

Outline: High Dimensional Supervised Problems (F. Chiaromonte)

Statistical Methods for Large, Complex Data

In the reference books:

Linear and Generalized Linear Models

- ISLR Chpt 3, 4
- CASI Chpt 8

Ridge, LASSO

- ISLR Chpt 6
- CASI Chpt 7, 16

Cross-validation

- ISLR, Chpt 5
- CASI, Chpt 12

A QUICK REVIEW OF SOME CONCEPTS AND NOTATION CONCERNING LINEAR MODELS

A classical **linear model** expresses a continuous response Y through an additive function comprising

- an intercept (constant)
- p terms (linear in the slope parameters)
- a random error independent from the random mechanisms underlying the terms.

On a generic observation “ i ”:

$$y_i = \beta_o + \beta' x_i + \varepsilon_i$$

Terms: $X = (X_1 \dots X_p)'$ features (predictors) or specified transformations and functions of the features (e.g. powers, products; observable).

Coefficient parameters: β_o intercept, $\beta = (\beta_1 \dots \beta_p)'$ slopes for each term.

Regression function: the conditional expected value $E(Y | X_1 \dots X_p) = \beta_o + \beta' X$

Independent random error: added to the regression function and assumed to have mean $E(\varepsilon)=0$ and the same variance parameter $\text{var}(\varepsilon)=\sigma^2$ on all observations (homoschedasticity).

In order to develop **inferential procedures**, the random error is also often assumed to be Gaussian:

$$\varepsilon \sim N_1(0, \sigma^2)$$

(T-based confidence intervals and tests for the slope coefficients and the mean response, T-based prediction intervals, F-based ANOVA tests).

The observations are assumed to be iid from the joint distribution of $(Y, X_1 \dots X_p)$. If $X_1 \dots X_p$ are viewed as fixed (conditioning; not random), the errors are assumed to be iid across observations. Thus one has, for the vector of errors associated to the n observations in the sample:

$$\underline{\varepsilon}_{(n)} \sim N_n(0, \sigma^2 I)$$

Model in **matrix notation**:

$$\underline{Y}_{(n)} = \underline{1}_{(n)}\beta_o + \underline{X}_{(n,p)}\beta + \underline{\varepsilon}_{(n)}$$

$$\text{for simplicity } \underline{Y}_{(n)} = \underline{X}_{(n,p+1)}\beta + \underline{\varepsilon}_{(n)}$$

$$\underline{X}_{(n,p+1)} = (\underline{1}_{(n)} \quad \underline{X}_{(n,p)}) \quad \beta = \begin{pmatrix} \beta_o \\ \beta \end{pmatrix}$$

Estimating model parameters (fitting):

Because of linearity in the coefficient parameters, whatever the terms represent, fitting can be performed through **least squares** with an explicit, close form solution.

An estimate of the error variance is obtained dividing the minimized sum of squared deviations (Residual Sum of Squares; RSS) by the appropriate number of degrees of freedom.

$$\hat{\beta} = \operatorname{argmin} \|\underline{Y} - \underline{X}\beta\|^2 = (\underline{X}'\underline{X})^{-1} \underline{X}'\underline{Y}$$

$$\hat{\sigma}^2 = \frac{1}{n - (p + 1)} \|\underline{Y} - \underline{X}\hat{\beta}\|^2 = \frac{1}{n - (p + 1)} RSS$$

Implemented in most statistical software packages, including R. See <https://stat.ethz.ch/R-manual/R-devel/library/stats/html/lm.html>

* Intercept = “slope” of the constant term 1 (or assume response is centered, intercept = 0)

If the model is correct (i.e. not misspecified, though some of the coefficients may in fact be 0) the LS coefficient estimators, as well as the LS estimator of the error variance, are **unbiased** for the corresponding parameters.

$$E(\hat{\beta}) = \beta \quad E(\hat{\sigma}^2) = \sigma^2$$

Moreover, again if the model is not misspecified, the LS coefficient estimators are **BLUE** i.e. the best (most accurate; minimum sampling variance) among unbiased estimators expressed as linear functions of the response observations – **Gauss-Markov Theorem**.

Finally, if one assumes Gaussian error, the LS coefficient estimators coincide with the **Maximum Likelihood (ML)** estimators – and the ML estimator for the error variance, which is biased, divides the minimized sum of squares by n instead of the degrees of freedom.

Why? The exponent of the Gaussian likelihood is inversely proportional to the sum of squared deviations:

$$L(\beta | \underline{Y}; \underline{X}) \propto \exp \left\{ -\frac{1}{2\sigma^2} \| \underline{Y} - \underline{X}\beta \|^2 \right\}$$

Residuals diagnostics and remedial measures:

Lots of techniques that, based on residuals (observable proxies for the unobservable errors) aim to detect departures from

$$\underline{\varepsilon}_{(n)} \sim N_n(0, \sigma^2 I)$$

i.e. Gaussian iid random errors with 0 mean and shared variance σ^2 .

Lots of approaches to manipulate the data, e.g.:

- variance stabilizing transformations of the response to address heteroschedasticity
- identification and removal of outliers/influential observations (*case diagnostics*)

and the model specification, e.g.:

- add/remove terms to address mean patterns in the residuals

to come closer to meeting the assumptions.

Also important, **Multicollinearity**:

Linear dependencies among predictors increase the variance in effect estimates, may even make LS solution unstable.

Diagnose through

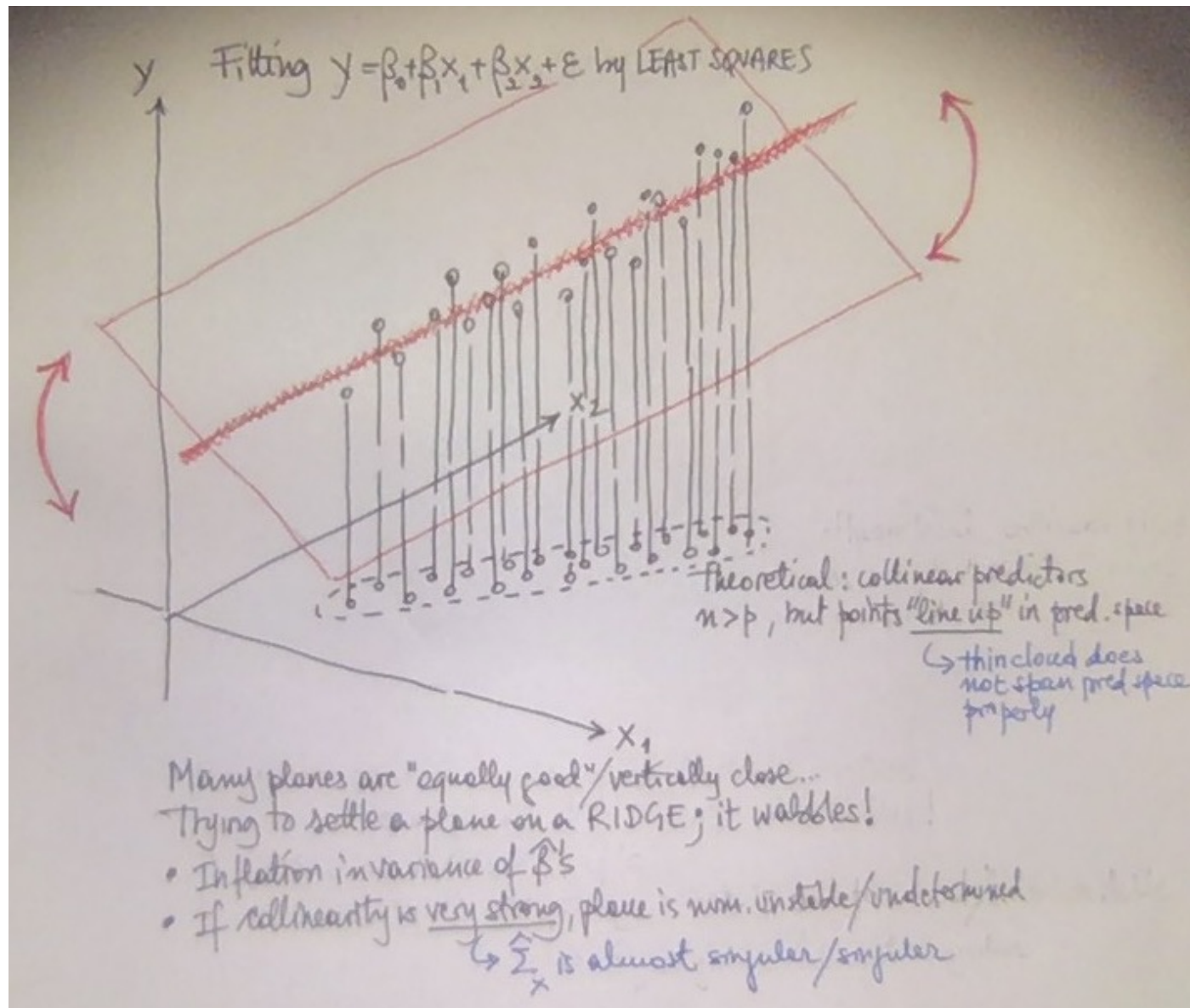
- Pair-wise correlations or scatter-plot matrices
- More complete, **partial R-squares** and **Variance Inflation Factors** (do not depend on the response)

$$R_{X_j | \text{other } Xs}^2 \qquad VIF_j = \frac{1}{1 - R_{X_j | \text{other } Xs}^2}$$

Remedies

- stabilize the fit (**Ridge**)
- eliminate some predictors (**LASSO**, Best Subsets – feature selection)
- reduce dimension creating “composite” predictors (unsupervised, e.g. Principal Components Analysis; supervised, e.g. Sliced Inverse Regression)

... coming next



The OMITTED VARIABLE BIAS (OVB) phenomenon

Important focus for Economics (less so for other fields): Trying to move towards causation, parsing controllable/exogenous and endogenous features. Related emphasis on model misspecification. Suppose an omitted feature Z

- (i) affects the response (non-zero coefficient in the true model), and
- (ii) correlates with one or more of the X's in the working model (non-zero $\text{Cov}(X,Z)$)

then:

$$\begin{array}{ll} \underline{Y} = \underline{X}\beta^* + Z\beta_Z + \underline{\varepsilon}^* & \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{true model} \\ \underline{\varepsilon}^* : E(\underline{\varepsilon}^*) = 0, \text{Cov}(\underline{\varepsilon}^*) = \theta^2 I, \text{ indep of } X, Z & \\ \underline{Y} = \underline{X}\beta + \underline{\varepsilon} & \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{working model} \\ \underline{\varepsilon} = Z\beta_Z + \underline{\varepsilon}^* : E(\underline{\varepsilon}) \neq 0, \text{Cov}(\underline{\varepsilon}) \neq \sigma^2 I, \text{ not indep of } X & \\ E(\hat{\beta}) = \beta^* + \underbrace{(\underline{X}'\underline{X})^{-1}\underline{X}'Z\beta_Z}_{\text{BIAS (fct of Cov}(X,Z))} & \end{array}$$

- The error of the working model includes Z which correlates with the X's; thus, this error is not independent from the random mechanism underlying the X's.
- The LS coefficient estimators for the features in the working model are biased; they subsume parts of the effects of the omitted Z through its correlations with the X's.
- The bias does not vanish with increasing n – causing inconsistency.

IMPORTANT REMARKS:

- Omitting anything that does not carry interdependencies with X's may induce a bigger and/or non-spherical error variability in the model – but does not introduce a bias in the LS estimators of the coefficients of the X's.
- Including more features in the model reduces the risk of OVB – but inflates the variability of the LS coefficients estimators. *Under-specifying vs overfitting; a variance/bias trade-off.*

Generalizations:

Linear models can be extended to comprise **categorical predictors**, properly encoding (dummies) their main effects and interactions with the continuous predictors.

In **Generalized Linear Models** a link function is introduced to represent the dependence of Y on $X_1 \dots X_p$, and the stochasticity of Y given $X_1 \dots X_p$ is modeled differently, not through an additive random error.

- Counts (Poisson regression)
- Binary labels (logit or Binomial regression; probit regression)
- Multi-class labels (Multinomial regression)

All (including the standard Normal regression) are implemented in the R package **GLM**

<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html>

For instance, Logistic regression
$$g(x_i) = \ln \left(\frac{p(x_i)}{1 - p(x_i)} \right) = \beta_0 + \beta' x_i \quad y_i \sim \text{Bernoulli}(p(x_i))$$

Note: $p(X) = E(Y | X_1 \dots X_p)$. LS estimation is replaced by ML estimation – often implemented numerically; more computation.

In linear models for time series or spatial data, special covariance structures for the errors are introduced to represent the dependence among observations: $\underline{\varepsilon}_{(n)} \sim N_n(0, \Sigma(\eta))$. LS estimation can be adapted to account for such dependence; **Generalized Least Squares**. More sophisticated estimation approaches (based on ML or Bayesian techniques) also exist; more computation necessary.

LOGISTIC REGRESSION

Binary Y, encoded (arbitrarily) as {0,1}

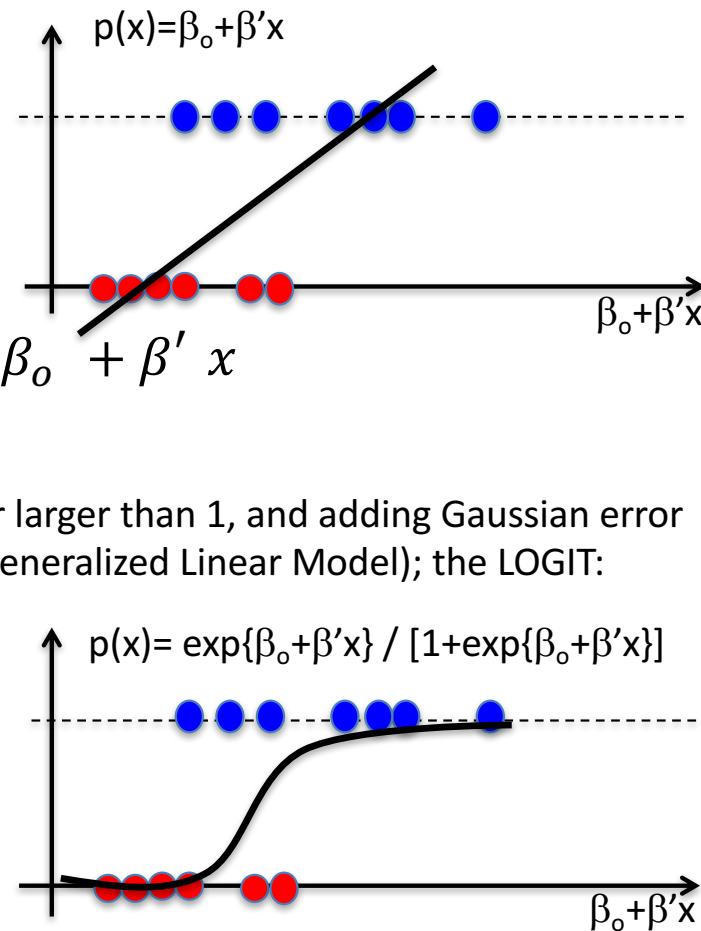
The standard Linear Model would be:

$$E(Y|x) = \Pr(Y = 1|x) = p(x) = \beta_o + \beta' x$$

$$Y|x \sim p(x) + \varepsilon, \varepsilon \sim N(0, \beta\sigma^2)$$

... can produce $p(x)$ estimates smaller than 0 or larger than 1, and adding Gaussian error makes no sense. Instead, use a *link function* (Generalized Linear Model); the LOGIT:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_o + \beta' x$$



Log of the *odds ratio* of a “1” modeled as a linear function of x.

Stochastic component in the mechanism producing the data is represented through a Bernoulli scheme, not through a Gaussian error about the expected value.

Parameters are estimated by *maximum likelihood*; $p(x)$ can be predicted for any level of x :

$$\hat{p}(x) = \frac{\exp\{\hat{\beta}_0 + \hat{\beta}'x\}}{1 + \exp\{\hat{\beta}_0 + \hat{\beta}'x\}}$$

and the label can be predicted (classification) based on whether this is > 0.5 (a threshold).

Remark: likewise a standard linear model, some of the features included as predictors could be *binary or categorical* (through dummy variables).

Remark: MLEs in Logistic regression are highly *unstable* when the two classes are well separated. In these cases it is preferable to use other classification algorithms.

We have reviewed a very powerful, rich and versatile framework for supervised data analysis.

Why do we need to go beyond it?

Even when all the assumptions comprised in the modeling are right, so in particular the LS coefficient estimators are unbiased (no omitted variables), their accuracy (notwithstanding the Gauss-Markov Theorem) may deteriorate severely when the feature space is large relative to the (iid) data at our disposal...

When does this happen?

When the features (terms) have strong linear associations to each other (multicollinearity) and/or n is not very large relative to p (possibly $n < p$) the data cloud does not “span” the feature space properly; it is “thin”, possibly it collapses in lower dimension.

As a consequence

- The LS coefficient estimators undergo variance inflation; if the data collapses (almost collapses) in lower dimension the estimates are non-uniquely determined (numerically unstable)
- Relatedly, we run into overfitting; the in-sample MSE may be very small, but the out-of-sample MSE is very large.

To overcome this problem: *constrain the LS as to reduce the estimators' variance and emiliorate overfitting*.

This *introduces a bias*, but the bias may be minor relative to the gain in variance – so that accuracy overall improves.

Additionally, constraining may result in a smaller, more *parsimonious and interpretable model* (some features are eliminated).

The same approach can be extended to the case of Generalized Linear Models, e.g., a logistic or multinomial regression for binary or multi-class classification. Here the ML (as opposed to the LS) is constrained.

Note: in classification problems, as a measure of accuracy one considers in-sample and out-of-sample misclassification rates.

- **Shrinkage/Regularization**

- Ridge Regression
- LASSO Regression

Penalized version of LS produces an alternative estimator.

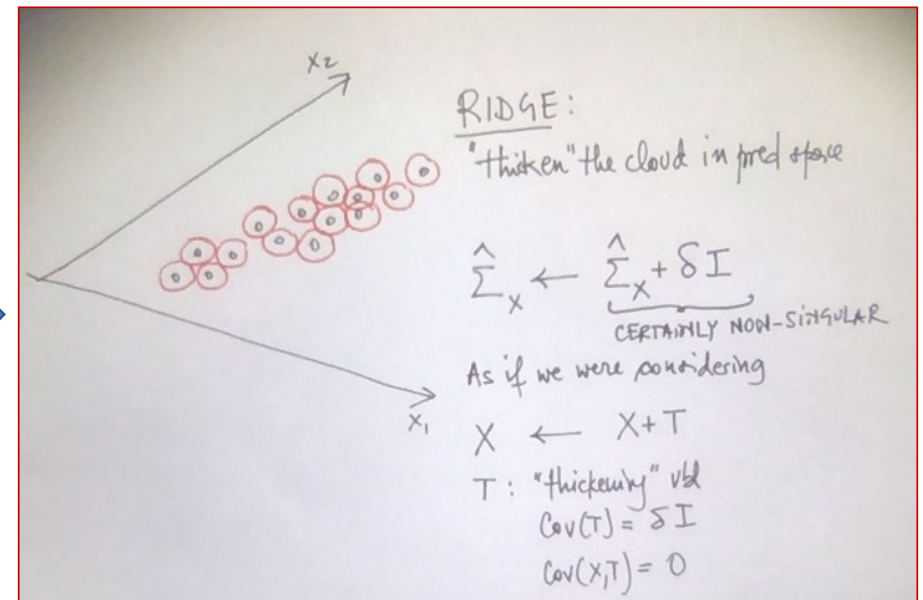
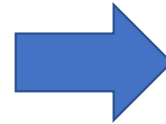
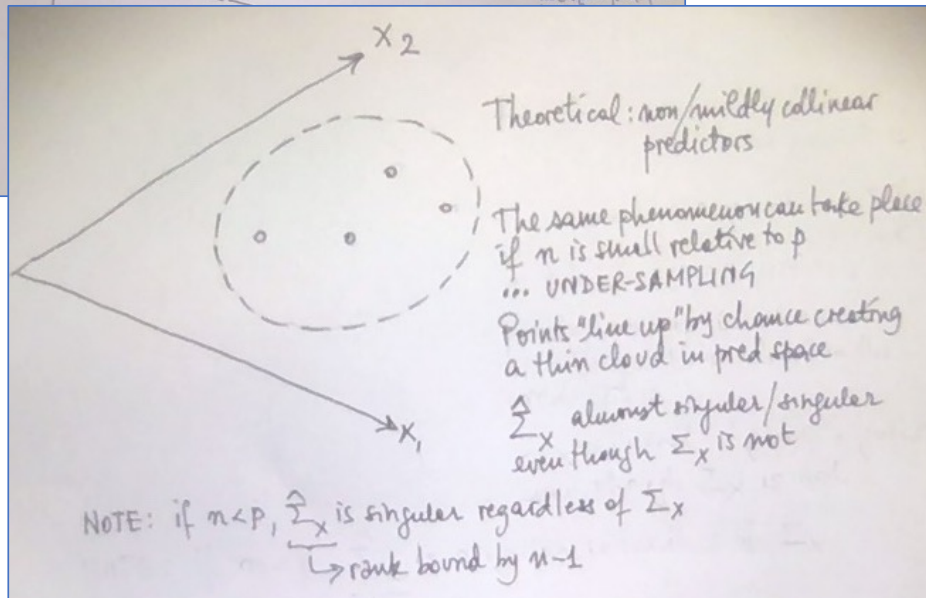
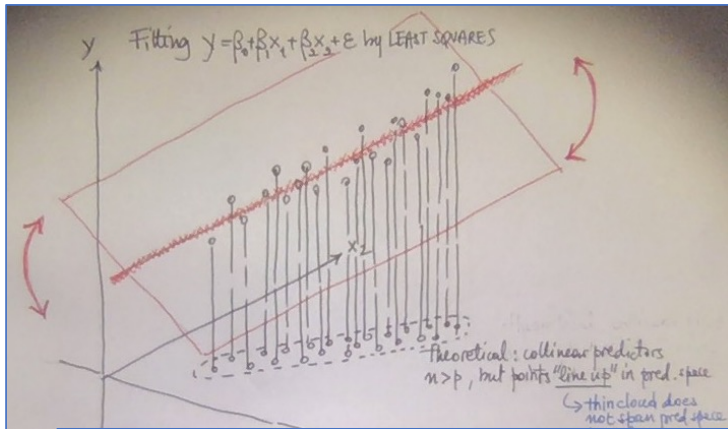
Ridge ('60-'70s): fitting procedure to *overcome multicollinearity* in regressions with highly interdependent features.

Loosely, in the LS, replaces $S_X = \underline{X}'\underline{X}$ (p x p sample covariance matrix of centered X's) with $S_X(\lambda) = \underline{X}'\underline{X} + \lambda I_p$, $\lambda > 0$ which is always invertible – spherically “increasing” the features’ spread in the data decreases the sampling variability of the estimator.

LASSO ('90s): fitting procedure to *produce sparse solutions* in regressions with a large number of features, only a subset of which is expected to matter.

Loosely, it performs “soft” features selection (more below), without specifying how many coefficients should be set to 0.

Both formulated as a constrained LS: *Size constraint on β* using different norms in R^p .



Constrained Least Squares (using different norms)

Ridge $\hat{\beta}_{Ridge} = \operatorname{argmin} \left\{ \| \underline{Y} - \underline{X}\beta \|^2 + \lambda \| \beta \|_{(2)} \right\}$

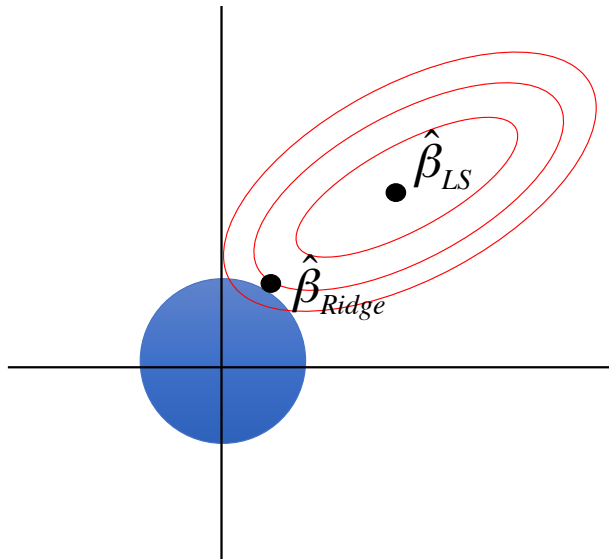
$$\| \beta \|_{(2)} = \sum_{j=1}^p \beta_j^2 \quad \text{L2 Norm}$$

LASSO $\hat{\beta}_{LASSO} = \operatorname{argmin} \left\{ \| \underline{Y} - \underline{X}\beta \|^2 + \lambda \| \beta \|_{(1)} \right\}$

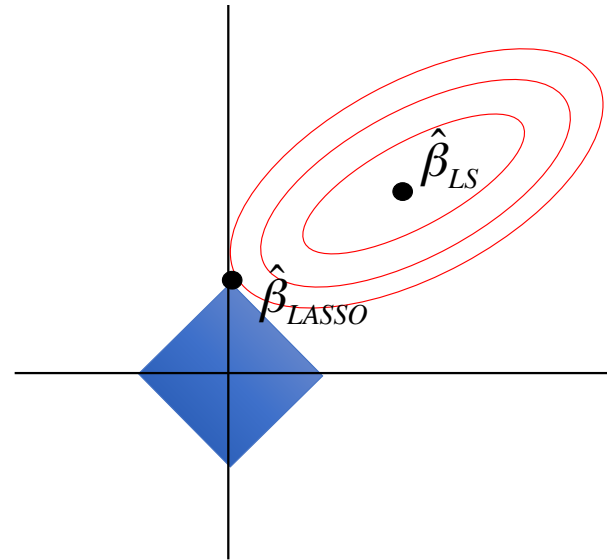
$$\| \beta \|_{(1)} = \sum_{j=1}^p |\beta_j| \quad \text{L1 Norm}$$

Note: while LS is scale equivariant, Ridge and LASSO are not – important to perform them after scaling features (e.g., divide each by its sd, so all will have the same unitary spread)

Cartoons with $p=2$



the diamond shape of the L1 constraint makes it more likely that the solution lies in a corner



$$\| \underline{Y} - \underline{X}\beta \|^2 = \min_{\beta \in \mathbb{R}^p}$$

$$\sum_{j=1}^p \beta_j^2 \leq s_\lambda$$

size constraint

$$\| \underline{Y} - \underline{X}\beta \|^2 = \min_{\beta \in \mathbb{R}^p}$$

$$\sum_{j=1}^p |\beta_j| \leq s_\lambda$$

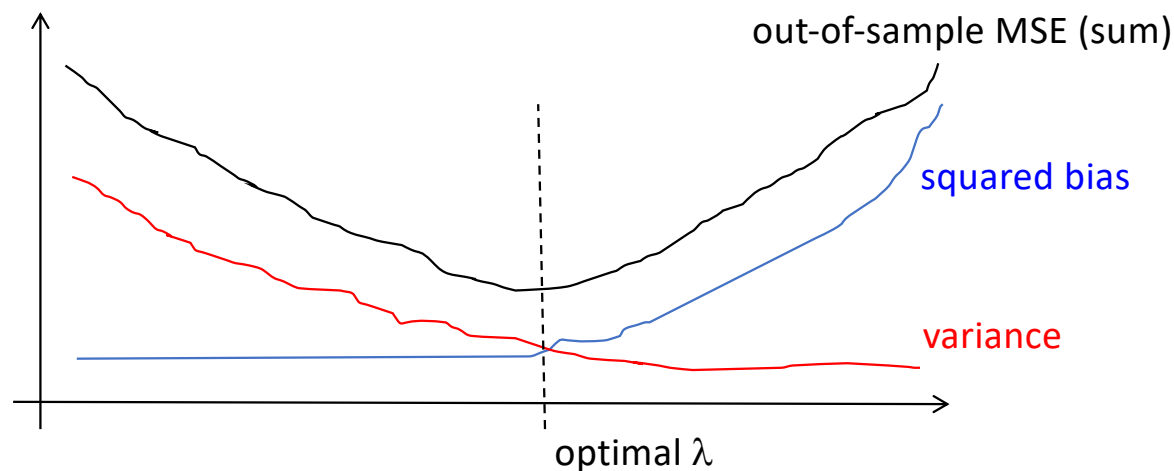
(maximal) VALUE OF THE SIZE CONSTRAINT (PENALTY PARAMETER) FIXED...

Extremely efficient algorithms exist to minimize the objective function with penalty and find the constrained solutions (convex optimization). LASSO is of course more computationally expensive than LS (with its close form solution), but not prohibitive also for very large p .

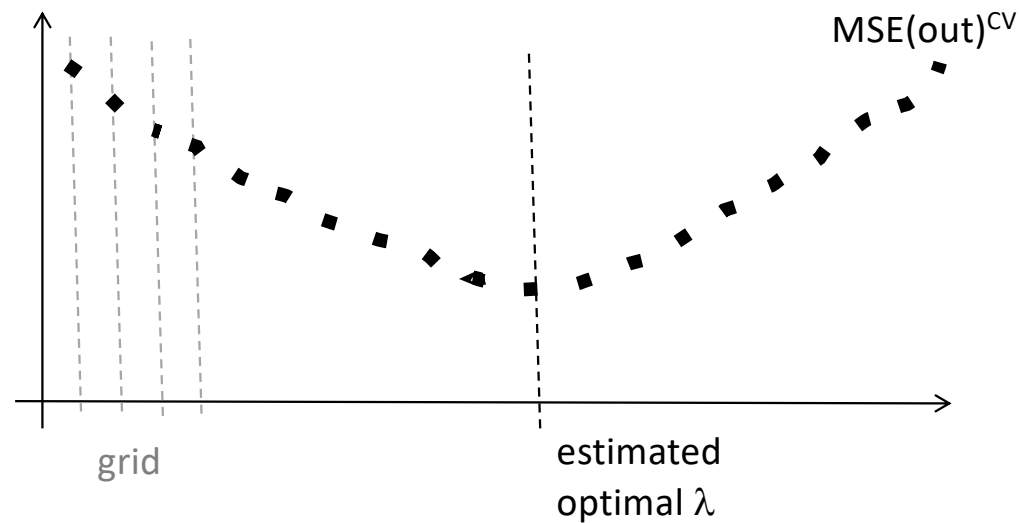
The critical part is *selecting an appropriate λ (s_λ)*.

Ridge and LASSO can improve over LS because they reduce the sample variability of the coefficient estimators and ameliorate overfitting. But they create (if LS is unbiased) or increase a bias component. As λ (and thus the weight of the penalty vs the RSS term in the objective function) grows, the variance decreases but the bias increases – in the limit for infinite λ all coefficient estimates will be 0, with no variance!

We want the “sweet spot” where the out-of-sample MSE is minimized:



In applications we don't know how the out-of-sample MSE varies as a function of λ . But we can estimate it on a grid of values using **Cross-Validation** (... coming next) :



Tuning the penalty (i.e. selecting an appropriate λ) by CV substantially adds to the computational burden; we need to iterate the optimization algorithm many times.

But it is essential for an effective use of Ridge and LASSO... and algorithms for running LASSO are very fast!

Note: we have introduced the geometry of the constraints taking λ as fixed, then discussed the selection of λ .

One could imagine selecting (e.g., by cross-validation) also some aspects of the geometry of the constraints? For instance, L1 or L2 norm? A combination of norms?

Elastic Nets – combine Ridge and LASSO penalization (not in ISLR).

Implementation for Linear (Gaussian) as well as Generalized Linear Models in the R package **GLMnet** (Ridge, LASSO, and Elastic Nets in between)

<https://cran.r-project.org/web/packages/glmnet/glmnet.pdf>

<https://cran.r-project.org/web/packages/glmnet/index.html>

COMPUTATIONAL ASSESSMENT AND TUNING OF STATISTICAL PROCEDURES

For a very long time, the properties of statistical procedures were assessed using mathematical manipulations, facts about probability distributions and asymptotic results.

Mathematical tractability defined a narrow scope for statistics:

- Consider only simple procedures
- Introduce strong assumptions on the stochastic mechanism generating the data, and/or
- Prove properties only for large samples

'70s onward: replacing math with computation has substantially expanded the scope.

IN PROCEDURES WHOSE PERFORMANCE DEPENDS CRUCIALLY ON TUNING PARAMETERS, E.G., RIDGE AND LASSO, HOW DO WE IDENTIFY THEIR OPTIMAL/SATISFACTORY VALUES... BASED ON THE DATA?

CROSS VALIDATION: A computational approach to evaluate the out-of-sample accuracy of a statistical procedure used for prediction (supervised problems). Computationally expensive but now viable.

Given a **model or procedure** comprising a **tuning parameter** (e.g., **polynomial regression** of **degree r** ; **lowess regression** with **smoothing parameter s** ; **classifier** with **threshold t** ; **LASSO regression** with **penalty λ**)... how do we choose the tuning parameter?

Mean Squared Error (MSE) for regressions: prediction of a continuous response – the quantity that is minimized in-sample by Least Squares fitting in the case of parametric LMs

$$MSE(in) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$\hat{y}_i = \begin{cases} \hat{\beta}_o + \hat{\beta}' x_i & \text{(parametric)} \\ \ell(x_i) & \text{(lowess fit)} \end{cases}$$

Misclassification rate (Err) for classifiers; prediction of a categorical response

$$Err(in) = \frac{1}{n} \sum_{k=0}^n Ind(\hat{y}_i \neq y_i)$$

$$\hat{y}_i = \text{classpred}(x_i) \quad (\text{classifier})$$

Residual Deviance for GLMs, from Maximum Likelihood estimation.

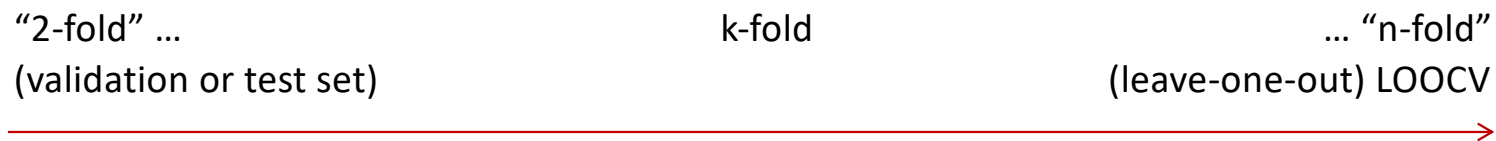
“in” = in-sample, same as “train” (training data for the procedure)

In-sample MSE (Err) can be a poor approximation of **out-of-sample MSE (Err)**; how closely we manage to “reproduce” the training data, especially with a large or complex model having lots of degrees of freedom, can say little about how accurately we would predict the response on independent test data.

In-sample MSE (Err) can substantially underestimate out-of-sample MSE (Err); if we overfit we are learning idiosyncratic components of the training data, not the underlying mechanism.

Traditionally, formulae were developed for specific problems (e.g., Mallow’s C_p for regression – good estimator of the out-of-sample MSE under assumptions and for large samples).

Cross Validation (CV) allows us to produce a reliable estimate of the out-of-sample MSE for a collection of values of the tuning parameter (a collection of possible models), and then select the one that provides the lowest.



- Divide the data at random in k subsets of equal size $S_1, S_2 \dots S_k$

- For $j=1, 2 \dots k$

Form Training and Test Sets
(sizes $n(k-1)/k$ and n/k , respectively)

$$TrSet = \bigcup_{h \neq j} S_h \quad TsSet = S_j$$

Train model/procedure on Training Set \hat{M}_j

Compute MSE on Test Set

$$MSE_j = \frac{1}{n/k} \sum_{i \in S_j} (y_i - \hat{M}_j(x_i))^2$$

- Average measurements over the folds

$$MSE(out)_k^{CV} = \frac{1}{k} \sum_{j=1}^k MSE_j$$

Note: with k values of MSE we can also produce a standard deviation...

Estimates the out-of-sample MSE (Err) without the *underestimation* one has in-sample, but with an *overestimation* due to using a smaller sample size in training!

Select k based on **computational burden**, as well as a **variance-bias trade off**.

“2-fold”
(validation or test set)

k-fold

“n-fold”
(leave-one-out) LOOCV



As k increases (from 2 to n)

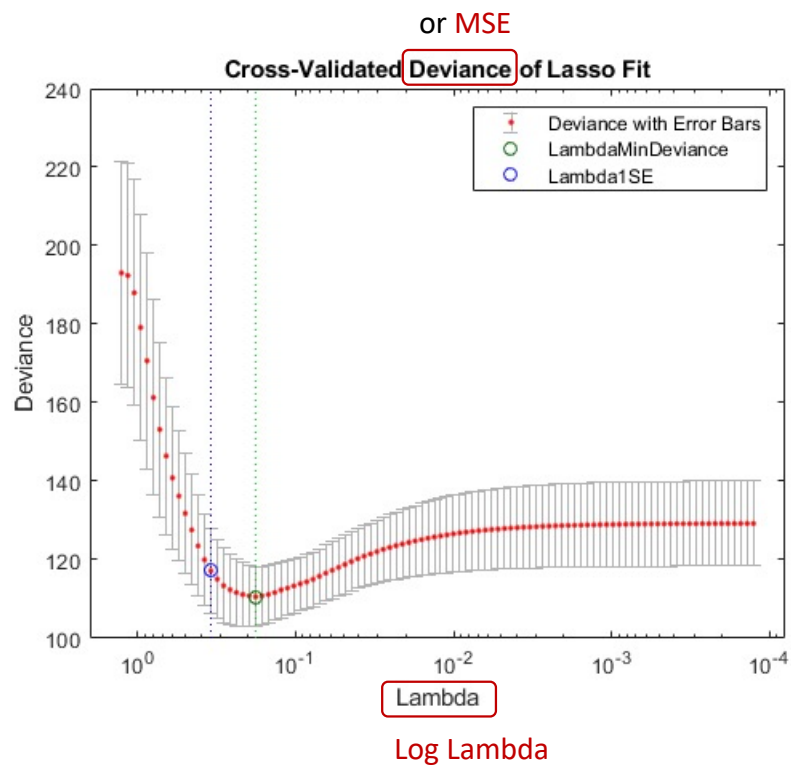
- Computational burden increases.
- Overestimation bias for $MSE(out)$ decreases: training sets used in each fold grow in size, up to $n-1 \sim n$.
- Variance for estimation of $MSE(out)$ increases: training sets used in each fold have larger overlaps; *while we average more numbers (k) they are more dependent* – less actual replication.

Common reasoning: using $k > 10$ often induces an increase in computational burden and variance that is not justified by a substantial decrease in the overestimation bias...

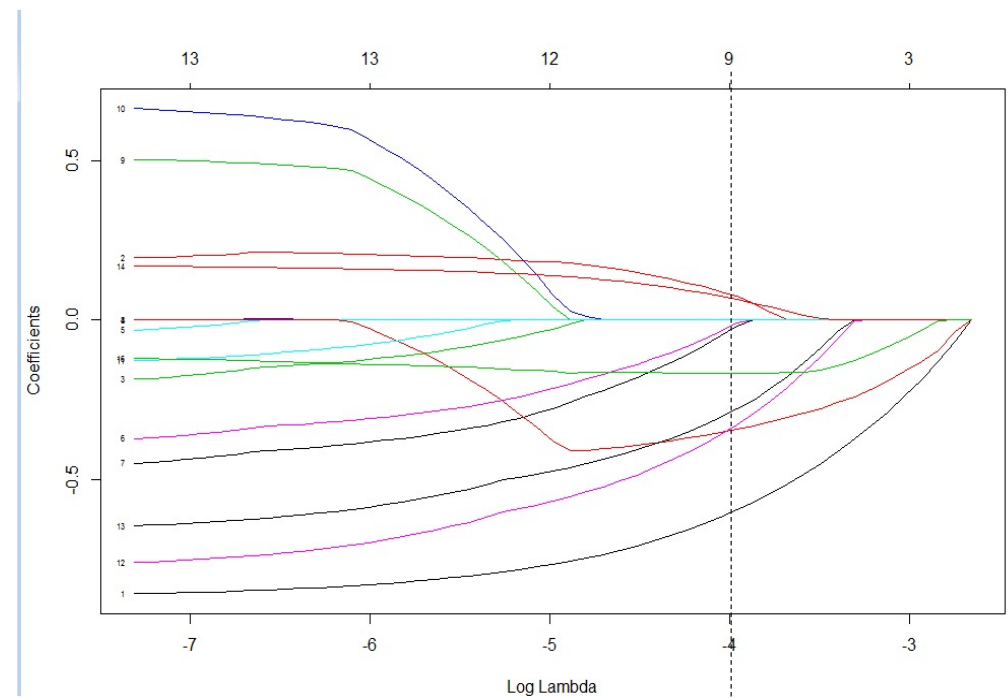
Note: in the old fashioned “validation (test) set approach”, the calculation is not even repeated flipping the roles of the two folds as training and testing sets.

BACK TO THE LASSO (for an LM or GLM)

Choosing the right penalty parameter

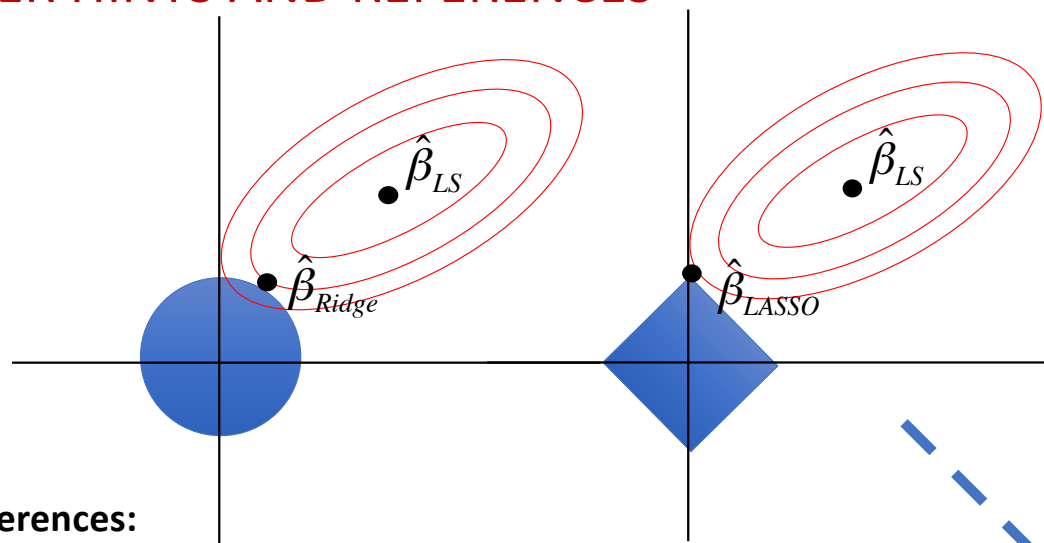


How coefficient estimates shrink (and go to 0) with penalization



at a given level of penalization, how many and which coefficients are non-0?

FURTHER HINTS AND REFERENCES



CONVEX constrained optimization

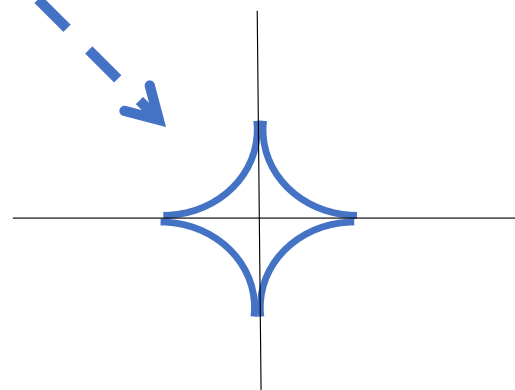
very efficient computational approaches.

Estimates are biased (even when the LS for the full model is not)... but **more stable**, and **sparse** (L1).

Very broad literature, lots of variants, including those to **incorporate group or order structure** for features.

Some references:

- Hastie T., Tibshirani R., Friedman J. (2009). Elements of Statistical learning 2nd ed. Springer.
- Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. JRSS B, 58(1) 267-288.
- Zou H. Hastie T. (2005) Regularization and variable selection via the elastic net. JRSS B, 67(2) 301-320.
- Tibshirani R., Saunders M. (2005) Sparsity and smoothness via the fused lasso. JRSS B, 67(1) 91-108
- Yuan M., Lin Y. (2006) Model selection and estimation in regression with grouped variables. JRSS B, 68(1) 49-67.
- Fan J. Li R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. JASA, 96(456) 1348-1360



... abandon CONVEXITY

reduced bias, but harder computational problem.

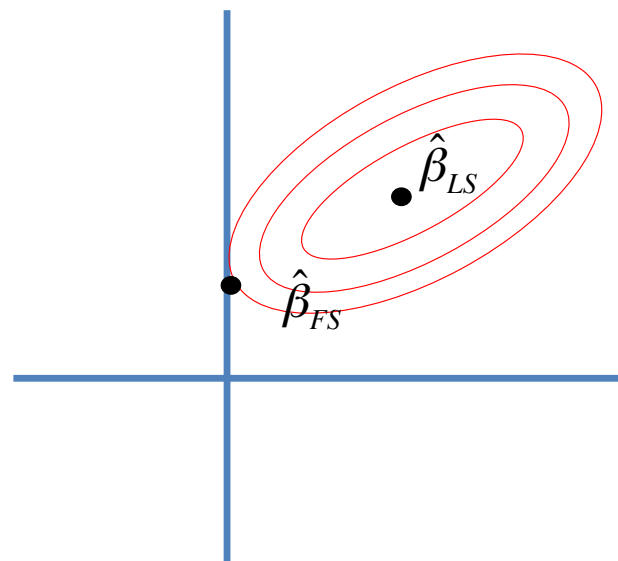
Traditional (“hard”) feature selection: the most **NON-CONVEX constrained optimization**

L0 norm counts non-0 coefficients, but size of each non-0 coefficient is unconstrained.

Much harder, previously not computationally viable; now *Mixed Integer Optimization*.

Some (non-traditional) references:

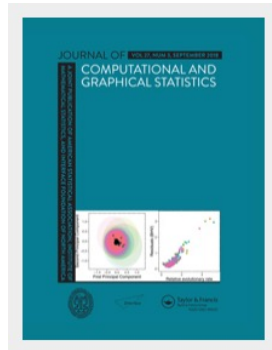
- Bertsimas D., King A., Mazumder R. (2016) Best subset selection via a modern optimization lens. AOS 44(2) 813-852.



$$\| \underline{Y} - \underline{X}\beta \|^2 = \min_{\beta \in \mathbb{R}^p}$$

$$\sum_{j=1}^p \text{Ind}(\beta_j \neq 0) \leq c$$

size constraint



Research Article


MIP-BOOST: Efficient and Effective L_0 Feature Selection for Linear Regression

Ana Kenney , Francesca Chiaromonte & Giovanni Felici

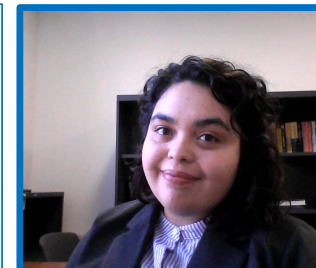
Received 19 Sep 2019, Accepted 14 Oct 2020, Accepted author version posted online: 17 Nov 2020,

Published online: 04 Jan 2021

 Download citation

 <https://doi.org/10.1080/10618600.2020.1845184>

 Check for updates



SIZE CONSTRAINT TUNED BY CROSS VALIDATION
... the traditional **Best Subset Selection** problem

Important: can also add further “integer” constraints to capture structure.

MIP-BOOST: Efficient and Effective L_0 Feature Selection for Linear Regression

Abstract: Recent advances in mathematical programming have made mixed integer optimization a competitive alternative to popular regularization methods for selecting features in regression problems. The approach exhibits unquestionable foundational appeal and versatility, but also poses important challenges. Here, we propose MIP-BOOST, a revision of standard mixed integer programming feature selection that reduces the computational burden of tuning the critical sparsity bound parameter and improves performance in the presence of feature collinearity and of signals that vary in nature and strength. The final outcome is a more efficient and effective L_0 feature selection method for applications of realistic size and complexity, grounded on rigorous cross-validation tuning and exact optimization of the associated mixed integer program. Computational viability and improved performance in realistic scenarios is achieved through **three independent but synergistic proposals**.

First: a **novel bisection procedure** designed specifically for tuning the sparsity bound, which significantly reduces the needed number of evaluations, cutting the computational burden of the MIP approach.

Second: a **novel cross-validation scheme** that exploits the structure of the MIP and of the simplex algorithm used for its solution to significantly reduce the computational effort of repeating calculations across folds (also warm starts and surrogate lower bounds).

Third: **whitening**, a pre-processing step that, by handling feature collinearities, can both reduce computational burden and improve solution quality. Whitening can be applied prior to any feature selection technique but it benefits MIP more than it does other approaches.