UNIVERSITY OF
COPENHAGEN

The Choice of Normalization Directly Affects Feature Selection in Regularized Regression

DSTS's Two-Day Meeting, Autumn 2024

Johan Larsson

https://jolars.co, @jolars@mastodon.social

Department of Mathematical Sciences, Copenhagen University

November 13, 2024

This Talk

Problem and Motivation

Feature normalization has large effects in regularized regression (lasso, ridge) but there is no research on this.

This Talk

Problem and Motivation

Feature normalization has large effects in regularized regression (lasso, ridge) but there is no research on this.

Results

- Class balance has a normalization-dependent impact on the model.
- In mixed data, choice of normalization implictly weighs features' importances.

Figure 1: Joint work with Jonas Wallin

Overview

Preliminaries

General Setup

- Data consists of a fixed matrix of features $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ and a response vector $\boldsymbol{y} \in \mathbb{R}^{n}$.
- $\boldsymbol{y}$ comes from a linear model, that is,

$$y_i = \beta_0^* + \boldsymbol{x}_i^\intercal \boldsymbol{\beta}^* + \varepsilon_i \quad \text{for} \quad i \in 1, \ldots, n,$$

  where $\boldsymbol{\beta}^*$ is the vector of *true* coefficients.
- ε_i is the measurement noise, generated from some random variable.

The Elastic Net

Linear regression plus a combination of the ℓ_1 and ℓ_2 penalties:

$$\boldsymbol{\beta}^* = \arg\min_{\beta \in \mathbb{R}^p} \left(\frac{1}{2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \underbrace{\lambda_1 \|\boldsymbol{\beta}\|_1}_{\text{lasso}} + \underbrace{\frac{\lambda_2}{2} \|\boldsymbol{\beta}\|_2^2}_{\text{ridge}} \right)$$

The Elastic Net

Linear regression plus a combination of the ℓ_1 and ℓ_2 penalties:

$$\boldsymbol{\beta}^* = \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \left(\frac{1}{2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \underbrace{\lambda_1 \|\boldsymbol{\beta}\|_1}_{\text{lasso}} + \underbrace{\frac{\lambda_2}{2} \|\boldsymbol{\beta}\|_2^2}_{\text{ridge}} \right)$$

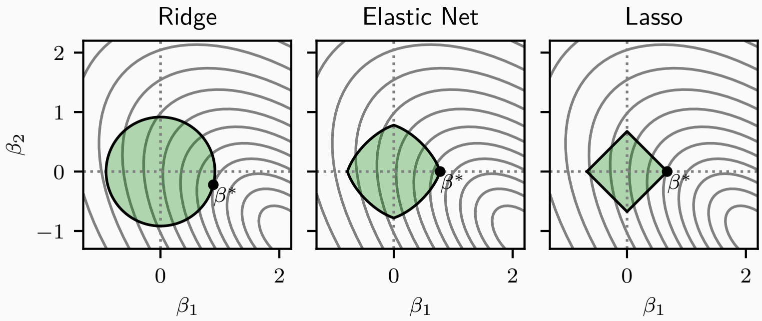


Figure 2: Elastic net is a combination of the lasso and ridge.

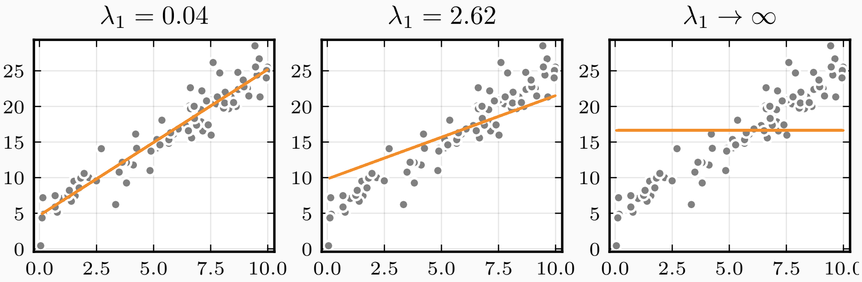


Figure 3: Regularization for a simple linear regression problem

Regularization

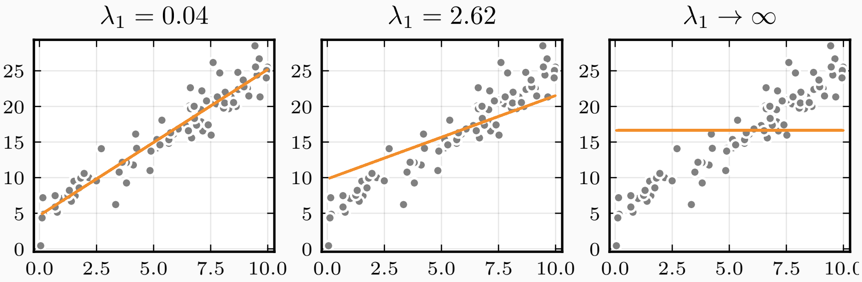


Figure 3: Regularization for a simple linear regression problem

Why Regularize? (Lasso, Ridge)

- Uniqueness when $p \gg n$
- To overcome overfitting

Regularization

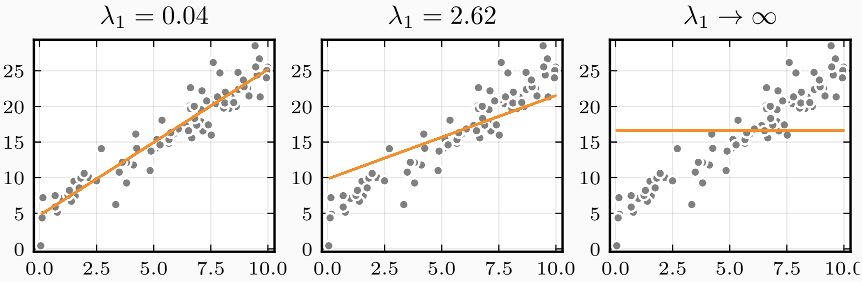


Figure 3: Regularization for a simple linear regression problem

Why Regularize? (Lasso, Ridge)

- Uniqueness when $p \gg n$
- To overcome overfitting

Why Sparsity? (Lasso)

- Interpretability
- The sparsity bet

The Elastic Net Path

- Don't know optimal λ_1 and λ_2 in advance.

The Elastic Net Path

- Don't know optimal λ_1 and λ_2 in advance.
- Instead we use hyper-parameter optimization (e.g. cross-validation).

The Elastic Net Path

- Don't know optimal λ_1 and λ_2 in advance.
- Instead we use hyper-parameter optimization (e.g. cross-validation).
- Common parametrization:

$$\lambda_1 = \alpha\lambda,$$
$$\lambda_2 = (1-\alpha)\lambda$$

with $\alpha \in [0, 1]$.

The Elastic Net Path

- Don't know optimal λ_1 and λ_2 in advance.
- Instead we use hyper-parameter optimization (e.g. cross-validation).
- Common parametrization:

$$\lambda_1 = \alpha\lambda,$$
$$\lambda_2 = (1-\alpha)\lambda$$

  with $\alpha \in [0, 1]$.
- For each α, solve the elastic net over a sequence of λ: the **elastic net path**.

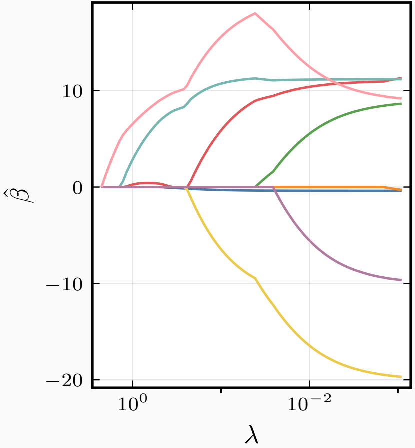


Figure 4: The elastic net path

Sensitivity to Scale

Lasso and ridge penalties are **norms**—feature scales matter!

Sensitivity to Scale

Lasso and ridge penalties are **norms**—feature scales matter!

Example

Assume

$$\boldsymbol{X} \sim \mathrm{Normal}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}4 & 0\\0 & 1\end{bmatrix}\right), \qquad \boldsymbol{\beta}^* = \begin{bmatrix}\frac{1}{2}\\1\end{bmatrix}.$$

Sensitivity to Scale

Lasso and ridge penalties are **norms**—feature scales matter!

Example

Assume

$$\boldsymbol{X} \sim \mathrm{Normal}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}\right), \qquad \beta^* = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}.$$

Model	$\hat{\beta}$	$\hat{\beta}_{\mathsf{std}}$
OLS	$\begin{bmatrix} 0.50 & 1.00 \end{bmatrix}^\intercal$	$\begin{bmatrix} 1.00 & 1.00 \end{bmatrix}^\intercal$

Sensitivity to Scale

Lasso and ridge penalties are **norms**—feature scales matter!

Example

Assume

$$\boldsymbol{X} \sim \mathrm{Normal}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}\right), \qquad \beta^* = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}.$$

Model	$\hat{\beta}$	$\hat{\beta}_{\text{std}}$
OLS	$\begin{bmatrix} 0.50 & 1.00 \end{bmatrix}^\intercal$	$\begin{bmatrix} 1.00 & 1.00 \end{bmatrix}^\intercal$
Lasso	$\begin{bmatrix} 0.38 & 0.50 \end{bmatrix}^\intercal$	$\begin{bmatrix} 0.74 & 0.50 \end{bmatrix}^\intercal$

Sensitivity to Scale

Lasso and ridge penalties are **norms**—feature scales matter!

Example

Assume

$$\boldsymbol{X} \sim \mathrm{Normal}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}\right), \qquad \boldsymbol{\beta}^* = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}.$$

Model	$\hat{\boldsymbol{\beta}}$	$\hat{\boldsymbol{\beta}}_{\text{std}}$
OLS	$\begin{bmatrix} 0.50 & 1.00 \end{bmatrix}^\intercal$	$\begin{bmatrix} 1.00 & 1.00 \end{bmatrix}^\intercal$
Lasso	$\begin{bmatrix} 0.38 & 0.50 \end{bmatrix}^\intercal$	$\begin{bmatrix} 0.74 & 0.50 \end{bmatrix}^\intercal$
Ridge	$\begin{bmatrix} 0.37 & 0.41 \end{bmatrix}^\intercal$	$\begin{bmatrix} 0.74 & 0.41 \end{bmatrix}^\intercal$

Sensitivity to Scale

Lasso and ridge penalties are **norms**—feature scales matter!

Example

Assume

$$\boldsymbol{X} \sim \mathrm{Normal}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}4 & 0\\0 & 1\end{bmatrix}\right), \qquad \boldsymbol{\beta}^* = \begin{bmatrix}\frac{1}{2}\\1\end{bmatrix}.$$

Model	$\hat{\boldsymbol{\beta}}$	$\hat{\boldsymbol{\beta}}_{\text{std}}$
OLS	$\begin{bmatrix}0.50 & 1.00\end{bmatrix}^\intercal$	$\begin{bmatrix}1.00 & 1.00\end{bmatrix}^\intercal$
Lasso	$\begin{bmatrix}0.38 & 0.50\end{bmatrix}^\intercal$	$\begin{bmatrix}0.74 & 0.50\end{bmatrix}^\intercal$
Ridge	$\begin{bmatrix}0.37 & 0.41\end{bmatrix}^\intercal$	$\begin{bmatrix}0.74 & 0.41\end{bmatrix}^\intercal$

Large scale means less penalization because the size of β_j can be smaller for an equivalent effect (on $\boldsymbol{y}$).

Normalization

- Scale sensitivity can be mitigated by normalizing the features.

Normalization

- Scale sensitivity can be mitigated by normalizing the features.
- Let $\tilde{\boldsymbol{X}}$ be the normalized feature matrix, with elements

$$\tilde{x}_{ij} = \frac{x_{ij} - c_j}{s_j}.$$

Normalization

- Scale sensitivity can be mitigated by normalizing the features.
- Let $\tilde{\boldsymbol{X}}$ be the normalized feature matrix, with elements

$$\tilde{x}_{ij} = \frac{x_{ij} - c_j}{s_j}.$$

- After fitting, we transform the coefficients back to their original scale via

$$\hat{\beta}_j = \frac{\hat{\beta}_j^{(n)}}{s_j} \quad \text{for} \quad j = 1, 2, \ldots, p,$$

where $\hat{\beta}_j^{(n)}$ is a coefficient from the normalized problem.

Table 1: Common ways to normalize $\boldsymbol{X}$

Normalization	Centering (c_j)	Scaling (s_j)
Standardization	$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}$	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}$
Min–Max	$\min_i(x_{ij})$	$\max_i(x_{ij}) - \min_i(x_{ij})$
Unit Vector (L2)	0	$\sqrt{\sum_{i=1}^{n} x_{ij}^2}$
Max–Abs	0	$\max_i(\lvert x_{ij}\rvert)$
Adaptive Lasso	0	β_j^{OLS}

Motivation and Aims

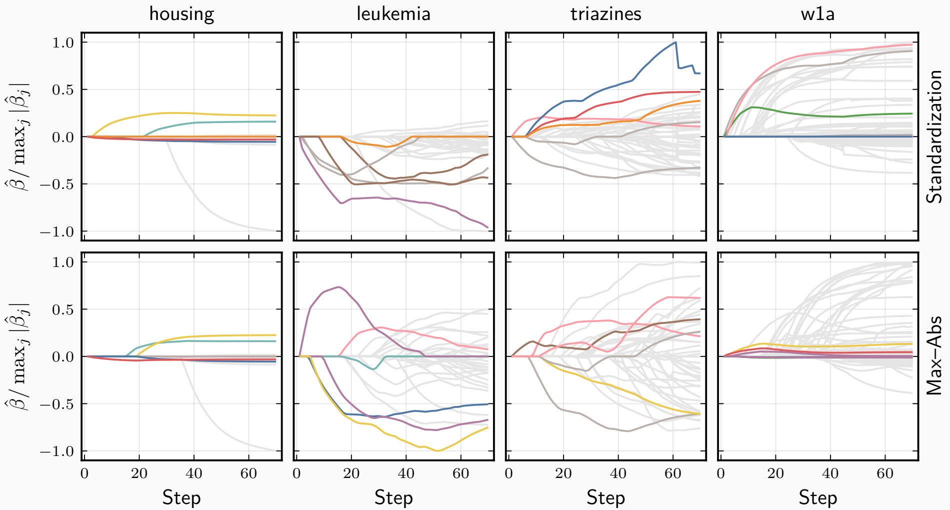


Figure 5: Normalization matters. Lasso paths under two different types of normalization (standardization and max–abs normalization). The union of the first ten features selected in any of the settings are colored.

Motivation

- So, normalization matters but there has been no research into this.

Motivation

Motivation

- So, normalization matters but there has been no research into this.
- Everyone agrees you need to normalize, but how to do so is usually motivated by being “standard”.

Motivation

- So, normalization matters but there has been no research into this.
- Everyone agrees you need to normalize, but how to do so is usually motivated by being "standard".
- Documentation for popular machine learning packages advocate different normalization strategies when data is sparse.

Motivation

- So, normalization matters but there has been no research into this.
- Everyone agrees you need to normalize, but how to do so is usually motivated by being "standard".
- Documentation for popular machine learning packages advocate different normalization strategies when data is sparse.
- Approach depends on field: statisticians standardize, signal processors use ℓ_1 normalization, machine learning people scale to $[0, 1]$ or $[-1, 1]$.)

Motivation

- So, normalization matters but there has been no research into this.
- Everyone agrees you need to normalize, but how to do so is usually motivated by being "standard".
- Documentation for popular machine learning packages advocate different normalization strategies when data is sparse.
- Approach depends on field: statisticians standardize, signal processors use ℓ_1 normalization, machine learning people scale to $[0, 1]$ or $[-1, 1]$.)

Motivation

- So, normalization matters but there has been no research into this.
- Everyone agrees you need to normalize, but how to do so is usually motivated by being "standard".
- Documentation for popular machine learning packages advocate different normalization strategies when data is sparse.
- Approach depends on field: statisticians standardize, signal processors use ℓ_1 normalization, machine learning people scale to $[0, 1]$ or $[-1, 1]$.)

Aims

- Binary features, particularly with respect to the class balance thereof

Motivation

- So, normalization matters but there has been no research into this.
- Everyone agrees you need to normalize, but how to do so is usually motivated by being "standard".
- Documentation for popular machine learning packages advocate different normalization strategies when data is sparse.
- Approach depends on field: statisticians standardize, signal processors use ℓ_1 normalization, machine learning people scale to $[0, 1]$ or $[-1, 1]$.)

Aims

- Binary features, particularly with respect to the class balance thereof
- A mix of binary and normally distributed features

Results

Orthogonal Features

There is no explicit solution to the elastic net problem in general (unless $\lambda_1 = 0$).

[1] We have also assumed that the features are mean-centered here.

Orthogonal Features

There is no explicit solution to the elastic net problem in general (unless $\lambda_1 = 0$).

But if we assume that the features are orthogonal, that is

$$\tilde{\boldsymbol{X}}^\intercal \tilde{\boldsymbol{X}} = \operatorname{diag}(\tilde{\boldsymbol{x}}_1^\intercal \tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_p^\intercal \tilde{\boldsymbol{x}}_p),$$

then there is:[1]:

$$\hat{\beta}_j = \frac{\mathrm{S}_{\lambda_1}\left(\tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{y}\right)}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)},$$

where

$$\mathrm{S}_\lambda(z) = \operatorname{sign}(z)\max(|z| - \lambda, 0).$$

[1]We have also assumed that the features are mean-centered here.

There is no explicit solution to the elastic net problem in general (unless $\lambda_1 = 0$).

But if we assume that the features are orthogonal, that is

$$\tilde{\boldsymbol{X}}^\intercal \tilde{\boldsymbol{X}} = \operatorname{diag}(\tilde{\boldsymbol{x}}_1^\intercal \tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_p^\intercal \tilde{\boldsymbol{x}}_p),$$

then there is:[1]:

$$\hat{\beta}_j = \frac{\mathrm{S}_{\lambda_1}\left(\tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{y}\right)}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)},$$

where

$$\mathrm{S}_\lambda(z) = \operatorname{sign}(z)\max(|z| - \lambda, 0).$$

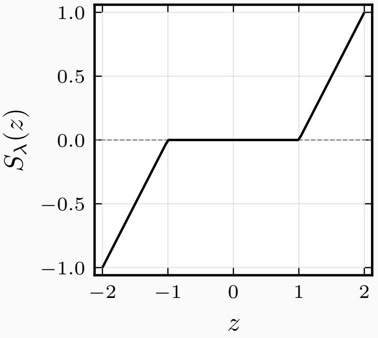


Figure 6: Soft thresholding

[1]We have also assumed that the features are mean-centered here.

Bias and Variance of the Elastic Net Estimator

The goal is computing the expected value of the elastic net estimator,

$$\operatorname{E}\hat{\beta}_j = \frac{\operatorname{E}\operatorname{S}_{\lambda_1}\left(\tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{y}\right)}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)},$$

since we treat $\boldsymbol{X}$ as fixed.

Bias and Variance of the Elastic Net Estimator

The goal is computing the expected value of the elastic net estimator,

$$\operatorname{E} \hat{\beta}_j = \frac{\operatorname{E} \operatorname{S}_{\lambda_1}\left(\tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{y}\right)}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)},$$

since we treat $\boldsymbol{X}$ as fixed.

Letting $Z = \tilde{\boldsymbol{x}}^\intercal \boldsymbol{y}$ and assuming that ε_i is i.i.d. Normally-distributed with mean zero and finite variance σ_ε^2, we have

$$Z \sim \operatorname{Normal}\left(\mu = \tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{x}_j \beta_j, \sigma^2 = \sigma_\varepsilon^2 \|\tilde{\boldsymbol{x}}_j\|_2^2\right).$$

Next, will will turn to $\operatorname{E} \operatorname{S}_{\lambda_1}(Z)$.

Bias

The expected value of the soft-thresholding estimator is

$$\begin{aligned}
\mathrm{E}\,\mathrm{S}_\lambda(Z) &= \int_{-\infty}^{\infty} \mathrm{S}_\lambda(z) f_Z(z)\,\mathrm{d}z \\
&= \int_{-\infty}^{-\lambda} (z+\lambda) f_Z(z)\,\mathrm{d}z \\
&\quad + \int_{\lambda}^{\infty} (z-\lambda) f_Z(z)\,\mathrm{d}z.
\end{aligned}$$

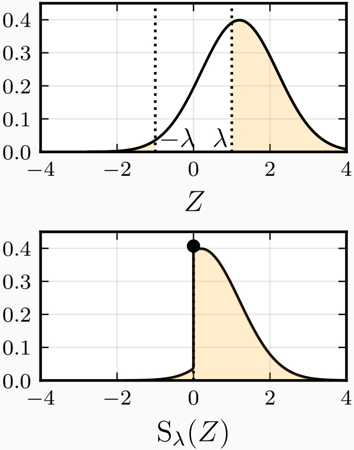


Figure 7: Distributions of Z and its value after soft-thresholding.

Bias

The expected value of the soft-thresholding estimator is

$$\begin{aligned}\operatorname{E}\operatorname{S}_\lambda(Z) &= \int_{-\infty}^{\infty} \operatorname{S}_\lambda(z) f_Z(z)\,\mathrm{d}z \\ &= \int_{-\infty}^{-\lambda} (z+\lambda) f_Z(z)\,\mathrm{d}z \\ &\quad + \int_{\lambda}^{\infty} (z-\lambda) f_Z(z)\,\mathrm{d}z.\end{aligned}$$

The bias of $\hat{\beta}_j$ is

$$\operatorname{E}\hat{\beta}_j - \beta_j^* = \frac{1}{d_j}\operatorname{E}\operatorname{S}_\lambda(Z) - \beta_j^*,$$

where $d_j = s_j\,(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2)$.

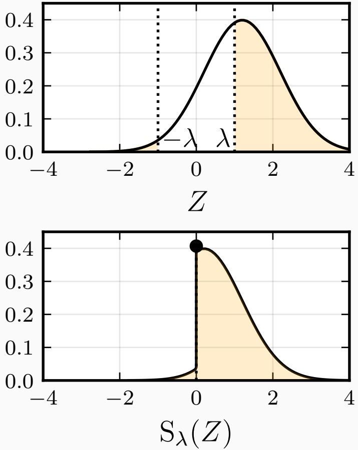


Figure 7: Distributions of Z and its value after soft-thresholding.

Variance

The variance of the soft-thresholding estimator is

$$\operatorname{Var} S_\lambda(Z) = \int_{-\infty}^{-\lambda} (z+\lambda)^2 f_Z(z)\,\mathrm{d}z + \int_{\lambda}^{\infty} (z-\lambda)^2 f_Z(z)\,\mathrm{d}z - (\operatorname{E} \mathrm{S}_\lambda(Z))^2$$

and consequently the variance of the elastic net estimator is

$$\operatorname{Var} \hat{\beta}_j = \frac{1}{d_j^2} \operatorname{Var} \mathrm{S}_\lambda(Z).$$

Binary Features

Recall that

$$Z \sim \operatorname{Normal}\left(\mu = \tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{x}_j \beta_j, \sigma^2 = \sigma_\varepsilon^2 \|\tilde{\boldsymbol{x}}_j\|_2^2\right)$$

and assume we have a binary feature $\boldsymbol{x}_j$, such that $x_{ij} \in \{0, 1\}$. Let $q \in [0, 1]$ be the class balance of this feature, that is: $q = \bar{\boldsymbol{x}}_j$.

In this case, we observe that

$$\|\tilde{\boldsymbol{x}}_j\|_2^2 = \frac{n(q - q^2)}{s_j^2},$$

$$\tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{x}_j = \frac{n(q - q^2)}{s_j}.$$

Binary Features

Recall that

$$Z \sim \operatorname{Normal}\left(\mu = \tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{x}_j \beta_j, \sigma^2 = \sigma_\varepsilon^2 \|\tilde{\boldsymbol{x}}_j\|_2^2\right)$$

and assume we have a binary feature $\boldsymbol{x}_j$, such that $x_{ij} \in \{0, 1\}$. Let $q \in [0, 1]$ be the class balance of this feature, that is: $q = \bar{\boldsymbol{x}}_j$.

In this case, we observe that

$$\begin{aligned} \|\tilde{\boldsymbol{x}}_j\|_2^2 &= \frac{n(q - q^2)}{s_j^2}, \\ \tilde{\boldsymbol{x}}_j^\intercal \boldsymbol{x}_j &= \frac{n(q - q^2)}{s_j}. \end{aligned}$$

And consequently

$$\mu = \frac{\beta_j^* n(q - q^2)}{s_j}, \qquad \sigma^2 = \frac{\sigma_\varepsilon^2 n(q - q^2)}{s_j^2}, \qquad d_j = \frac{n(q - q^2)}{s_j} + \lambda_2 s_j.$$

Noiseless Case for Binary Features

In the noiseless case, we have

$$\hat{\beta}_j = \frac{\mathrm{S}_{\lambda_1}(\tilde{\boldsymbol{x}}^\intercal \boldsymbol{y})}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)} = \frac{\mathrm{S}_{\lambda_1}\left(\frac{\beta_j^* n(q-q^2)}{s_j}\right)}{s_j\left(\frac{n(q-q^2)}{s_j^2} + \lambda_2\right)}.$$

Noiseless Case for Binary Features

In the noiseless case, we have

$$\hat{\beta}_j = \frac{\mathrm{S}_{\lambda_1}(\tilde{\boldsymbol{x}}^\intercal \boldsymbol{y})}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)} = \frac{\mathrm{S}_{\lambda_1}\left(\frac{\beta_j^* n(q-q^2)}{s_j}\right)}{s_j\left(\frac{n(q-q^2)}{s_j^2} + \lambda_2\right)}.$$

- Means that the elastic net estimator depends on class balance (q).

Noiseless Case for Binary Features

In the noiseless case, we have

$$\hat{\beta}_j = \frac{\mathrm{S}_{\lambda_1}(\tilde{\boldsymbol{x}}^\intercal \boldsymbol{y})}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)} = \frac{\mathrm{S}_{\lambda_1}\left(\frac{\beta_j^* n(q-q^2)}{s_j}\right)}{s_j\left(\frac{n(q-q^2)}{s_j^2} + \lambda_2\right)}.$$

- Means that the elastic net estimator depends on class balance (q).
- $s_j = q - q^2$ for lasso and $s_j = \sqrt{q - q^2}$ for ridge removes effect of q.

Noiseless Case for Binary Features

In the noiseless case, we have

$$\hat{\beta}_j = \frac{\mathrm{S}_{\lambda_1}(\tilde{\boldsymbol{x}}^\intercal \boldsymbol{y})}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)} = \frac{\mathrm{S}_{\lambda_1}\left(\frac{\beta_j^* n(q-q^2)}{s_j}\right)}{s_j\left(\frac{n(q-q^2)}{s_j^2} + \lambda_2\right)}.$$

- Means that the elastic net estimator depends on class balance (q).
- $s_j = q - q^2$ for lasso and $s_j = \sqrt{q - q^2}$ for ridge removes effect of q.
- Suggests the parametrization

$$s_j = (q - q^2)^\delta, \qquad \delta \geq 0,$$

which we will rely on for the rest of the talk.

Noiseless Case for Binary Features

In the noiseless case, we have

$$\hat{\beta}_j = \frac{\operatorname{S}_{\lambda_1}(\tilde{\boldsymbol{x}}^\intercal \boldsymbol{y})}{s_j\left(\tilde{\boldsymbol{x}}_j^\intercal \tilde{\boldsymbol{x}}_j + \lambda_2\right)} = \frac{\operatorname{S}_{\lambda_1}\left(\frac{\beta_j^* n(q-q^2)}{s_j}\right)}{s_j\left(\frac{n(q-q^2)}{s_j^2} + \lambda_2\right)}.$$

- Means that the elastic net estimator depends on class balance (q).
- $s_j = q - q^2$ for lasso and $s_j = \sqrt{q - q^2}$ for ridge removes effect of q.
- Suggests the parametrization

  $$s_j = (q - q^2)^\delta, \qquad \delta \geq 0,$$

  which we will rely on for the rest of the talk.
- Indicates there might be no (simple) s_j that will work for the elastic net.

Probability of Selection

Since $\boldsymbol{X}$ is fixed and $\boldsymbol{\varepsilon}$ is normal, we can compute the probability of selection:

$$\Pr(\hat{\beta}_j \neq 0) = \Phi\left(\frac{\mu - \lambda_1}{\sigma}\right) + \Phi\left(\frac{-\mu - \lambda_1}{\sigma}\right).$$

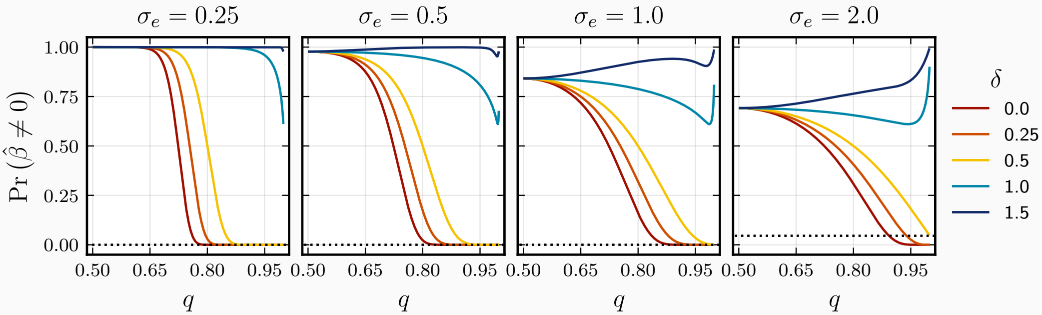


Figure 8: Probability that the elastic net selects a feature across different noise levels (σ_ε), types of normalization (δ), and class balance (q). The dashed line is asymptotic behavior for $\delta = 1/2$. Scaling used is $s_j \propto (q - q^2)^\delta$.

Implications

Rare Traits
Features with large class-imbalances might not be selected even if effect is very strong (e.g. rare SNPs, mutations).

Implications

Rare Traits

Features with large class-imbalances might not be selected even if effect is very strong (e.g. rare SNPs, mutations).

Subgroup Data

Results become dependent on data colection.

Collecting more data with different class balances influences the results (since class balances change).

Theorem

If $\boldsymbol{x}_j$ is a binary feature with class balance $q \in (0, 1)$ and $\lambda_1, \lambda_2 \in (0, \infty)$, $\sigma_\varepsilon > 0$, and $s_j = (q - q^2)^\delta$, $\delta \geq 0$, then

$$\lim_{q \to 1^+} \operatorname{E} \hat{\beta}_j = \begin{cases} 0 & \textit{if } 0 \leq \delta < \frac{1}{2}, \\ \frac{2n\beta_j^*}{n+\lambda_2} \Phi\left(-\frac{\lambda_1}{\sigma_\varepsilon \sqrt{n}}\right) & \textit{if } \delta = \frac{1}{2}, \\ \beta_j^* & \textit{if } \delta \geq \frac{1}{2}. \end{cases}$$

Asymptotic Results for Bias and Variance

Theorem

If $\boldsymbol{x}_j$ is a binary feature with class balance $q \in (0,1)$ and $\lambda_1, \lambda_2 \in (0, \infty)$, $\sigma_\varepsilon > 0$, and $s_j = (q - q^2)^\delta$, $\delta \geq 0$, then

$$\lim_{q \to 1^+} \operatorname{E} \hat{\beta}_j = \begin{cases} 0 & \text{if } 0 \leq \delta < \frac{1}{2}, \\ \frac{2n\beta_j^*}{n+\lambda_2} \, \Phi\left(-\frac{\lambda_1}{\sigma_\varepsilon \sqrt{n}}\right) & \text{if } \delta = \frac{1}{2}, \\ \beta_j^* & \text{if } \delta \geq \frac{1}{2}. \end{cases}$$

and

$$\lim_{q \to 1^+} \operatorname{Var} \hat{\beta}_j = \begin{cases} 0 & \text{if } 0 \leq \delta < \frac{1}{2}, \\ \infty & \text{if } \delta \geq \frac{1}{2}. \end{cases}$$

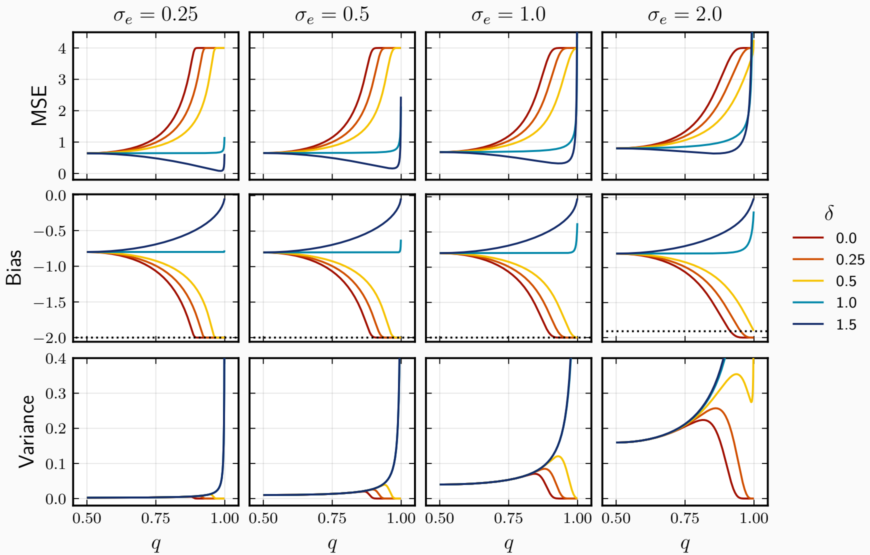
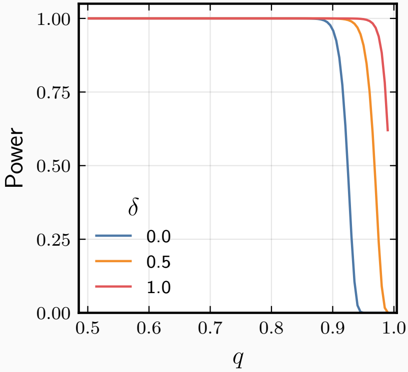
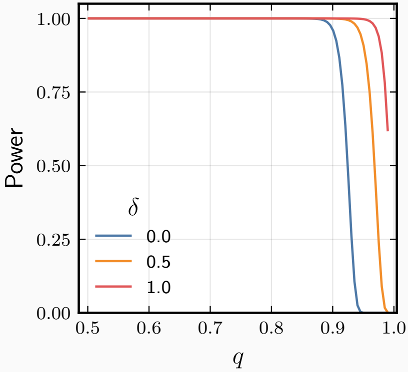


Figure 9: A bias variance tradeoff. Bias, variance, and mean-squared error for a one-dimensional lasso problem. Theoretical result for orthogonal features. Dotted line is asymptotic result or $\delta = 1/2$. Scaling used is $s_j \propto (q - q^2)^{\delta}$.

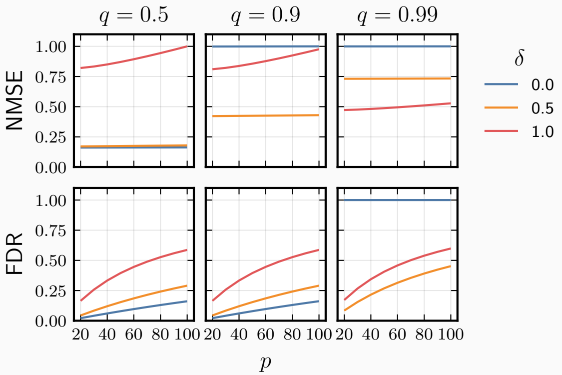


(a) Power in the sense of detecting all the true signals. Constant p.

Figure 10: Multiple features: 10 true signals and varying q and p. Mean squared error (MSE), false discovery rate (FDR), and power



(a) Power in the sense of detecting all the true signals. Constant p.



(b) False discovery rate (FDR) and normalized mean-squared error (NMSE).

Figure 10: Multiple features: 10 true signals and varying q and p. Mean squared error (MSE), false discovery rate (FDR), and power

Mixed Data

So far: all binary features. What about mixing binary and continuous (normal) features?

How to put binary features and normal features on the "same" scale?

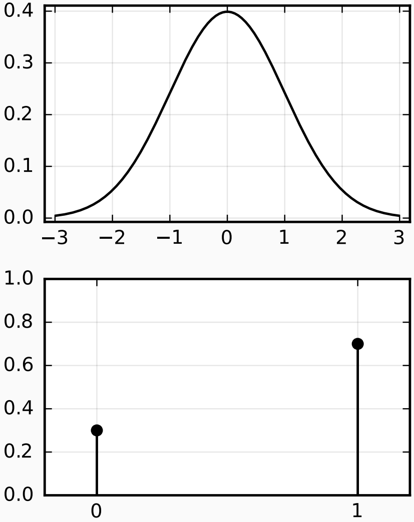


Figure 11: How do we match these?

Mixed Data

Our Definition of Comparability

The effects of a binary feature and a normally distributed feature are comparable if a flip in the binary feature has the same effect as a two-standard deviation change in the normal feature (Gelman 2008).

Our Definition of Comparability

The effects of a binary feature and a normally distributed feature are comparable if a flip in the binary feature has the same effect as a two-standard deviation change in the normal feature (Gelman 2008).

Examples

Assume entries in $\boldsymbol{x}_1$ are binary and $\boldsymbol{x}_2$ come from a random variable X_2. The effects are comparable in the following cases:

- $X_2 \sim \mathrm{Normal}(\mu, 1/2)$, $\beta_1^* = 1$, and $\beta_2^* = 1$.
- $X_2 \sim \mathrm{Normal}(\mu, 2)$, $\beta_1^* = 1$, and $\beta_2^* = 0.25$.

Choice of Scaling in Mixed Data

For the two-standard deviation notion of comparability to hold, we need to modify our scaling factor s_j.

Choice of Scaling in Mixed Data

For the two-standard deviation notion of comparability to hold, we need to modify our scaling factor s_j.

As before, we assume that $\boldsymbol{x}_1$ is binary and $X_2 \sim \mathrm{Normal}(\mu, 1/2)$, $\beta_1^* = \beta_2^* = 1$ so that they have *comparable* effects. Also assume we standardize $\boldsymbol{x}_2$.

We want $\hat{\beta}_1 = \hat{\beta}_2$. That is,

$$\underbrace{\frac{\mathrm{S}_{\lambda_1}\left(\frac{n(q-q^2)}{s_j}\right)}{s_1\left(\frac{n(q-q^2)}{s_1^2} + \lambda_2\right)}}_{\hat{\beta}_1} = \underbrace{\frac{\mathrm{S}_{\lambda_1}\left(\frac{n}{2}\right)}{\frac{1}{2}\left(n + \lambda_2\right)}}_{\hat{\beta}_2}.$$

Choice of Scaling in Mixed Data

For the two-standard deviation notion of comparability to hold, we need to modify our scaling factor s_j.

As before, we assume that $\boldsymbol{x}_1$ is binary and $X_2 \sim \mathrm{Normal}(\mu, 1/2)$, $\beta_1^* = \beta_2^* = 1$ so that they have *comparable* effects. Also assume we standardize $\boldsymbol{x}_2$.

We want $\hat{\beta}_1 = \hat{\beta}_2$. That is,

$$\underbrace{\frac{\mathrm{S}_{\lambda_1}\left(\frac{n(q-q^2)}{s_j}\right)}{s_1\left(\frac{n(q-q^2)}{s_1^2} + \lambda_2\right)}}_{\hat{\beta}_1} = \underbrace{\frac{\mathrm{S}_{\lambda_1}\left(\frac{n}{2}\right)}{\frac{1}{2}\left(n + \lambda_2\right)}}_{\hat{\beta}_2}.$$

The choice $s_1 = (2(q - q^2))^\delta$ works when classes are balanced ($q = 0.5$). But no clear choice for the elastic net case.

Experiments

Binary Features (Decreasing q)

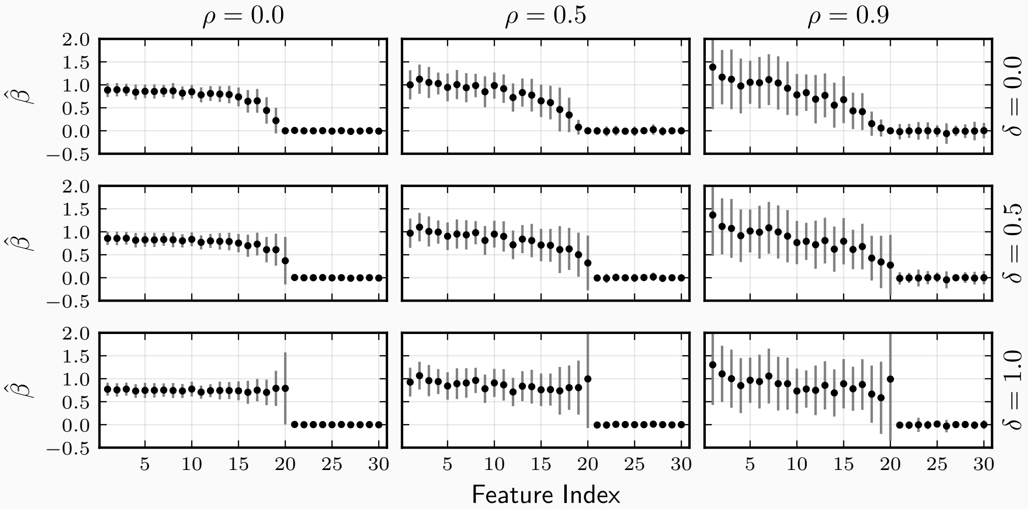


Figure 12: Lasso estimates for first 30 coefficients. First 20 features are true signals with a geometrically decreasing class balance from 0.5 to 0.99. ρ is a measure of autocorrelation.

Binary Features (Signal-to-Noise Ratio)

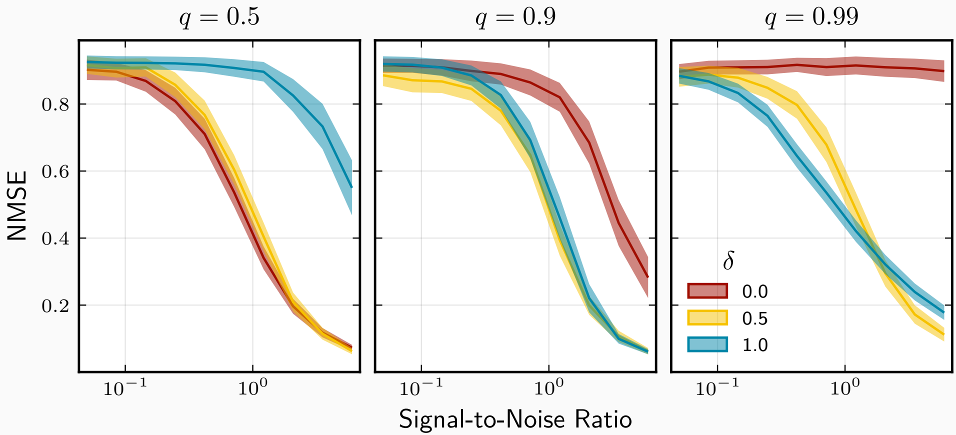


Figure 13: Normalized mean-squared test set error (NMSE).

Mixed Data

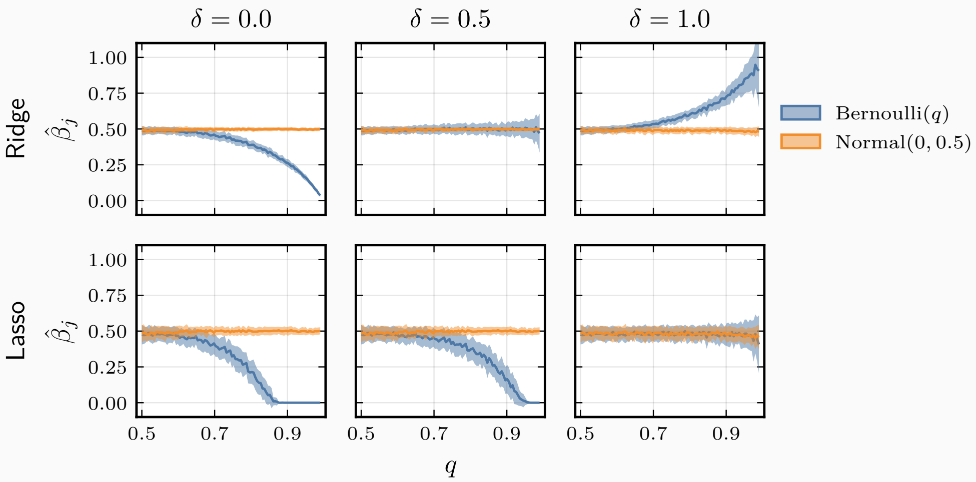


Figure 14: Comparison between lasso and ridge estimators for features generated to resemble features from various distributions.

Hyperparameter Optimization

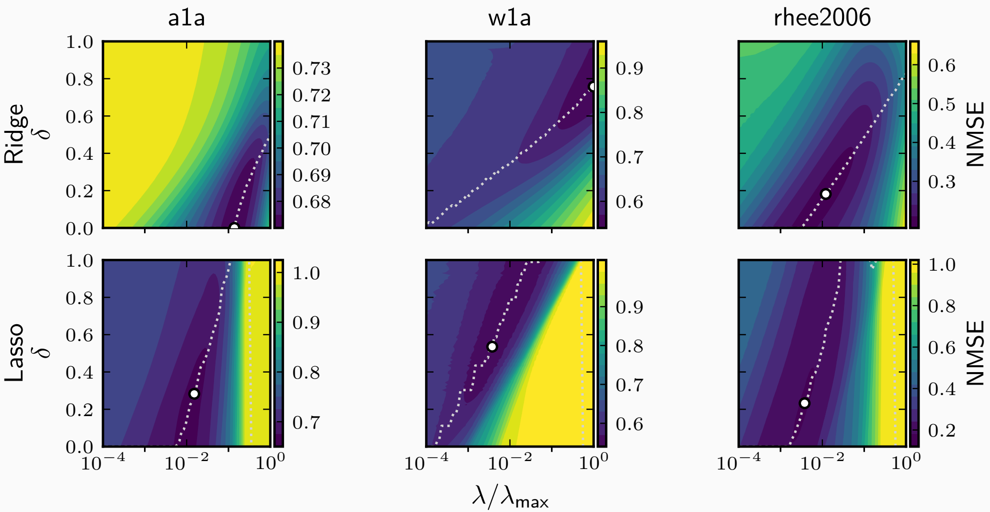


Figure 15: Contour plots of hold-out (validation set) error across a grid of δ and λ values for the lasso and ridge.

Summary

Conclusions

- Class balance plays a crucial role when using regularized regression on binary data.
- As far as we know the first paper to investigate the interplay between normalization and regularization
- New scaling approach to deal with class-imbalanced binary features
- Discussion and suggestions for dealing with mixed data

Summary

Conclusions

- Class balance plays a crucial role when using regularized regression on binary data.
- As far as we know the first paper to investigate the interplay between normalization and regularization
- New scaling approach to deal with class-imbalanced binary features
- Discussion and suggestions for dealing with mixed data

Limitations

- So far only theoretical results for limited cases:
  - Fixed data ($\boldsymbol{X}$), normal noise
  - Orthogonal features
  - Normal and binary features

Thank you!

- El Ghaoui, Laurent, Vivian Viallon, and Tarek Rabbani (Sept. 21, 2010). *Safe Feature Elimination in Sparse Supervised Learning*. Technical report UCB/EECS-2010-126. Berkeley: EECS Department, University of California. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-126.html.
- Gelman, Andrew (July 10, 2008). "Scaling Regression Inputs by Dividing by Two Standard Deviations". In: *Statistics in Medicine* 27.15, pp. 2865–2873. ISSN: 02776715, 10970258. DOI: 10.1002/sim.3107. URL: https://onlinelibrary.wiley.com/doi/10.1002/sim.3107 (visited on 09/27/2023).
- Tibshirani, Robert et al. (Mar. 2012). "Strong Rules for Discarding Predictors in Lasso-Type Problems". In: *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 74.2, pp. 245–266. ISSN: 1369-7412. DOI: 10/c4bb85. URL: https://iths.pure.elsevier.com/en/publications/strong-rules-for-discarding-predictors-in-lasso-type-problems (visited on 03/16/2018).

References ii

Zou, Hui and Trevor Hastie (2005). "Regularization and Variable Selection via the Elastic Net". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67.2, pp. 301–320. ISSN: 1369-7412. URL: www.jstor.org/stable/3647580 (visited on 03/12/2018).

Extras

Max–Abs Scaling of Continuous Features

- Min–max normalization is sometimes used in continuous data
- Very sensitive to outliers
- But also depend on sample size!
- In other words, results in model validation with varying sample sizes can yield very strange results.

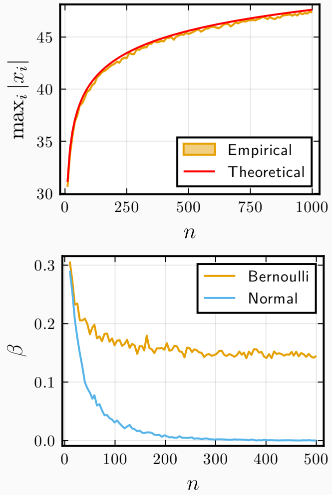


Figure 16: Effects of maximum absolute value scaling.

Hyperparameter Optimization (Support and NMSE)

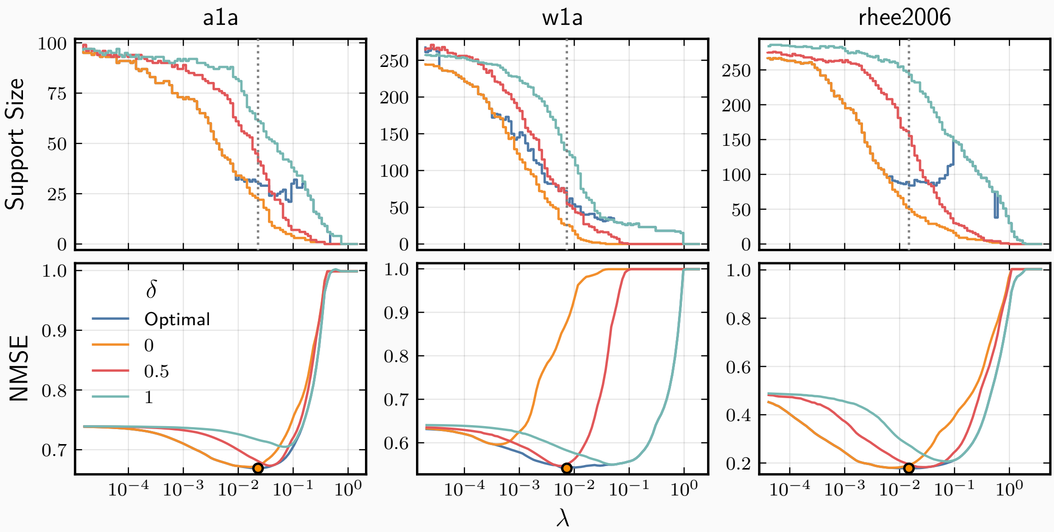


Figure 17: Support and NMSE of the lasso for different values of δ and λ.

Background on the Elastic Net

- Proposed by Zou and Hastie (2005).

Background on the Elastic Net

- Proposed by Zou and Hastie (2005).
- Lasso (ℓ_1) part:
  - Enables sparsity (interpretability, parsimony, feature selection)
  - Efficient when $p \gg n$ (due to screening rules (El Ghaoui, Viallon, and Rabbani 2010; Tibshirani et al. 2012))

Background on the Elastic Net

- Proposed by Zou and Hastie (2005).
- Lasso (ℓ_1) part:
  - Enables sparsity (interpretability, parsimony, feature selection)
  - Efficient when $p \gg n$ (due to screening rules (El Ghaoui, Viallon, and Rabbani 2010; Tibshirani et al. 2012))
- Ridge (ℓ_2) part
  - Mitigates lasso issue in correlated data
  - Better predictive performance when true signal is non-sparse