Ensemble of Regularized Linear Models

Ruben Zamar

Department of Statistics, University of British Columbia

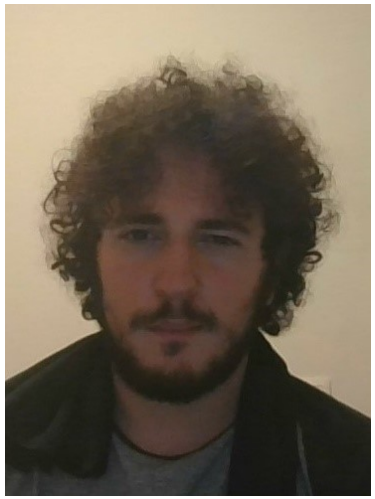
July 3, 2018

Joint work with

Laks Lakshmanan

Ezequiel Smucler

Anthony Christidis



THE CURSE OF DIMENSIONALITY

HAS BEEN WIDELY ACKNOWLEGED

BUT THE BLESSING OF DIMENSIONALITY

IS SELDOM APPRECIATED

Linear Regression

$$y_i = \mathbf{x}_i' \boldsymbol{\beta}_0 + \varepsilon_i, i = 1, \ldots, n$$

- y_i response.
- $\mathbf{x}_i \in \mathbb{R}^p$ vector with p predictors.
- ε_i random errors.
- $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ vector with p regression coefficients.

Notation

$$\text{Data} = (\mathbf{X}, \mathbf{y})$$

$$\underset{n\times p}{\mathbf{X}} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \qquad \underset{n\times 1}{\mathbf{y}} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$\underset{n\times p}{\mathbf{X}} = \begin{pmatrix} \mathbf{x}^1 & \mathbf{x}^2 & \cdots & \mathbf{x}^{p-1} & \mathbf{x}^p \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{pmatrix}$$

Centering and scaling

$$\frac{1}{n}\sum_{i=1}^{n} y_i = \frac{1}{n}\sum_{i=1}^{n} x_{ij} = 0 \qquad j = 1, ..., p$$

$$\frac{1}{n}\sum_{i=1}^{n} y_i^2 = \frac{1}{n}\sum_{i=1}^{n} x_{ij}^2 = 1 \qquad j = 1, ..., p$$

Least Squares (Gauss 1795)

The classical estimate is

$$\hat{\beta}_{LS} = \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i'\beta\right)^2 = \arg\min_{\beta \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\beta\|^2.$$

- Optimal when the errors are i.i.d. normal.
- Easy to compute.

High Dimension

Data-sets with $p > n$ are nowadays standard.

Many examples in fields like chemometrics, genomics and others.

Examples:

1. Reponse is the content of a chemical compound in an item, predictors are frequencies measured on a spectrum.

2. Response is the log survival time of patients suffering from a serious illness. Predictors are expression levels of several thousand genes.

Bias-variance trade-off

- Unless n is very large ($n/p > 20$, say) trading-off some bias for a decrease in variance may be reasonable.
- Larger models have less bias but more variance.

  Sparsity: many of the candidate variables included in the model are not very useful.
- A possible approach: fit the LS estimate to a reduced subset of predictors, but which one?

Best Subset Selection (Beale et al. 1967)

Fit LS to all possible subsets of predictors of size at most s, choose the fit with lowest estimated prediction error.

Requires fitting many LS estimates. Not feasible unless s is small.

Moreover, the procedure is unstable (see Breiman (1995)).

Lasso (Tibshirani 1996)

$$\hat{\boldsymbol{\beta}}_{Lasso} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i'\boldsymbol{\beta}\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|.$$
$$= \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \|\boldsymbol{\beta}\|_1.$$

- Regularizes the LS estimate, well defined even if $p > n$.
- The penalty term shrinks the coefficients towards zero.
- λ regulates the shrinkage; usually chosen using cross-validation. Bias - Variance trade-off.

Can we do better?

- Both LS and the Lasso estimate *a single model*.
- Optimally regularized models may not be able to take full advantage of the richness in the data.
- In cases with a very high number of correlated predictors, prediction accuracy may be improved by fitting several models to the data and aggregating them.

A toy model

$$y = 0x_1 + x_2 + x_3 + \varepsilon$$

1. $(\varepsilon, x_1, x_2, x_3)$ are jointly normal
2. $Cov(\varepsilon, x_j) = 0, \;\; j = 1, 2, 3$
3. $Cov(x_1, x_2) = Cov(x_1, x_3) = 0$
4. $Cov(x_2, x_3) = 0.90$
5. $Var(\varepsilon) = Var(x_1) = Var(x_2) = Var(x_3) = 1$

A toy numerical experiment

Generate 5000 independent observations from the model (test sample)

Repeat the following 500 times:

- Generate a sample of 10 independent observations from the model (training sample)
- Predict the test sample using each of the following procedures
  1. Ordinary least squares (OLS)
  2. Elastic net with cross-validated tuning parameter (ENET)
  3. Ensemble of
     - OLS using x_1 and x_2
     - OLS using x_3

Performance evaluation

$\widehat{y}_{ki}^{\{\text{OLS}\}} =$ prediction for y_i $(i = 1, ..., 5000)$ using OLS and the k^{th} training sample.

$$\text{PMSE}_k^{\text{OLS}} = \frac{1}{5000} \sum_{i=1}^{5000} \left(\widehat{y}_{ki}^{\{\text{OLS}\}} - y_i \right)^2$$

$$\text{PMSE}^{\text{OLS}} = \frac{1}{500} \sum_{k=1}^{500} \text{PMSE}_k^{\text{OLS}}$$

Similarly, we compute $\text{PMSE}^{\text{ENET}}$ and PMSE^{ENS}.

Results

PREDICTION METHOD	PMSE
OLS	1.74
ELASTIC NET	2.09
ENSEMBLE	1.33

Intuitive explanation of results

- In each ensembled model, a reduction in variance due to:
  1. lower dimensionality
  2. less multicollinearity
- An additional reduction of variance in the ensemble of the models due to the averaging of nearly uncorrelated predictions
- A big relative increase of bias in the ensemble model,
- Decisive dominance of variance over bias.

	LS	ESEMBLE
Average Variance	0.74	0.32
Average Bias	0.0026	0.0094

Cheating?

- We have cheated in the "toy example". Why?
- Because we have used our knowledge of the true model to form the ensemble.

Search for an "optimal" ensemble

Suppose the number of ensembled models, G, is equal to two.

Even in this simple case we must evaluate a large number of possible splits/models:

Model 1	Model 2	Left-Out Variables
$x_{i_1}, x_{i_2}, ..., x_{i_{p_1}}$	$x_{j_1}, x_{j_2}, ..., x_{j_{p_2}}$	$x_{k_1}, x_{k_2}, ..., x_{k_{p_3}}$
$\beta^1_{i_1}, \beta^1_{i_2}, ..., \beta^1_{i_{p_1}}$	$\beta^2_{j_1}, \beta^2_{j_2}, ..., \beta^2_{j_{p_2}}$	$0, \ 0, \ \ldots \ , 0$

Search for an "optimal" ensemble

$$G = 2 \quad \text{and} \quad p_1 + p_2 + p_3 = p \quad \text{(no overlap)}$$

$$\#\ \textbf{possible ensemblings} = 3^p$$

$$G \geq 2 \quad \text{and} \quad p_1 + p_2 + \cdots + p_{G+1} = p \quad \text{(no overlap)}$$

$$\#\ \textbf{possible ensemblings} = (G+1)^p$$

$$G \geq 2 \quad \text{and} \quad p_1 + p_2 + \cdots + p_{G+1} > p \quad \text{(allowing overlap)}$$

$$\#\ \textbf{possible ensemblings} = (G^p+1)^p$$

Notation

$$\underset{n\times G}{\mathbf{Y}} = \begin{pmatrix} \mathbf{y} & \mathbf{y} & \cdots & \mathbf{y} \end{pmatrix} = \begin{pmatrix} y_1 & y_1 & \cdots & y_1 \\ y_2 & y_2 & \cdots & y_2 \\ \vdots & \vdots & & \vdots \\ y_n & y_n & \cdots & y_n \end{pmatrix}$$

$$\underset{p\times G}{\boldsymbol{\beta}} = \begin{pmatrix} \beta_1^1 & \beta_1^2 & \cdots & \beta_1^G \\ \vdots & \vdots & & \vdots \\ \beta_p^1 & \beta_p^2 & \cdots & \beta_p^G \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta}^1 & \boldsymbol{\beta}^2 & \cdots & \boldsymbol{\beta}^G \end{pmatrix}$$

A non-convex relaxation

Minimize

$$O(\mathbf{y}, \mathbf{X}, \beta) =$$

$$\sum_{g=1}^{G} \left(\frac{1}{2n} \|\mathbf{y} - \mathbf{X}\beta^g\|^2 + p_{\lambda_S}(\beta^g) + q_{\lambda_D,g}(\beta^1, \ldots, \beta^G) \right),$$

where

- G number of models, allowing overlap.
- $\beta^g \in \mathbb{R}^p$ coefficients for model g
- p_{λ_S} penalty function (**sparsity**)
- $q_{\lambda_D,g}$ penalty function (**diversity**)

A non-convex relaxation

For example,

$$p_{\lambda_S}(\boldsymbol{\beta}^g) = \|\boldsymbol{\beta}^g\|_1, \qquad \text{(LASSO penalty)}$$

and

$$q_{\lambda_D,g}(\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^G) = \frac{\lambda_D}{2} \sum_{h \neq g} \sum_{j=1}^{p} |\beta_j^h \beta_j^g|.$$

Looking at the terms for each single model

$$O_g(\mathbf{y}, \mathbf{X}, \beta) =$$

$$= \frac{\|\mathbf{y} - \mathbf{X}\beta^g\|^2}{2n} + \lambda_S \|\beta^g\|_1 + \frac{\lambda_D}{2} \sum_{h \neq g} \sum_{j=1}^{p} |\beta_j^h \beta_j^g|$$

$$= \frac{\|\mathbf{y} - \mathbf{X}\beta^g\|^2}{2n} + \sum_{j=1}^{p} |\beta_j^g| (\lambda_S + \frac{\lambda_D}{2} \sum_{h \neq g} |\beta_j^h|)$$

$$= \frac{\|\mathbf{y} - \mathbf{X}\beta^g\|^2}{2n} + \sum_{j=1}^{p} |\beta_j^g| w_{j,g}$$

with

$$w_{j,g} = (\lambda_S + \frac{\lambda_D}{2} \sum_{h \neq g} |\beta_j^h|)$$

Coordinate descent

At each step in the coordinate descent algorithm:

- We solve an Elastic Net type problem, where the weight of the L_1-penalty depends on the current solution
- Hence, each step in the coordinate-descent algorithm is a convex minimization problem
- The coordinates most penalized in model g are those that have large coefficients in the other models

An R package that implements the procedures presented in this talk, called ensembleEN is available from https://github.com/esmucler/ensembleEN.

The diversity penalty λ_D

- To gain some intuition about our **Diversity Penalty,** λ_D , we consider an extreme situation:
  - $p = 1$, $G = 3$, and $\lambda_S = 1$
  - Surface level plot: Find the values of $\left(\beta_1^1, \beta_1^2, \beta_1^3\right)$ that satisfy the equation:

$$|\beta_1^1| + |\beta_1^2| + |\beta_1^3| + \lambda_D \left(|\beta_1^1 \beta_1^2| + |\beta_1^1 \beta_1^3| + |\beta_1^3 \beta_1^2|\right) = 1.$$

Surfaces for different values of λ_D

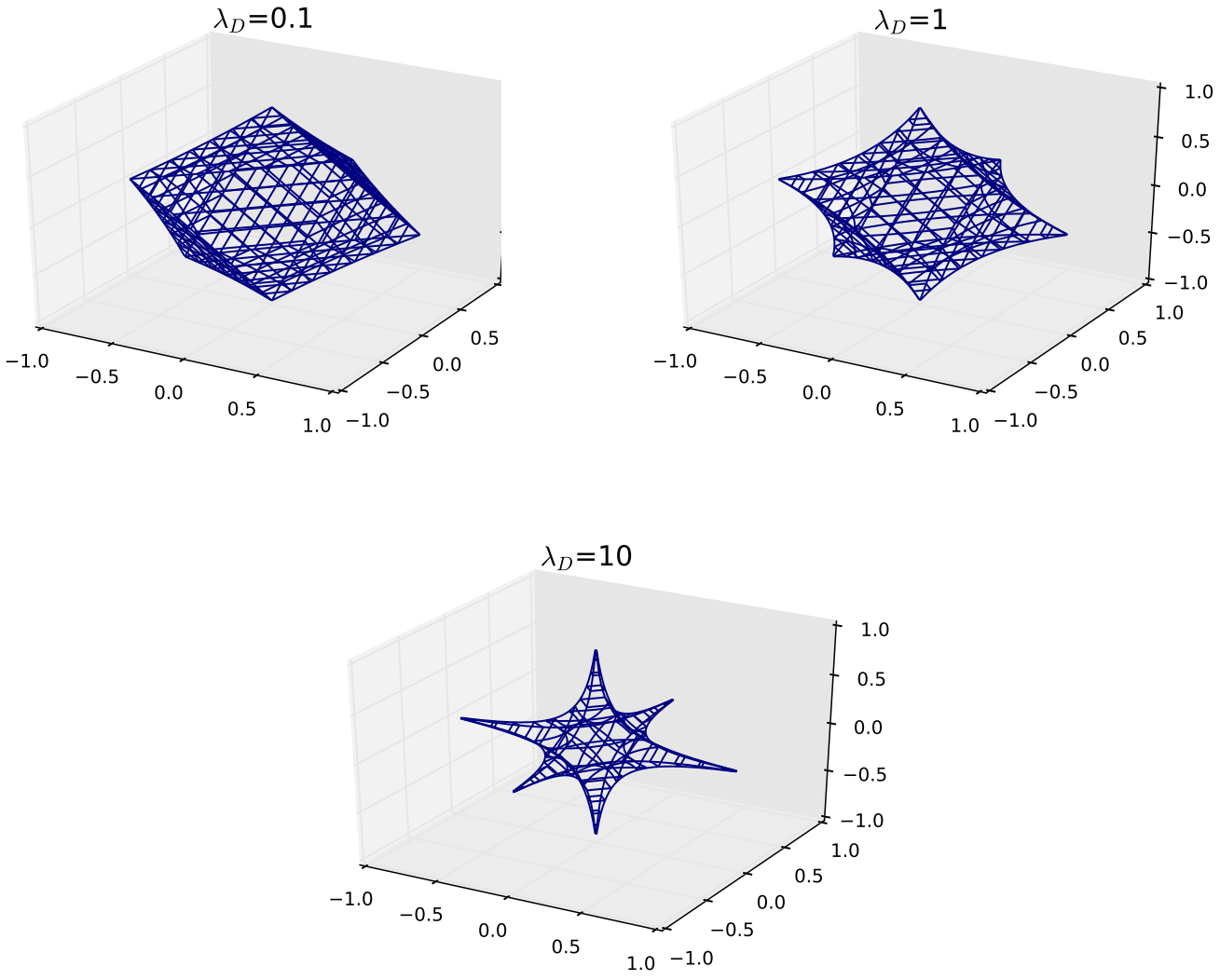


Figure: Plots of the full penalty term for $\lambda_S = 1$ and three different values of λ_D.

The objective function using matrix notation

The minimization problem can be posed as an 'artificial' multivariate linear regression problem:

$$O(\mathbf{y}, \mathbf{X}, \beta) =$$

$$\frac{1}{2n}\|\mathbf{Y}-\mathbf{X}\beta\|_F^2+\lambda_S\left(\frac{(1-\alpha)}{2}\|\beta\|_F^2+\alpha\|\beta\|_1\right)+\frac{\lambda_D}{2}\left(\||\beta|'|\beta|\|_1-\|\beta\|_F^2\right),$$

where

- $\|\cdot\|_F$ is the Frobenius norm,
- $|\beta|$ is the absolute value coordinate-wise and
- $\|\cdot\|_1$ is the sum of the absolute values of the matrix entries.
- **The diversity penalty penalizes correlation between the different models**

Aggregating the final predictions

- Let $\hat{\boldsymbol{\beta}} = \left(\hat{\boldsymbol{\beta}}^1, \hat{\boldsymbol{\beta}}^2, ..., \hat{\boldsymbol{\beta}}^G\right)$ be the ensemble problem solution.
- Let $\hat{y}^g = \mathbf{x}'\hat{\boldsymbol{\beta}}^g$ $(g = 1, ..., G)$ be the g^{th} model prediction.

We can aggregate the predictions $(\hat{y}^1, ..., \hat{y}^G)$ in several ways:

- Plain average prediction (used in this work)
- Weighted average prediction (weights proportional to each model precision)
- Stacking (Breiman 1996)

Plain Average

$$\hat{y} = \frac{1}{G} \sum \hat{y}^g$$

$$= \frac{1}{G} \sum \mathbf{x}' \hat{\beta}^g$$

$$= \mathbf{x}' \left(\frac{1}{G} \sum \hat{\beta}^g \right)$$

$$= \mathbf{x}' \hat{\beta}^*$$

$$\hat{\beta}^* = \frac{1}{G} \sum \hat{\beta}^g$$

Weighted Average

$$\hat{y} = \frac{\sum w_g \hat{y}^g}{\sum w_j}$$

$$= \frac{\sum w_g \mathbf{x}' \hat{\boldsymbol{\beta}}^g}{\sum w_j}$$

$$= \mathbf{x}' \left(\frac{\sum w_g \hat{\boldsymbol{\beta}}^g}{\sum w_j} \right)$$

$$= \mathbf{x}' \hat{\boldsymbol{\beta}}^{\#}$$

$$\hat{\boldsymbol{\beta}}^{\#} = \frac{\sum w_j \hat{\boldsymbol{\beta}}^g}{\sum w_j}$$

Stacking

- $\hat{\beta}^g_{(i)}$ $(g = 1, ..., G)$ coefficients computed leaving out i^{th} case
- $\hat{y}^g_i = \mathbf{x}'_i \hat{\beta}^g_{(i)}$

$\hat{y}^1_1$	$\hat{y}^2_1$	$\cdots$	$\hat{y}^G_1$	y_1
$\hat{y}^1_2$	$\hat{y}^2_2$	$\cdots$	$\hat{y}^G_2$	y_2
$\vdots$	$\vdots$		$\vdots$	$\vdots$
$\hat{y}^1_n$	$\hat{y}^2_n$	$\cdots$	$\hat{y}^G_n$	y_n

Stacking

$$J(\boldsymbol{\alpha}) = \sum \left[y_i - \sum \alpha_g \hat{y}_i^g \right]^2$$

$$\widehat{\boldsymbol{\alpha}} = \arg \min_{\alpha_i \geq 0, \sum \alpha_i^2 = 1} J(\boldsymbol{\alpha})$$

$$\hat{\boldsymbol{\beta}}^s = \sum \widehat{\alpha}_g \widehat{\beta}^g$$

Application to Chemometric data

- The glass data set (Lamberge et al., 2000) was obtained from an electron probe X-ray microanalysis of archaeological glass samples
- The spectrum on 486 frequencies was measured on a total of 180 glass samples
- The goal is to predict the concentrations of several chemical compounds using the spectrum

Application to Chemometric data

- We randomly split the data into a training set that has 50% of the observations and a testing set that has the remaining 50%.
- for this example we used $G = 10$
- This procedure is repeated 500 times and the resulting prediction MSEs are averaged.
- MSEs are reported relative to the best method.

Application to Chemometric data

	Na2O	MgO	Al2O3	SO3	Cl
Lasso	1.17	1.10	1.22	1.12	1.36
Ens-Lasso	1.00	1.00	1.00	1.00	1.00

Table: Average relative PMSEs over 500 random splits into training and testing sets

Tuning Parameters

- The values of the penalties, λ_S and λ_D can be chosen by cross-validation.
- We find that increasing the number of models G does not, in general, leads to overfitting
- We then recommend using the largest computationally convenient value for G.

Simulation

1. The **Lasso**, using the package `glmnet`.
2. The **Elastic Net** with $\alpha = 3/4$, using the package `glmnet`.
3. The sure independence screening (SIS), followed by fitting a SCAD penalized least squares estimator, computed using the package **SIS-SCAD**.
4. The MC+ penalized least squares estimator, using the package **SparseNet**.
5. The Relaxed Lasso, using the package **Relaxed**.
6. The forward stepwise algorithm, using the package, called **Stepwise**.
7. The Cluster Representative Lasso, proposed in using code **CRL** kindly provided by Buhlmann.
8. Random Forest of, using the package **RF**.
9. The Random GLM method of using the package **RGLM**.

Some simulation results

- We generate 500 replications of a linear model with normal predictors and errors, $p = 1000$ and $n = 50$.

  $\beta_0 = (2, 2, 2, \ldots, 0, 0)$, the blocks of 2's has length $[1000\zeta]$.

- The active variables are correlated only with each other, everything else is uncorrelated.

Results

SNR		$\zeta = 0.05$		$\zeta = 0.1$		$\zeta = 0.2$	
		PMSE	SE	PMSE	SE	PMSE	SE
3	Lasso	1.55	0.01	1.46	0.01	1.40	0.01
	Ens-Lasso	1.35	0.01	1.24	0.01	1.18	0.01
10	Lasso	2.30	0.02	2.03	0.01	1.90	0.01
	Ens-Lasso	1.85	0.01	1.53	0.01	1.35	0.01

Table: Mean PMSEs and standard errors for Scenario 1 with ρ =0.2, n=100, p=1000.

A consistency result

Theorem

Assume

- ε_i *are i.i.d. zero mean normals.*
- $\log(p_n)/n \to 0$.
- $\|\beta_0\|_1 = o(\sqrt{n/\log(p_n)})$.

Then there exist sequences of penalty parameters λ_S^n *and* λ_D^n *such that*

$$\frac{1}{n}\left\|\left(\frac{1}{G}\sum_{g=1}^{G}\mathbf{X}\hat{\beta}^g\right) - \mathbf{X}\beta_0\right\|_2^2 \to 0 \text{ in probability.}$$

Some questions

- Non-convex optimization problem $\rightarrow$ no guarantees for convergence. Is there a convex relaxation?
- Can we, in theory, guarantee better predictions than the Lasso?

Some possible extensions

- GLMs, for example logistic regression: replace quadratic loss with logistic loss.
- Other sparsity penalties: SCAD, MC+.
- Robustness to outliers: replace squared loss by a bounded loss function.

Software/further reading

- An R package called ensembleEN implementing the method is available from CRAN.
- The paper this talk is based on is available on arXiv.

Thank you