation technique

- **Unbiasedness.** Parameter estimates should be unbiased or very close to unbiased, in particular for parameters that are far from zero.
- **Consistency** of the estimated parameters; the parameter estimator converges in probability to the true parameter as $n \to \infty$.

Properties of a good regularisation technique

- **Unbiasedness.** Parameter estimates should be unbiased or very close to unbiased, in particular for parameters that are far from zero.
- **Consistency** of the estimated parameters; the parameter estimator converges in probability to the true parameter as $n \to \infty$.
- **Continuity.** Parameter estimates should be continuous as functions of the data.

Properties of a good regularisation technique

- **Unbiasedness.** Parameter estimates should be unbiased or very close to unbiased, in particular for parameters that are far from zero.
- **Consistency** of the estimated parameters; the parameter estimator converges in probability to the true parameter as $n \to \infty$.
- **Continuity.** Parameter estimates should be continuous as functions of the data.

When the purpose of estimation is also variable selection, the following can be added.

Properties of a good regularisation technique

- **Unbiasedness.** Parameter estimates should be unbiased or very close to unbiased, in particular for parameters that are far from zero.
- **Consistency** of the estimated parameters; the parameter estimator converges in probability to the true parameter as $n \to \infty$.
- **Continuity.** Parameter estimates should be continuous as functions of the data.

When the purpose of estimation is also variable selection, the following can be added.

- **Sparsity.** This means that (as many as possible of the) parameters whose true value is zero should be estimated as zero.

Properties of a good regularisation technique

- **Unbiasedness.** Parameter estimates should be unbiased or very close to unbiased, in particular for parameters that are far from zero.
- **Consistency** of the estimated parameters; the parameter estimator converges in probability to the true parameter as $n \to \infty$.
- **Continuity.** Parameter estimates should be continuous as functions of the data.

When the purpose of estimation is also variable selection, the following can be added.

- **Sparsity.** This means that (as many as possible of the) parameters whose true value is zero should be estimated as zero.
- **Sign consistency.** With probability converging to one in the number of observations, all parameter estimates have the same sign as the true parameter (where $sign(0) = 0$).

Shortly some common methods

- Add a penalty $g_\lambda(\beta)$ to the loss function $\sum_i \frac{1}{2}||Y - X\beta||_2^2 + g_\lambda(\beta)$.

Shortly some common methods

- Add a penalty $g_\lambda(\beta)$ to the loss function $\sum_i \frac{1}{2}||Y - X\beta||_2^2 + g_\lambda(\beta)$.
- **OLS**: $g_\lambda(\beta) = 0$
- **Lasso**: $g_\lambda(\beta) = \lambda \sum_i |\beta_i|$
- **Ridge**: $g_\lambda(\beta) = \lambda \sum_i \beta_i^2$
- **Bridge**: $g_\lambda(\beta;\gamma) = \lambda \sum_i |\beta_i|^\gamma$; $\gamma > 0$
- **SCAD**: $g_\lambda(\beta;a) = \sum_i \left[\mathbf{1}(|\beta_i| < \lambda)\lambda|\beta_i| + \mathbf{1}(\lambda \le |\beta_i| \le a\lambda)\frac{|\beta_i|^2 - 2a\lambda|\beta_i| + \lambda^2}{2(a-1)} + \mathbf{1}(|\beta_i| > a\lambda)\frac{(a+1)\lambda^2}{2} \right]$; $a > 1$.
- **Adaptive lasso**: $g_\lambda(\beta;\gamma) = \lambda \sum_i \omega_i|\beta_i|$; where ω_i is weights based on a previous estimate $\hat{\beta}$. One example is $\omega_i = \frac{1}{|\hat{\beta}_i|^\gamma}$; $\gamma > 0$

Shortly some common methods

	Unbiased	Consistency	Continuity	Sparsity	Sign consistency
OLS	Yes	Yes	Yes	No	No
Lasso	No	Yes	Yes	Yes	Yes
Ridge	No	Yes	Yes	No	No
Bridge	When $\gamma < 1$	Yes	When $\gamma \geq 1$	When $\gamma \leq 1$	When $\gamma \leq 1$
SCAD	Yes	Yes	Yes	Yes	Yes
Adaptive lasso	Yes	Yes	Yes	Yes	Yes

Table: A table of which of the common approaches fulfills the requested properties.

Idea behind our method

- $\tilde{\beta} = \arg\min_{\beta} \frac{1}{2}||Y - X\beta||^2 + \sum_k^p g_\lambda(\beta_k; \phi)$

Idea behind our method

- $\tilde{\beta} = \arg\min_\beta \frac{1}{2}||Y - X\beta||^2 + \sum_k^p g_\lambda(\beta_k; \phi)$
- $\tilde{\beta}, \tilde{\mathrm{u}} = \arg\min_{\beta,\mathrm{u}} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k; \phi) + \tilde{g}_\gamma(u)$

Idea behind our method

- $\tilde{\beta} = \arg\min_\beta \frac{1}{2}||Y - X\beta||^2 + \sum_k^p g_\lambda(\beta_k; \phi)$
- $\tilde{\beta}, \tilde{\mathrm{u}} = \arg\min_{\beta,\mathrm{u}} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k; \phi) + \tilde{g}_\gamma(u)$
- We have considered the function g to be a regular lasso, $g = ||\cdot||_1$, and ridge $g = \frac{1}{2}||\cdot||_2^2$. These will give the methods *Entropy Weighted Lasso* (EWL) and *Entropy Weighted Ridge* (EWR)

Idea behind our method

- $\tilde{\beta} = \arg\min_\beta \frac{1}{2}||Y - X\beta||^2 + \sum_k^p g_\lambda(\beta_k;\phi)$
- $\tilde{\beta}, \tilde{\mathrm{u}} = \arg\min_{\beta,\mathrm{u}} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k;\phi) + \tilde{g}_\gamma(u)$
- We have considered the function g to be a regular lasso, $g = ||\cdot||_1$, and ridge $g = \frac{1}{2}||\cdot||_2^2$. These will give the methods *Entropy Weighted Lasso* (EWL) and *Entropy Weighted Ridge* (EWR)
- $\tilde{g}_\gamma$ is considered to be $\tilde{g}_\gamma(u) = \sum_i \gamma(u_i \log u_i - u_i + 1)$

Idea behind our method

- $\tilde{\beta} = \arg\min_{\beta} \frac{1}{2}||Y - X\beta||^2 + \sum_k^p g_\lambda(\beta_k; \phi)$
- $\tilde{\beta}, \tilde{\mathrm{u}} = \arg\min_{\beta,\mathrm{u}} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k; \phi) + \tilde{g}_\gamma(u)$
- We have considered the function g to be a regular lasso, $g = ||\cdot||_1$, and ridge $g = \frac{1}{2}||\cdot||_2^2$. These will give the methods *Entropy Weighted Lasso* (EWL) and *Entropy Weighted Ridge* (EWR)
- $\tilde{g}_\gamma$ is considered to be $\tilde{g}_\gamma(u) = \sum_i \gamma(u_i \log u_i - u_i + 1)$
- Putting it all together gives us an optimisation problem of the form

$$\tilde{\beta}, \tilde{\mathrm{u}} = \arg\min_{\beta,u} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_{k=1}^{p} u_k g_\lambda(\beta_k; \phi) +$$

$\sum_{k=1}^p \gamma(u_i \log u_i - u_i + 1)$

Simplification of expression

- $\tilde{\beta}, \tilde{\mathrm{u}} = \arg\min_{\beta,u} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k; \phi) + \sum_i \gamma(u_i \log u_i - u_i + 1)$

Simplification of expression

- $\tilde{\beta}, \tilde{u} = \arg\min_{\beta,u} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k;\phi) + \sum_i \gamma(u_i \log u_i - u_i + 1)$
- Minimizing with respect to u gives us $u_k = e^{-\frac{1}{\gamma}g_\lambda(\beta_k;\phi)}$

Simplification of expression

- $\tilde{\beta}, \tilde{\text{u}} = \arg\min_{\beta,u} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k;\phi) + \sum_i \gamma(u_i \log u_i - u_i + 1)$
- Minimizing with respect to u gives us $u_k = e^{-\frac{1}{\gamma} g_\lambda(\beta_k;\phi)}$
- Putting this into our original minimization problem gives us that $\tilde{\beta} = \arg\min_\beta \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p \gamma(1 - e^{-\frac{1}{\gamma} g_\lambda(\beta_k;\phi)})$

Simplification of expression

- $\tilde{\beta}, \tilde{u} = \arg\min_{\beta,u} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k;\phi) + \sum_i \gamma(u_i \log u_i - u_i + 1)$
- Minimizing with respect to u gives us $u_k = e^{-\frac{1}{\gamma} g_\lambda(\beta_k;\phi)}$
- Putting this into our original minimization problem gives us that $\tilde{\beta} = \arg\min_\beta \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p \gamma(1 - e^{-\frac{1}{\gamma} g_\lambda(\beta_k;\phi)})$
- $g_\lambda(\beta_k;\phi) = \lambda \sum |\beta| \Rightarrow \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_k|})$ (EWL)
- $g_\lambda(\beta_k;\phi) = \lambda \sum \frac{1}{2}\beta_k^2 \Rightarrow \gamma(1 - e^{-\frac{\lambda}{2\gamma}\beta_k^2})$ (EWR)

Simplification of expression

- $\tilde{\beta}, \tilde{u} = \arg\min_{\beta,u} \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p u_k g_\lambda(\beta_k;\phi) + \sum_i \gamma(u_i \log u_i - u_i + 1)$
- Minimizing with respect to u gives us $u_k = e^{-\frac{1}{\gamma} g_\lambda(\beta_k;\phi)}$
- Putting this into our original minimization problem gives us that $\tilde{\beta} = \arg\min_\beta \sum_i \frac{1}{2}||Y - X\beta||_2^2 + \sum_k^p \gamma(1 - e^{-\frac{1}{\gamma} g_\lambda(\beta_k;\phi)})$
- $g_\lambda(\beta_k;\phi) = \lambda \sum |\beta| \Rightarrow \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_k|})$ (EWL)
- $g_\lambda(\beta_k;\phi) = \lambda \sum \frac{1}{2}\beta_k^2 \Rightarrow \gamma(1 - e^{-\frac{\lambda}{2\gamma}\beta_k^2})$ (EWR)
- Some interesting observations
  - Letting $\gamma \to \infty \Rightarrow \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_k|^j}) \to \lambda|\beta_k|^j$
  - Letting $\frac{\lambda}{\gamma} \to \infty \Rightarrow \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_k|^j}) \to \gamma 1_{\beta_k \neq 0}$

Some interesting theoretical results

Theorem

Let s_1^2 be the smallest eigenvalue of X^TX. The minimisation problem

$$\tilde{\beta} = \min_{\beta} \frac{1}{2}||Y - X\beta||_2^2 + \gamma \sum_{j=1}^{p} (1 - e^{-\frac{\lambda}{\gamma}|\beta_j|})$$

is convex whenever $\gamma > \frac{\lambda^2}{s_1^2}$.

Some interesting theoretical results

Theorem

Let s_1^2 be the smallest eigenvalue of X^TX. The minimisation problem

$$\tilde{\beta} = \min_{\beta} \frac{1}{2}||Y - X\beta||_2^2 + \gamma \sum_{j=1}^{p} (1 - e^{-\frac{\lambda}{\gamma}|\beta_j|})$$

is convex whenever $\gamma > \frac{\lambda^2}{s_1^2}$.

outline of proof: Within each orthant we study the hessian

$$b^T \frac{\partial^2 L(\beta)}{\partial \beta^2} b \geq b^T \left(X^T X - \frac{\lambda^2}{\gamma} \mathrm{I} \right) b$$

which is positive if $\gamma > \frac{\lambda^2}{s_1^2}$. We then combine all orthants by noticing that we can expand the domain of each orthant and look at the maximum over convex functions.

Theorem

Assume $\frac{\gamma_n}{n} \to \gamma_0 \geq 0$, $\frac{\lambda_n}{n} \to \lambda_0 \geq 0$, $\lim_{n\to\infty} \frac{X^T X}{n} = C$ *is nonsingular and that* f *is a convex function. Let*

$$\tilde{\beta} = \arg\min_{\beta} \frac{1}{2}||Y - X\beta||_2^2 + \gamma_n \sum_{j=1}^{p} (1 - e^{-\frac{\lambda_n}{\gamma_n} f(\beta_j)})$$

Then $\tilde{\beta} \to_p \arg\min(Z)$ *where*

$$Z(\beta) = (\beta - \beta^*) C (\beta - \beta^*) + \gamma_0 \sum_{j=1}^{p} (1 - e^{-\frac{\lambda_0}{\gamma_0} f(\beta_j)}),$$

where we define the second term to be 0 *if* $\lambda_0 = 0$ *or* $\gamma_0 = 0$*. Hence if* $\gamma_n = o(n)$ *or* $\lambda_n = o(n)$*, then* $\tilde{\beta}$ *is consistent.*

outline of proof:

- Let $Z_n(\beta) = \frac{1}{2n}||Y - X\beta||^2 + \frac{\gamma_n}{n}\sum_{i=1}^{p}(1 - e^{-\frac{\lambda_n}{\gamma_n}f(\beta_i)})$.

outline of proof:

- Let $Z_n(\beta) = \frac{1}{2n}||Y - X\beta||^2 + \frac{\gamma_n}{n}\sum_{i=1}^{p}(1 - e^{-\frac{\lambda_n}{\gamma_n}f(\beta_i)})$.
- Our statements are proved if

$$\sup_{\beta \in K} |Z_n(\beta) - Z(\beta) - \frac{\sigma^2}{2}| \to_p 0$$

for any compact set $K \subset R^p$ and that

$$\tilde{\beta} = O_p(1)$$

.

outline of proof:

- Let $Z_n(\beta) = \frac{1}{2n}||Y - X\beta||^2 + \frac{\gamma_n}{n}\sum_{i=1}^{p}(1 - e^{-\frac{\lambda_n}{\gamma_n}f(\beta_i)})$.
- Our statements are proved if

$$\sup_{\beta \in K} |Z_n(\beta) - Z(\beta) - \frac{\sigma^2}{2}| \to_p 0$$

for any compact set $K \subset R^p$ and that

$$\tilde{\beta} = O_p(1)$$

.

- The first one follows from the convergence criteria that we have in the theorem and the second one can be shown by finding a bounding ball of the solutions.

Theorem

Assume that $||C_{11}^{-1}||_\infty < \frac{K_1}{n}$ *and* $||C_{21}||_\infty < K_2 n$ *for some constants* $K_1, K_2 < \infty$ *independent of* n*, where* $||\cdot||_\infty$ *is the* ∞*-operator norm. Assume also that there is a constant* $\delta > 0$ *such that for all* n*,* $n > \lambda_n > n^{1/2+2\delta}$ *and* $\lambda_n|\beta_{nj}|/\gamma_n > n^{2\delta}$ *and* $|\beta_{nj}| > n^{-1/2+2\delta}$ *for all* $j = 1, \ldots, r_n$*. Assume in addition that* $\gamma_n > 1$ *and* $q_n < e^{n^\delta}$*.*
Let L *be the minimisation objective*

$$L(\beta) = L_n(\beta; Y) = \frac{1}{2}||Y - X\beta)||_2^2 + \gamma_n \sum_{j=1}^{p_n}(1 - e^{-\frac{\lambda_n}{\gamma_n}|\beta_j|}).$$

Then with probability at least $1 - e^{-n^\delta}$*,* L *has a local minimum* $\bar{\beta}$ *such that with probability* $1 - e^{-n^\delta}$*,* $||\bar{\beta} - \beta^*||_\infty < n^{-1/2+\delta}$ *and* $sign(\bar{\beta}) = sign(\beta^*)$*.*
Hence if L *has a unique minimum, then*

$$\tilde{\beta}_n = \arg\min_\beta \left[\frac{1}{2}||Y - X\beta)||_2^2 + \gamma_n \sum_{j=1}^{p_n}(1 - e^{-\frac{\lambda_n}{\gamma_n}|\beta_j|})\right]$$

satisfies with probability at least $1 - e^{-n^\delta}$ *that* $||\tilde{\beta} - \beta^*||_\infty < n^{-1/2+\delta}$ *and* $sign(\tilde{\beta}) = sign(\beta^*)$*.*

Outline of proof: We can rewrite the loss function as

$$L(\beta) = \frac{1}{2}||\xi||_2^2 + \frac{1}{2}(\beta - \beta^*)X^T X(\beta - \beta^*) + \sum_j \left[\sqrt{n}(\beta_j - \beta_j^*)Z_j + \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_j|}) \right]$$

Where $Z = (Z_1, ..., Z_{r+q})^T \sim N(0, X^T X/n)$.

Outline of proof: We can rewrite the loss function as

$$L(\beta) = \frac{1}{2}||\xi||_2^2 + \frac{1}{2}(\beta - \beta^*)X^T X(\beta - \beta^*) + \sum_j \left[\sqrt{n}(\beta_j - \beta_j^*)Z_j + \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_j|})\right]$$

Where $Z = (Z_1, ..., Z_{r+q})^T \sim N(0, X^T X/n)$. Write β as $\beta = (\phi^T, \psi^T)^T$ where the true values of ψ is $\psi^* = 0$.

Outline of proof: We can rewrite the loss function as

$$L(\beta) = \frac{1}{2}||\xi||_2^2 + \frac{1}{2}(\beta - \beta^*)X^T X(\beta - \beta^*) + \sum_j \left[\sqrt{n}(\beta_j - \beta_j^*)Z_j + \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_j|})\right]$$

Where $Z = (Z_1, ..., Z_{r+q})^T \sim N(0, X^T X/n)$. Write β as $\beta = (\phi^T, \psi^T)^T$ where the true values of ψ is $\psi^* = 0$. The KKT conditions can now be expressed as

$$\sqrt{n}Z_\phi + C_{11}(\bar{\phi} - \phi^*) + \lambda[e^{-\frac{\lambda_n}{\gamma_n}|\phi|}]_{j=1}^r = 0$$

$$\forall j : -\lambda < \sqrt{n}Z_{\psi,j} + (C_{21}(\bar{\phi} - \phi^*))_j < \lambda$$

Outline of proof: We can rewrite the loss function as

$$L(\beta) = \frac{1}{2}||\xi||_2^2 + \frac{1}{2}(\beta - \beta^*)X^TX(\beta - \beta^*) + \sum_j \left[\sqrt{n}(\beta_j - \beta_j^*)Z_j + \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_j|})\right]$$

Where $Z = (Z_1, ..., Z_{r+q})^T \sim N(0, X^TX/n)$. Write β as $\beta = (\phi^T, \psi^T)^T$ where the true values of ψ is $\psi^* = 0$. The KKT conditions can now be expressed as

$$\sqrt{n}Z_\phi + C_{11}(\bar{\phi} - \phi^*) + \lambda[e^{-\frac{\lambda_n}{\gamma_n}|\phi|}]_{j=1}^r = 0$$

$$\forall j : -\lambda < \sqrt{n}Z_{\psi,j} + (C_{21}(\bar{\phi} - \phi^*))_j < \lambda$$

Use the convergence rate of Newtons method to show that ϕ converges well within the orthant of ϕ^*.

Outline of proof: We can rewrite the loss function as

$$L(\beta) = \frac{1}{2}||\xi||_2^2 + \frac{1}{2}(\beta - \beta^*)X^TX(\beta - \beta^*) + \sum_j \left[\sqrt{n}(\beta_j - \beta_j^*)Z_j + \gamma(1 - e^{-\frac{\lambda}{\gamma}|\beta_j|})\right]$$

Where $Z = (Z_1, ..., Z_{r+q})^T \sim N(0, X^TX/n)$. Write β as $\beta = (\phi^T, \psi^T)^T$ where the true values of ψ is $\psi^* = 0$. The KKT conditions can now be expressed as

$$\sqrt{n}Z_\phi + C_{11}(\bar{\phi} - \phi^*) + \lambda[e^{-\frac{\lambda_n}{\gamma_n}|\phi|}]_{j=1}^r = 0$$

$$\forall j : -\lambda < \sqrt{n}Z_{\psi,j} + (C_{21}(\bar{\phi} - \phi^*))_j < \lambda$$

Use the convergence rate of Newtons method to show that ϕ converges well within the orthant of ϕ^*. Finally show that this solution also fulfils the second KKT condition.

Theorem

The minimization problem

$$\tilde{\beta} = \min_{\beta} L(\beta) = \min_{\beta} \frac{1}{2}||Y - X\beta||^2 + \gamma \sum_i (1 - e^{-\frac{\lambda}{\gamma}\beta_i^2}).$$

is convex if $\lambda < \frac{s_1^2 e^{\frac{3}{2}}}{4}$

Theorem

The minimization problem

$$\tilde{\beta} = \min_{\beta} L(\beta) = \min_{\beta} \frac{1}{2}||Y - X\beta||^2 + \gamma \sum_i (1 - e^{-\frac{\lambda}{\gamma}\beta_i^2}).$$

is convex if $\lambda < \frac{s_1^2 e^{\frac{3}{2}}}{4}$

outline of proof: Calculate the hessian och check for positive eigenvalues.

Properties

	Unbiased	Consistency	Continuity	Sparsity	Sign consistency
OLS	Yes	Yes	Yes	No	No
Lasso	No	Yes	Yes	Yes	Yes
Ridge	No	Yes	Yes	No	No
Bridge	When $\gamma < 1$	Yes	When $\gamma \geq 1$	When $\gamma \leq 1$	When $\gamma \leq 1$
SCAD	Yes	Yes	Yes	Yes	Yes
Adaptive lasso	Yes	Yes	Yes	Yes	Yes

Properties

	Unbiased	Consistency	Continuity	Sparsity	Sign consistency
OLS	Yes	Yes	Yes	No	No
Lasso	No	Yes	Yes	Yes	Yes
Ridge	No	Yes	Yes	No	No
Bridge	When $\gamma < 1$	Yes	When $\gamma \geq 1$	When $\gamma \leq 1$	When $\gamma \leq 1$
SCAD	Yes	Yes	Yes	Yes	Yes
Adaptive lasso	Yes	Yes	Yes	Yes	Yes
EWL	Yes	Yes	When $\gamma \geq \lambda^2/s_1^2$	Yes	Yes
EWR	Yes	Yes	When $\lambda \leq s_1^2 e^{\frac{3}{2}}/4$	No	No

Table: A table of which of the common approaches fulfills the requested properties.

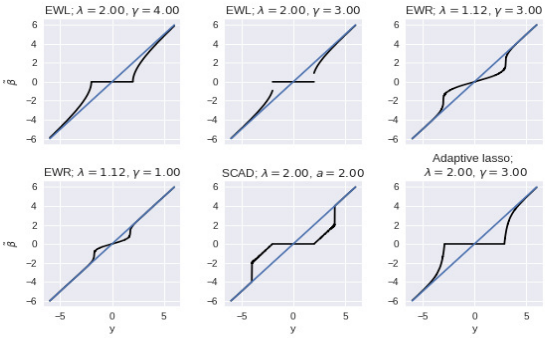

Figure: How $\tilde{\beta}$ depends on y with EWL, EWR, SCAD and adaptive lasso for a single data point y and a single parameter β. In this case $s_1^2 = 1$ and EWL is convex whenever $\gamma > \lambda^2$ and EWR is convex whenever $\lambda < e^{3/2}/4 \approx 1.12$.

Training algorithm (EWL)

- The EWL estimator can be optimised with a coordinate gradient descent.

Training algorithm (EWL)

- The EWL estimator can be optimised with a coordinate gradient descent.
- The coordinate-wise derivative is given by

$$-\rho_j + n\beta_j + \left\{ \begin{array}{ll} -\lambda e^{\frac{\lambda}{\gamma}\beta_j} & \beta_j < 0 \\ [-\lambda, \lambda] & \beta_j = 0 \\ \lambda e^{-\frac{\lambda}{\gamma}\beta_j} & \beta_j > 0 \end{array} \right\}$$

where $\rho_j = X_j^T(Y - X_{i\neq j}\beta_{j\neq i})$

Training algorithm (EWL)

- The EWL estimator can be optimised with a coordinate gradient descent.
- The coordinate-wise derivative is given by

$$-\rho_j + n\beta_j + \left\{ \begin{array}{ll} -\lambda e^{\frac{\lambda}{\gamma}\beta_j} & \beta_j < 0 \\ [-\lambda, \lambda] & \beta_j = 0 \\ \lambda e^{-\frac{\lambda}{\gamma}\beta_j} & \beta_j > 0 \end{array} \right\}$$

where $\rho_j = X_j^T(Y - X_{i\neq j}\beta_{j\neq i})$

- Solving for β in each setting results in the solution

$$\beta_j = \begin{cases} \frac{\gamma}{\lambda} W(-\frac{\lambda^2}{n\gamma} e^{-\rho_j \frac{\lambda}{n\gamma}}) + \frac{\rho_j}{n} & \rho_j > \lambda \\ 0 & -\lambda \leq \rho_j \leq \lambda \\ -\frac{\gamma}{\lambda} W(-\frac{\lambda^2}{n\gamma} e^{\rho_j \frac{\lambda}{n\gamma}}) + \frac{\rho_j}{n} & \rho_j < -\lambda \end{cases}$$

- W is the Lambert W-function, i.e. the inverse of $f(x) = xe^x$.

Algorithm 1 Training algorithm with weighted L1 regularisation.

```
procedure Train(X,Y,λ,γ,N = max number of iterations, ϵ = tolerance)
    λ ← λ/n
    γ ← γ/n
    β ← 0
    for iteration = 0...N do
        β̂ ← β
        perm = random permutation of [1...m]
        for j ∈ perm do
            ρ ← X_j^T (Y − X_{i≠j} β_{i≠j})/n
            if |ρ| > λ then
                β_j ← (γ/λ) sign(ρ) W(−(λ²/γ) e^{−|ρ| λ/γ}) + ρ
            else
                β_j ← 0
        if max(|β − β̂|) < ϵ then
            Break loop
    return β
```

$$\beta_j \leftarrow \frac{\gamma}{\lambda} sign(\rho) W\left(-\frac{\lambda^2}{\gamma} e^{-|\rho|\frac{\lambda}{\gamma}}\right) + \rho$$

Training algorithm (EWR)

- To minimize the EWR loss function we will start with the minimisation problem of both β and u.

$$\arg\min_{\beta,u} \frac{1}{2}||Y - X\beta||_2^2 + \sum_{k=1}^{p} \frac{u_k\lambda}{2}\beta_k^2 + \sum_{k=1}^{p} \gamma(u_i \log u_i - u_i + 1)$$

Training algorithm (EWR)

- To minimize the EWR loss function we will start with the minimisation problem of both β and u.

$$\arg\min_{\beta,u} \frac{1}{2}||Y - X\beta||_2^2 + \sum_{k=1}^{p} \frac{u_k\lambda}{2}\beta_k^2 + \sum_{k=1}^{p} \gamma(u_i \log u_i - u_i + 1)$$

- We can now split this in to two minimisation problems that both can be solved for analytically.

$$\arg\min_{\beta} \frac{1}{2}||Y - X\beta||_2^2 + \sum_{i} [\lambda u_i f(\beta_i) + \gamma(u_i \log(u_i) - u_i + 1)] =$$

$$\left[X^T X + \lambda diag(\mathrm{u})\right]^{-1} X^T Y$$

$$\arg\min_{\mathrm{u}} \frac{1}{2}||Y - X\beta||_2^2 + \sum_{i=1}^{n} [\lambda u_i f(\beta_i) + \gamma(u_i \log(u_i) - u_i + 1)] = e^{-\frac{\lambda}{2\gamma}\beta^2}$$

Training algorithm (EWR)

Algorithm 2 Training algorithm with weighted L2 regularisation.

procedure Train(X,Y,λ,γ,N = max number of iterations, ϵ = tolerance)
    u $\leftarrow$ 1
    $\beta \leftarrow 0$
    for iteration $= 0 \dots N$ **do**
        $\hat{\beta} \leftarrow \beta$
        $\beta \leftarrow (X^T X + \lambda diag(\mathrm{u}))^{-1} X^T Y$
        u $\leftarrow e^{-\beta^2 \frac{\lambda}{2\gamma}}$
        if $\max(|\beta - \hat{\beta}|) < \epsilon$ **then**
            Break loop
    return β

Experiments

1. **Experiment 1.**
   - The data matrix X and true covariates β^* are sampled independently.
   - Data are samples as $Y = X\beta^* + \epsilon$ where $\epsilon \in N(0, \sigma^2 I)$.
   - σ are varied on the interval $[0, 40]$.

1. **Experiment 1.**
   - The data matrix X and true covariates β^* are sampled independently.
   - Data are samples as $Y = X\beta^* + \epsilon$ where $\epsilon \in N(0, \sigma^2 I)$.
   - σ are varied on the interval $[0, 40]$.
2. **Experiment 2.**
   - The data matrix X with internal correlation ρ between some columns.
   - The true covariates β^* are samples independently.
   - Data are samples as $Y = X\beta^* + \epsilon$ where $\epsilon \in N(0, 30^2 * I)$.
   - ρ are varied on the interval $[0, 0.8]$.

Experiment 1

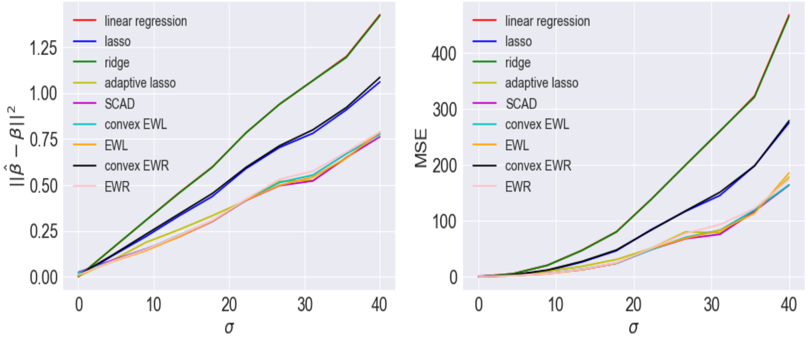

Figure: The average L_2 distance between the estimated parameters $\hat{\beta}$ and the true parameters β (**left**) and the mean squared error of predictions on test data (**right**) over 100 runs as functions of the signal to noise ratio (SNR) for nine models on uncorrelated covariates.

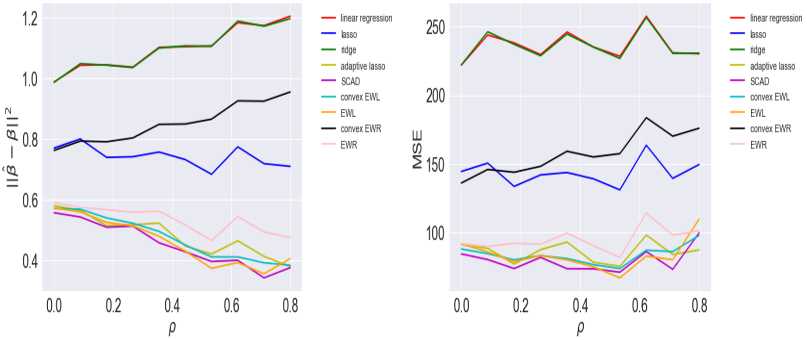

Figure: The L_2 distance between the estimated parameters $\hat{\beta}$ and the true parameters β^* (**left**) and the mean square error of predictions on the test set (**right**) as functions of correlation between covariates as ρ varies between 0 and 0.8. The results displayed are averages over 100 runs. The solid lines correspond to the mean distance and the dashed lines correspond to the 95% confidence intervals.

Eventual feature work

- Can this methods be used in a neural network setting? When will the non-convexity be a problem?
- Is there an efficient way to find the optimal values of λ and γ?
- It we add an additional regularisation term $\frac{\kappa}{2}\beta_k^2$ to the loss function we can guarantee convexity without consider s_1. How similar will this be to elastic net?
- Is there another regularisation function for the weights that make a more suitable estimator?

Thank you!

Thank you!

Questions?