Afi Escuela de Finanzas

Regresión Avanzada

Máster en Data Science y Big Data en Finanzas

Javier Nogales – PhD Matemáticas

Catedrático, Estadística e IO, UC3M

fcojavier.nogales@uc3m.es 2022

Organization

- Subject organized in 4 topics
- 12 hours in total: 3 sessions
- Practical course: 50% basic concepts + 50% computer labs (using R)
- Evaluation: 100% final exercise (40% GLM + 60% this topic)

Objectives

- Relax some of the assumptions in classical linear regression (normality, loss functions, etc.)
- Deal with the curse of dimensionality in high-dimensional problems
- Handle the R language for advanced regression (including caret package)

Outline

1. General Loss Functions

Linear regression: a brief review

- Remember the usual framework in **Machine Learning / Statistics**:

$$\text{Data} = \text{Model} + \text{Noise}$$

- When we focus on one variable predicted by others:

$$y = g(x_1, \ldots, x_k) + \text{Noise}$$

- Most widely used tool (approximation) with p variables:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon$$

- This is the Multiple Linear Regression tool

- Note the variables in the true model, g, may be different in the approximation

Linear regression: a brief review

- For the model $y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$

- Assumptions

1. $E(\epsilon_i) = 0 \quad \forall i$
2. $\text{Var}(\epsilon_i) = \sigma^2 \quad \forall i$
3. $E(\epsilon_i \epsilon_j) = 0 \quad \forall i \neq j$
4. $\epsilon_i \sim \text{Normal} \quad \forall i$

1. $E(y_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ip} \quad \forall i$
2. $\text{Var}(y_i) = \sigma^2 \quad \forall i$
3. $\text{Cov}(y_i, y_j) = 0 \quad \forall i \neq j$
4. $y_i \sim \text{Normal} \quad \forall i$

- Note (3) and (4) imply the observations are independent

- We do not need (4) for estimation, but for inference

What happens if some assumptions do not hold?

- In practice, the output may not be continuous
- The errors may not be normal
- The relation between the output and the predictors may not be linear

Generalized Linear Models (GLM): try to extend linear models by relaxing previous assumptions in an unified way

Non-linear regression: specify a parametric non-linear (in parameters) relation and estimate the parameters using an optimization solver

Regression in high-dimension: OLS has high variance in high dimension, advanced tools try to reduce the variance while increasing the bias

Non-parametric regression / machine learning: a non-linear function is estimated (learned) that cannot be parametrized

Non-linear Regression

- Are these models linear?

$$
\begin{aligned}
y_i &= \beta_0 + \beta_1 x_{i1}^2 + \beta_2 x_{i2}^2 + \beta_3 x_{i1} x_{i2} + \epsilon_i \\
\log(y_i) &= \beta_0 + \beta_1 x_{i1} + \epsilon_i
\end{aligned}
$$

- Are these models linear?

$$
\begin{aligned}
y_i &= \beta_0 + \beta_1 x_{i1}^{\beta_2} + \epsilon_i \\
y_i &= \frac{\beta_0}{1 + \frac{x_{i1}}{\beta_1^2}} + \beta_2 x_{i2} + \epsilon_i
\end{aligned}
$$

Parametric Non-linear Regression

- Analyze non-linear relationships in the form:

$$y = F(\beta; x) + \text{Noise}$$

- We assume the parametric function F is known, except for the parameters β

- Estimation is done through nonlinear least-squares:

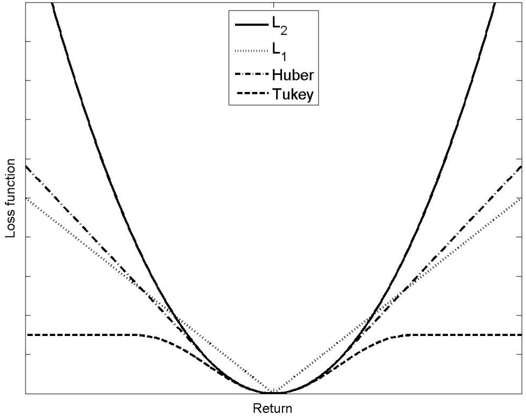
$$\text{minimize}_{\beta} \quad \frac{1}{2}\sum_{i}(y_i - F(\beta; x_i))^2$$

- Optimization algorithms are needed (gradients and Hessian matrix)

- Logistic regression can be viewed as a particular case, and also it is a particular case of GLM

General Loss Functions

- Residuals in OLS can be written as $\sum_i (y_i - x_i^T \beta)^2 = ||y - X\beta||_2^2$
- Hence, other versions if we change the norm (loss function)
- For instance, loss function in OLS is $\rho(x) = x^2$



Least-absolute or L_1 regression

- The loss function is $\rho(x) = |x|$
- It gives less weight to large errors (outliers):

$$\text{minimize}_\beta \quad ||y - X\beta||_1 = \sum_i |y_i - x_i^T \beta|$$

No explicit solution

- This approach is robust against outliers: the absolute value gives less weight to points far from the center than OLS

Chebyshev regression

- Minimax approach:

$$\text{minimize}_\beta \quad ||y - X\beta||_\infty = \min \max |y_i - x_i^T \beta|$$

  No explicit solution

- This approach is also robust but in a minimax sense: it minimizes the worst residual
- Very conservative approach

Robust regression

- A general robust framework:

$$\text{minimize}_\beta \quad \sum_i \rho\left(\frac{y_i - F_i(\beta; x_i)}{\sigma_i}\right),$$

where σ_i is a scale parameter (perhaps not included) and ρ is a loss (error) measure

Robust regression: particular cases

- Least squares: $\rho(x) = \frac{x^2}{2}$
- L_p estimators: $\rho(x) = |x|^p$, with $1 < p < 2$
- Lorentz estimator: $\rho(x) = \log(1 + \frac{1}{2}x^2)$
- Truncated least-squares estimator:

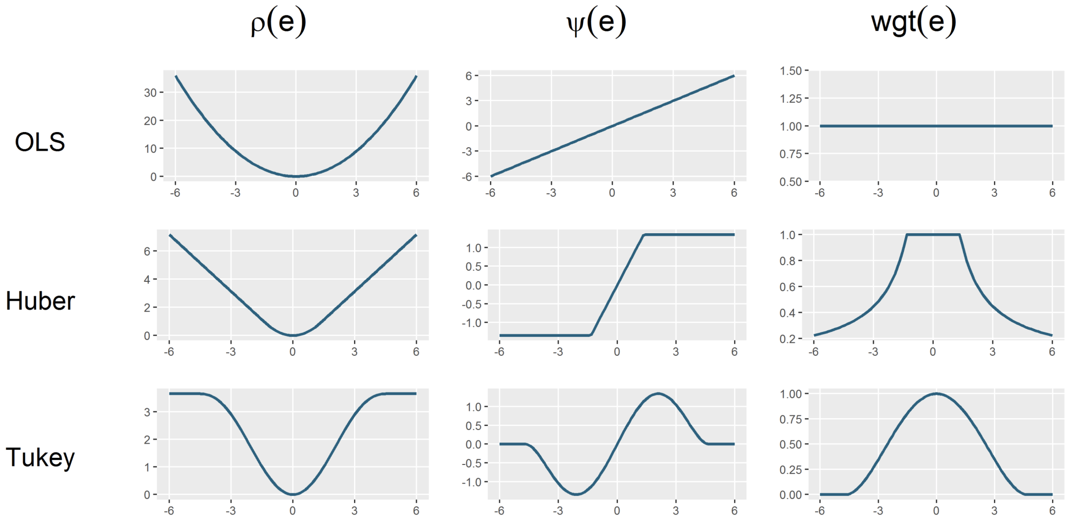
$$\rho(x,k) = \begin{cases} x^2, & |x| < \sqrt{k}, \\ k, & \text{otherwise} \end{cases}$$

- Huber function, etc.

$$\rho(x,k) = \begin{cases} \frac{1}{2}x^2, & |x| < k, \\ k|x| - \frac{1}{2}k^2, & \text{otherwise} \end{cases}$$

- Others

Robust regression: Loss, Influence, and Weight Functions



2. Regression Tools in High Dimension

Advanced Regression

How to improve simple linear models when the dimension is high?

- Try to improve the interpretability while attaining good predictive performance
- Need to replace least squares with some alternative fitting tools
- Need to balance prediction accuracy versus model interpretability (feature selection)

Redundant information, collinearity and overfitting

- To understand the impact of high dimension on estimates and predictions, we need to understand first the differences between redundant information, collinearity and overfitting

- Redundant information: no relevant/valuable information is added to explain or predict better

- We tend to think more data is better, but this is usually wrong:

  More data = more information + much more noise

- Redundant information implies collinearity and overfitting, but what's that?

Collinearity

- Phenomenon indicating the predictors are linearly related (redundant information in a linear way)
- Hence, it affects basically linear models
- In high dimension (p/n is large), collinearity appears almost sure
- That implies a close-to-singularity matrix $X'X$ (large condition number)
- That implies beta's are estimated with high noise
- That implies confusing and misleading results about the effects: non-reliable t-tests for parameters but reliable F-test, non-significant parameters in multiple regression but significant ones in simple regressions, etc.
- In linear models, collinearity implies overfitting, but what's that?

Overfitting

- Condition indicating the model is focusing on noise rather than information (more parameters than needed)

- In practice, overfitting causes very good predictions in past data, but bad predictions in new data

- But how harmful is collinearity and overfitting in linear models for prediction?

  Not harmful for prediction if new testing samples are within the range of training ones

  Otherwise, there is extrapolation risk, and no limits in prediction error

How harmful is overfitting in linear models?

- Estimation by OLS is not reliable (bad explanation):

$$E(\mathrm{Var}(\hat{\beta})) = \sigma^2 \ \mathrm{trace}((X^T X)^{-1})/p$$

- Prediction of output is reliable within the range of historical data (in-sample):

$$E(\mathrm{Var}(\hat{y})) = E(\mathrm{Var}(X\hat{\beta})) = \sigma^2 \ \mathrm{trace}(X(X^T X)^{-1} X^T) = \sigma^2 p/n$$

- But maybe not reliable if predictors are outside that range:

$$\mathrm{Var}(\hat{y}_0) = \mathrm{Var}(x_0' \hat{\beta}) = \sigma^2 \ (x_0'(X^T X)^{-1} x_0)$$

- Hence, how can we estimate with some accuracy β? To explain

- And, are all the p variables really needed to predict y? To predict better

To explain or to predict?

In regression/classification, there are three sources of uncertainty:

- The error in the coefficients when the linear approximation is true (estimation error)
- The error in the linear approximation when the true model is non-linear, or contains other variables (model bias)
- The noise in the DGP: Data $=$ Model $+$ Error (irreducible error)
- Error decomposition: Prediction Error = Data - $\widehat{\text{Model}}$

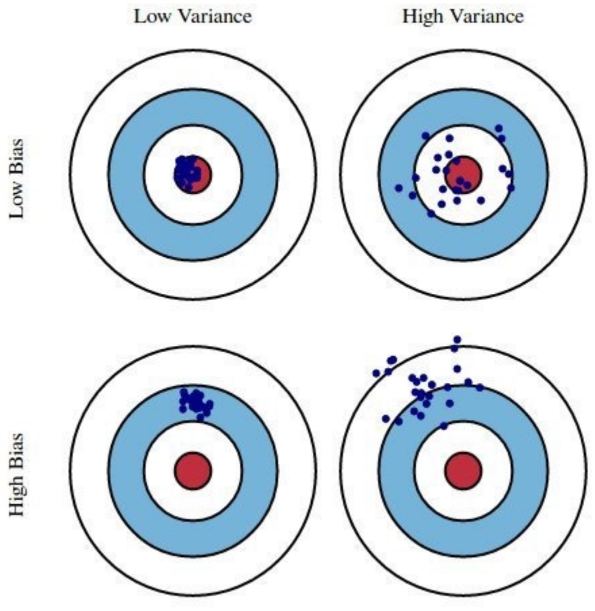
$$\boxed{(\text{Prediction Error})^2 = \sigma^2 + \text{Bias}^2 + \text{Var}}$$

To explain or to predict?

$$\boxed{(\text{Prediction Error})^2 = \sigma^2 + \text{Bias}^2 + \text{Var}}$$

- Statistics:
  - Focus on minimizing Bias (by assuming knowledge about population, DGP)
  - Hence, able to obtain formulas for Var that provides explanation (inference, effects of predictors on response)
  - The Var can be large in practice
- Machine Learning:
  - Focus on minimizing $\text{Bias}^2 + \text{Var}$
  - No assumptions needed (discover knowledge), hence no formulas and no explanation
  - But good prediction performance

Predictions: Bias vs Variance



Prediction error of linear models

- General decomposition: $(\text{Prediction Error})^2 = \sigma^2 + \text{Bias}^2 + \text{Var}$
- For a linear model $y = x^T\beta + \epsilon$ with p variables (features) and n observations
- Assume no bias: we are using the same variables as in the population
- Prediction error in training set:

$$(\text{Prediction Error})^2 = E((y-\hat{y})^2\,|x) = \sigma^2 + 0 + \sigma^2\frac{p}{n}$$

  It does not depend on collinearity, but increases with p

- Prediction error in testing set:

$$(\text{Prediction Error})^2 = \sigma^2 + 0 + \sigma^2(x_{\text{test}}^T(X_{\text{train}}^T X_{\text{train}})^{-1}x_{\text{test}})$$

  That implies reliable predictions within the range of historical data ($x_{\text{test}} \in X_{\text{train}}$)

Overfitting and underfitting in practice

- Consider the following true data generating process (DGP):

$$y = x_1\beta_1 + \cdots + x_p\beta_p + \epsilon = x'_{\text{true}}\beta + \epsilon$$

where $E(\epsilon) = 0$ and $\text{Var}(\epsilon) = \sigma^2$. The corresponding OLS estimator is denoted by $\hat{\beta}_{\text{true}}$

- **Overfitting**: the model is estimated with more variables than needed ($q > p$ variables):

$$y = x_1\beta_1 + \cdots + x_q\beta_q + \epsilon = x'_{\text{over}}\beta_{\text{over}} + \epsilon$$

If $\hat{\beta}_{\text{over}}$ denotes the OLS estimator, then the prediction is unbiased but with larger variance:

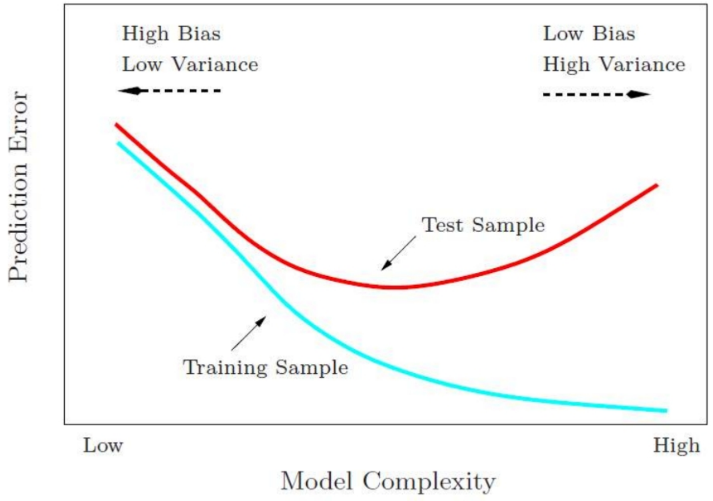
$$E(x'_{\text{over}}\hat{\beta}_{\text{over}}) = x'_{\text{true}}\beta_{\text{true}}, \quad \text{Var}(x'_{\text{over}}\hat{\beta}_{\text{over}}) \geq \text{Var}(x'_{\text{true}}\hat{\beta}_{\text{true}})$$

- **Underfitting**: the model is estimated with less variables than needed ($q < p$ variables). Then, the prediction is biased but with smaller variance, i.e.

$$E(x'_{\text{under}}\hat{\beta}_{\text{under}}) \neq x'_{\text{true}}\beta_{\text{true}}, \quad \text{Var}(x'_{\text{under}}\hat{\beta}_{\text{under}}) \leq \text{Var}(x'_{\text{true}}\hat{\beta}_{\text{true}})$$

Overfitting

When a model fits or predict very well the training data bat bad the testing data
(*p* is large compared with *n*)



Regression Tools in High Dimension

- Variable selection
- Regularization (shrinkage estimation)
- Dimension Reduction

- **Model Selection**

Model Selection

Assume we have a high-dimensional regression problem:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon = X\beta + \varepsilon$$

where p/n is large

Two main questions:

- How can we estimate with some accuracy β?
- Are all the p variables needed to predict y?

Model Selection

- Strong objective: identify the subset of the p variables that is really associated to the response
  This may be prohibitive: if $p = 50$, then more than 10^{15} models to try!

- Weak objective: identify a subset of the p variables that is relevant to the response
  But, what is the meaning of relevant?

- Some quality measures: adjusted R^2, Akaike information criterion (AIC), Bayesian information criterion (BIC), based on cross-validation, based on fitting in testing set,...

Model Selection: AIC

- For a given estimate $\hat{\beta}_d$ of the model $y = X_d\beta_d + \varepsilon_d$, where X_d represent a (column) subset of X, let RSS_d be the residual sum of squares:

$$\text{RSS}_d = \|y - X_d\hat{\beta}_d\|^2$$

and $\hat{\sigma}_d^2 = \text{var}(\hat{\varepsilon_d})$ is the residual variance

- We need to penalize RSS to avoid overfitting

- Hence, instead of focusing on RSS, we focus on

$$\text{AIC}_d = \frac{1}{n\hat{\sigma}_d^2}(\text{RSS}_d + 2\ d\ \hat{\sigma}_d^2)$$

- Choose model with smallest AIC_d

Model Selection: BIC

- The BIC has a heavier penalty, hence select a smaller number of variables

$$\mathrm{BIC}_d = \frac{1}{n}(\mathrm{RSS}_d + \log(n)\ d\ \hat{\sigma}_d^2)$$

- Choose model with smallest BIC_d

Model Selection: Cross-validation

- Leave-one-out cross-validation:
  - split the sample in two parts, one with one observation (test) and the other with the rest (train)
  - fit the model with the training set, validate the model with the test set
  - repeat the process and compute the validation error using test observations
- *k*-Fold Cross-Validation: similar, but randomly divide the sample in *k* groups, where the training set uses $k - 1$ folds and the test set uses the remaining one

  $k = 5$ or $k = 10$ is typically used
- Fewer assumptions than AIC or BIC. Indeed, AIC is asymptotically similar to leave-one-out CV
- But more computational demanding

Model Selection: testing set

- Similar to cross-validation:
  - **randomly** split the sample in two parts: one with (let's say) 80% observations (training set) and the other with the rest (testing set)
  - fit the model with the training set, validate the model with the testing set
  - repeat the process many times, computing each time fitting measures (like R^2) in the testing set
- Computational demanding

Model Selection: Subset Selection

Best subset selection:

- Fit a regression for each possible combination of the p predictors: 2^p combinations
- Fix a relevant criterion (AIC, BIC, cross-validation, . . .)
- Choose the combination with smallest criterion

Model Selection: Stepwise Selection

- Best subset selection is computationally infeasible for $p > 40$

- Forward Stepwise Selection:
  - Start with a model with no predictors, with the corresponding quality measure (AIC, BIC, cross-validation,...)
  - Add one predictor each time, and select that with best fit. Compute the corresponding criterion for the best fit (AIC, BIC, cross-validation,...)
  - Add one (remaining) predictor each time and repeat the process
  - Select the model with smallest criterion

- This is an heuristic: only $1 + p(p+1)/2$ models considered (quadratic instead of exponential)

Model Selection: Stepwise Selection

- Backward Stepwise Selection:
  - Start with the full model, with the corresponding quality measure (AIC, BIC, cross-validation,...)
  - Remove one predictor each time, and select that with best fit. Compute the corresponding criterion for the best fit (AIC, BIC, cross-validation,...
  - Remove one (remaining) predictor each time and repeat the process
  - Select the model with smallest criterion
- This is again an heuristic: only $1 + p(p+1)/2$ models considered
- Not feasible when $n < p$
- Hybrid approaches: variables are added and removed to the model sequentially (stepwise regression)
- Same ideas can be used to other regression models like logistic regression (where the deviance plays the role of RSS)

- **Regularization Methods**

Regularization Methods

- How can we estimate with some accuracy β?
- Main idea: penalize coefficient estimates, i.e. shrink them to 0
- With this framework, no need to select variables previously
- Main tools: Ridge, Lasso, Elastic Net, etc.

Ridge Regression

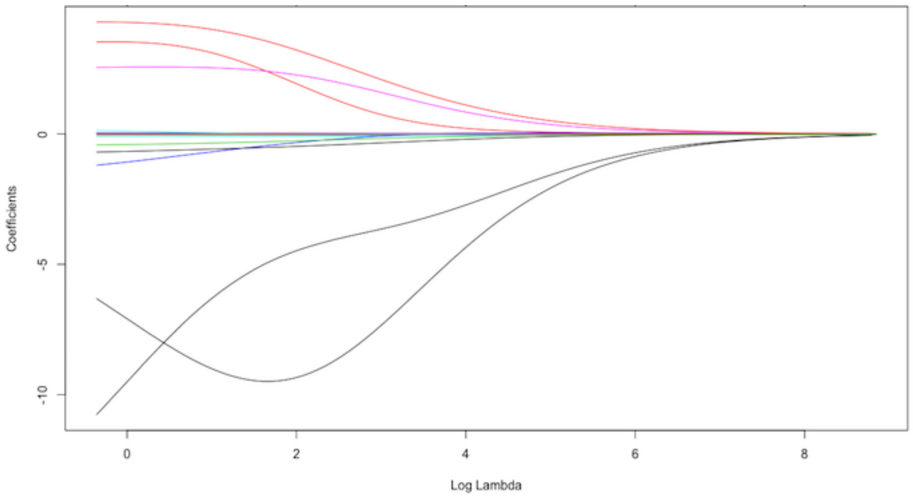
- Ridge regression: used in high dimension to mitigate overfitting (even if $p > n$)
- Also known as Tikhonov regularization: ill-conditioned problems

$$\text{minimize} \quad ||y - X\beta||_2^2 + \rho||\beta||_2^2$$

  where ρ is a tuning parameter, to be calibrated separately
- Explicit solution: $\hat{\beta} = (X^T X + \rho I)^{-1} X^T y$
- It adds some bias to the estimation to reduce a lot the variance: better MSE than OLS
- It is better the data matrix X is centered previously (no estimation of β_0, we do not want to shrink it). Then, $\hat{\beta}_0 = \bar{y}$
- It is also better to standardize the data, in order to make the estimation scale-invariant
- Low computational cost, good prediction accuracy, but dense solution (no variable selection)

Ridge Regression

- Ridge regression coefficients as $\log \rho$ increases



- Note ridge tends to OLS when ρ is small, and tends to 0 when ρ is large

Ridge Regression: some theoretical results

- It is a biased estimator:

$$\text{Bias}^2(\rho) = ||E(\hat{\beta}_{RR}(\rho)) - \beta^*||^2 = ||((X^TX + \rho I)^{-1}(X^TX) - I)\beta^*||^2 \neq 0$$

- It has better variance than OLS:

$$\text{Var}(\rho) = \sigma^2 \text{ trace } (X^TX + \rho I)^{-1}(X^TX)(X^TX + \rho I)^{-1} \leq \text{Var}(\hat{\beta}_{OLS}) = \sigma^2 \text{ trace } (X^TX)^{-1}$$

- MSE decomposition: $\text{MSE}(\hat{\beta}_{RR}(\rho)) = \text{Bias}^2(\rho) + \text{Var}(\rho)$

- Theorem (Theobald, 1974): there exists $\rho > 0$ such that

$$\text{MSE}(\hat{\beta}_{RR}(\rho)) < \text{MSE}(\hat{\beta}_{OLS})$$

Ridge Regression: a Bayesian view

- Up to now, β was fixed but unknown
- **Bayesian view:** treat it as a random quantity
- Before observing X and y, we assume a prior distribution for β, $p(\beta)$
- After observing X and y, we can compute the posterior distribution for β as

$$p(\beta|X, y) \propto p(\beta)\; p(X, y|\beta)$$

- If $p(\beta) \sim \mathcal{N}_p(0, 1/(2\rho^2)I)$ (parameters are close to 0), then the mean of the posterior distribution is equivalent to ridge regression

The Lasso

- Lasso regression: used in high dimension to mitigate overfitting (even if $p > n$)
- L_1 regularization: sparse solutions

$$\text{minimize}_\beta \quad \frac{1}{2}||y - X\beta||_2^2 + \rho||\beta||_1$$

- No explicit solution: non-differentiable problem
- It adds some bias to the estimation to reduce a lot the variance: better MSE than OLS
- Attains sparsity (model selection at the same time)
- Again, it is convenient the data matrix X is centered previously (no estimation of β_0, we do not want to shrink it). Then, $\hat{\beta}_0 = \bar{y}$
- State-of-the-art tool in Big Data Analytics

The Lasso

- Many ways to estimate the lasso regression: solving a quadratic optimization problem, solving the original (but non-differentiable) problem, using the LARS, using coordinate descent, . . .

- With non-prior information, ridge regression attains less variance than lasso (with similar bias)

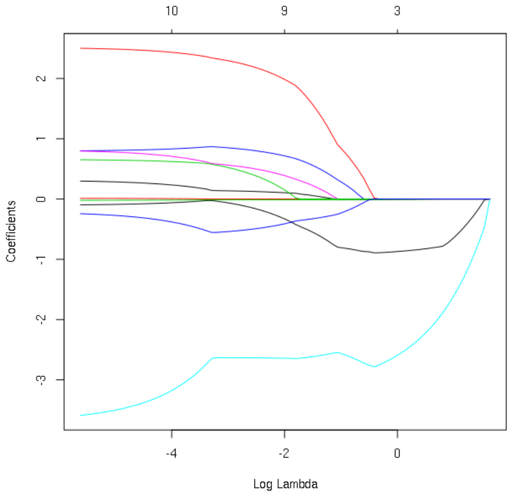
- If real model is sparse, then lasso performs better

- Take care: if $p > n$, the Lasso selects at most n variables (which can be small)

- **Bayesian view:** if the prior for β is a Laplace distribution (some parameters are 0), then the posterior mean is equivalent to the Lasso

The Lasso

- Lasso regression coefficients as $\log \rho$ increases



- Note lasso tends to OLS when ρ is small, and tends to 0 when ρ is large, but in a sparse way

Cardinality constraints or L_0 regression

- Based on 0-norm penalty:

$$\text{minimize}_\beta \quad \frac{1}{2}||y - X\beta||_2^2 + \rho||\beta||_0$$

  where $||\beta||_0$ is the number of non-zero elements in β

- $||\cdot||_0$ is not really a norm...
- This formulation is equivalent to best subset selection
- Can improve computational efficiency by using good MIP techniques (optimization)
- Not quite stable in practice

Regularization: Elastic Net

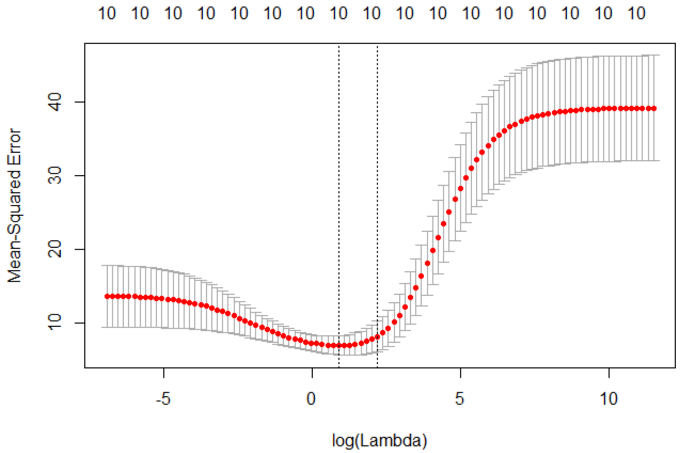
- Based on 1 and 2-norm penalties:

$$\text{minimize}_{\beta} \quad \frac{1}{2}||y - X\beta||_2^2 + \alpha\rho\,||\beta||_1 + \alpha(1-\rho)\,||\beta||_2^2$$

- The 1-norm controls sparsity
- The 2-norm stabilizes the regularization path
- But we need to calibrate two parameters…
- Recommended packages for regularization tools: glmnet (in R), scikit.learn (in Python)

Regularization: Tuning

- How to optimize the hyper-parameters ρ?
- Perform cross-validation and select the parameter by minimizing sum of squared residuals



Regression in high dimension: some conclusions

- Model selection tools are also dimension-reduction tools, but without changing the input. Usually they have smaller bias but larger variance
- Model selection focuses more on interpretation, less on prediction
- Regularization tools do not change the input, but change (bias) the estimator. Usually they have larger bias but smaller variance
- Lasso performs a bit worse than Ridge (in terms of prediction error) but provides model selection (better interpretation)

- **Dimension Reduction**

Dimension Reduction Tools

- Principal Component Regression (PCR): reduce dimension in X using just a few PCs, and then perform the regression
- Partial Least Squares (PLS): similar in spirit to PCR but using (besides X) the information in y to reduce the dimension

  Widely used in chemometrics, especially when $p >> n$
- PLS, PCR, and Ridge regressions perform similarly

  Ridge regression maybe preferred because simplicity

Principal Component Regression

- **Principal component regression**: instead of fitting a high-dimensional regression model:

$$y = X\beta + \varepsilon, \quad \text{where } \beta \in \mathcal{R}^p$$

  apply PCA: $Z = XA_r$ with $r << p$ and fit

$$y = Z\beta + \varepsilon, \quad \text{where now } \beta \in \mathcal{R}^r$$

- Because matrix Z is orthogonal, PCR is a sum of univariate regressions

- We need to standardize previously X and y

Partial Least Squares

- **Partial Least Squares**: similar in spirit to PCR
- PLS also looks for linear combinations of the predictors, but using y in addition to X to find the optimal combination
- In other words, PCR identifies directions in an unsupervised way whereas PLS uses the response to supervise the identification of the principal components
- I.e. PLS tries to find directions that help explain both the response and the predictors
- In general, PLS has better prediction performance than PCR, but not always
- We need to standardize previously X and y

Partial Least Squares: Procedure

- After standardizing the p predictors, the weights for the first linear combination are computed through the slopes of the simple regressions between y and X_j (instead of using the eigenvectors, as in PCA)
- That implies the weights of the linear combination are proportional to the correlation between y and X_j
- Hence, PLS gives the highest weight to the most correlated variable with the response
- Subsequent directions are found by taking residuals and then repeating the above step

Dimension-reduction tools: some conclusions

- PCR and PLS are are dimension-reduction tools that change the input, hence they are less interpretable

- PCR and PLS perform roughly similar to Ridge regression, so the last one is usually preferable (much simpler and smoother)

Optimization of hyper-parameters

Selecting the tuning parameter:

- How can we select the number of variables in step-wise regression, or the penalty parameters in regularization tools, or the number of directions in PCR or PLS?

- Cross-validation: choose a grid of those values, and compute the cross-validation error (quality measure) for each value of the grid

- Select the value with smallest cross-validation error

This is the usual way, but computationally expensive

- **Feature Selection**

Feature Selection

- Remember the usual framework in **Machine Learning / Statistics**:

$$\text{Data} = \text{Model} + \text{Noise}$$

- Supervised Learning with real variables/features causing the target:

$$y = g(x_1, \ldots, x_k) + \text{Noise}$$

- In Statistics or Machine Learning a set of p variables is used instead: $x_1, \ldots, x_p$
- Feature Selection: out of these p features, how to select the most promising ones
- Ideally, we should select the k real ones, but impossible in practice

Feature Selection

We can use previous knowledge:

- To perform p simple regressions and select the features with the highest correlation (or beta) with the output
- To perform one multiple regression over all the p variables and select the significant ones
- To consider the same multiple regression and focus on the highest t-values (variable importance)
- To apply stepwise regression
- To apply Lasso or Elastic Net

Feature Selection

We can use new knowledge:

- Recursive Feature Elimination: based on backward selection and variable importance (using LM or RF, etc.)
- Random Forest: variable importance
- Others: foci, boruta, etc.

3. Logistic Regression

From Regression to Classification

Logistic Regression

- Extension of classical multiple regression where the response was a continuous variable
- Now the response is a qualitative variable, usually binary, or the associated proportions
- Hence, logistic regression can be viewed as a particular case of
  - Non-linear regression
  - GLM (logit link)
  - Classification (linear classifier)
- The predictors can be either continuous or categorical
- Three types:
  - Binary: only two possible outcomes
  - Multinomial: more than two outcomes, no ordering
  - Ordinal: more than two outcomes, with a natural ordering

Logistic Regression

- Process to predict a categorical/qualitative variable as a function of explanatory variables (predictors)
- Related with Statistical Classification: first predict the probability of each category, then predict the category
- Questions answered by Logistic Regression:
  - Which predictors are associated with the response?
  - What is the relationship between the response and the predictors?
  - Is a linear approximation valid?
  - How accurate is the prediction?
- Based on conditional distribution of the response given the predictors

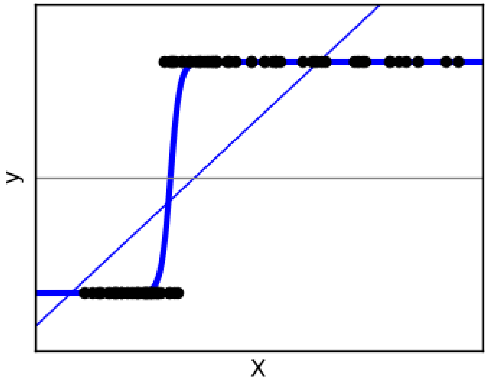
Applications

- Credit scoring, credit risk
- Stock trading
- Electronic fraud detection
- Spam filtering
- DNA sequence data to detect disease causing
- Probability of a heart disease in the next five years

Binary Logistic Regression

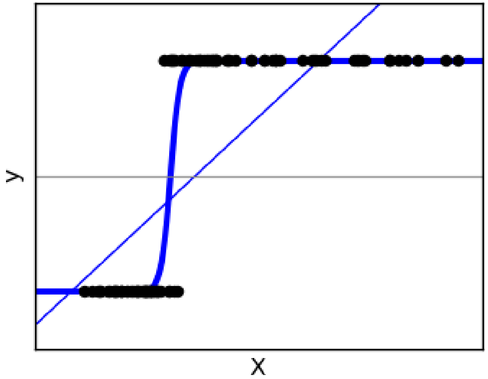
- Imagine we have just two groups: binary regression.
  Then, the possible y labels are 0 or 1 (first and second group, respectively).

- Instead of modeling y as a continuous variable and then $E(y|x_1, \dots, x_p) = \beta_0 + \beta^T x$



Binary Logistic Regression

- Better to consider y is binary and then model
$$E(y|x_1, \ldots, x_p) = p = P(y = 1|X = x) = F(\beta_0 + \beta^T x)$$
where F is a continuous function between 0 and 1



Binary Logistic Regression

- The quantity $\frac{p}{1-p}$ is called the odds
- An odds ratio of 1 indicates there is no association between the response and predictor
- A small value indicates very low probability that $y = 1$, whereas a large value indicates very high probability
- Usually, we impose that the log-odds or logit(p) is linear:

$$\text{logit}(p) = \log \frac{p}{1-p} = \beta_0 + \beta^T x$$

- In this case, a negative value indicates most likely $y = 0$, whereas a positive value indicates most likely $y = 1$ (more symmetry)
- If we increase an x-variable by one unit, the logit is increased by the corresponding β

Binary Logistic Regression

- Hence, $p = F(\beta_0 + \beta^T x) = \frac{\exp(\beta_0+\beta^T x)}{1+\exp(\beta_0+\beta^T x)} = \frac{1}{1+\exp(-\beta_0-\beta^T x)}$
- Note $P(y = 0|X = x) = 1 - p = \frac{1}{1+\exp(\beta_0+\beta^T x)}$
- Need optimization to estimate the β's (based on the MLE of y given x)
- Non-linear LS could be used, but MLE has better statistical properties (GLM)

Binary Logistic Regression

- MLE: $\max \prod_{i=1}^{n} p_i(x_i|\beta)^{y_i}(1 - p_i(x_i|\beta))^{1-y_i}$
- The multivariate distribution of X is not so important: explanatory variables can be non-Gaussian, contain categorical variables, etc.
- What is important is that the logit be linear
- Taking logs: $\max \sum_{i=1}^{n}(\beta_0 + \beta^T x_i)y_i - \sum_{i=1}^{n} \log(1 + \exp(\beta_0 + \beta^T x_i))$
- Use a convex optimization solver, problem is concave and differentiable
- After the estimation, for a given observation x, we can predict the associated probability:

$$\hat{p}(x) = P(y = 1) = \frac{\exp(\hat{\beta}_0 + \hat{\beta}^T x)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}^T x)} = \frac{1}{1 + \exp(-\hat{\beta}_0 - \hat{\beta}^T x)}$$

Binary Logistic Regression

- Note logistic regression is also a linear classifier: the decision boundary separating the two predicted classes is the solution of $\hat{\beta}_0 + \hat{\beta}^T x = 0$

- That means, we are minimizing the mis-classification rate: we say $y = 1$ when $p \geq 0.5$ and $y = 0$ otherwise. I.e. we say $y = 1$ when $\hat{\beta}_0 + \hat{\beta}^T x \geq 0$

- Moreover, the distance from a observation to the decision boundary is

$$\frac{\hat{\beta}_0 + \hat{\beta}^T x}{||\hat{\beta}||}$$

- Hence, class probabilities go towards the extremes (0 and 1) more rapidly when $\hat{\beta}$ is larger

- If mis-classification errors are asymmetric, instead of classifying $y = 1$ with rule $\hat{p} > 0.5$, change the 0.5 to be more conservative or risky

Multinomial Logistic Regression

- General case: G groups. No order
- Examples: departments at a business (e.g., marketing, sales, HR), search engine used (e.g., Google, Yahoo!, MSN), and color (black, red, blue, orange), etc.
- Hence, labels $y = \{0, 1, \ldots, G-1\}$
- First group is the control or reference group, $y = 0$
- For the first group:

$$p_0 = P(y = 0|X = x) = \frac{1}{1 + \exp(\beta_{0,1} + \beta_1^T x) + \cdots + \exp(\beta_{0,G-1} + \beta_{G-1}^T x)}$$

Multinomial Logistic Regression

- For the rest of the groups:

$$p_g = P(y = g|X = x) =$$

$$\frac{\exp(\beta_{0,g} + \beta_g^T x)}{1 + \exp(\beta_{0,1} + \beta_1^T x) + \cdots + \exp(\beta_{0,G-1} + \beta_{G-1}^T x)}, \; g = 1, \ldots, G - 1$$

- We need to impose $p_0 + \ldots + p_{G-1} = 1$
- Estimation is performed through optimization of the MLE
- Note in this case there are $G - 1$ linear classifiers

Multinomial Logistic Regression

- Take care: if we include explanatory variables that are not significant, then we may introduce a large bias in the estimation
- Hence, make a good variable selection before estimation, or estimate through shrinkage or regularization: add a penalty term of the form $\rho||\beta||$ in MLE
- The performance of the logistic regression model for $G \geq 3$ is not so good. Better performance attained by other classification tools. But for $G = 2$ the performance is good, as it allows for general predictors

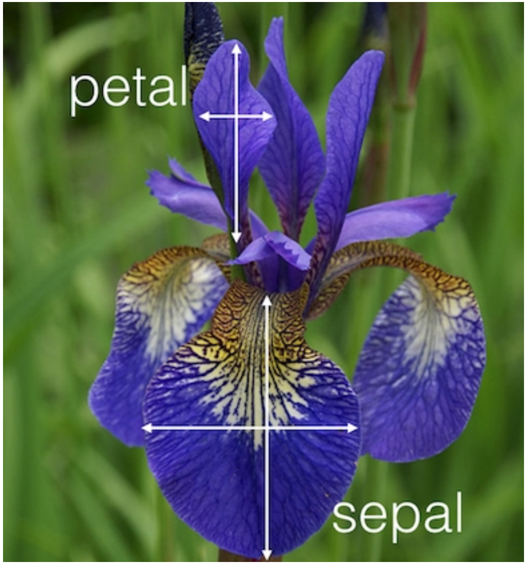
Ordinal Logistic Regression

- Again there are G groups, but now the order matters
- Examples: effectiveness on a scale of 1-5, levels of flavors for hot wings, medical condition (e.g., good, stable, serious, critical), etc.
- Now we have
$$\sum_{g=1}^{G^*} p_g = \frac{\exp(\beta_{0,g} + \beta^T x)}{1 + \exp(\beta_{0,g} + \beta^T x))},$$
where $G^* \leq G$, and $p_1 \leq p_2 \leq \cdots \leq p_G$
- Note this model is a cumulative sum of probabilities which involves just changing β_0
- Estimation is performed through optimization of the MLE

Logistic Regression: Classical Example

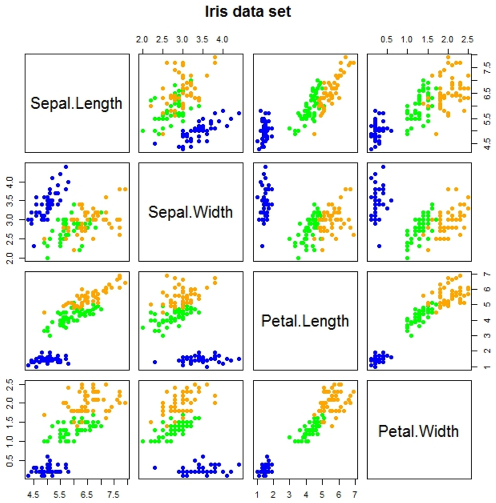
- Fisher’s Iris data set: best known data set in classification
- Data collected by Edgar Anderson in the 30’s to quantify the morphologic variation of Iris flowers of three related species
- It consists of 150 Iris flowers of three different species: setosa, versicolor, virginica
- Four types of measurements for each flower, the length and width of sepals and petals in centimeters, respectively
- Popularized by Ronald Fisher in 1936
- Based on the combination of the four features, Fisher developed a linear discriminant model to distinguish the species from each other

Iris Flower



Iris Scatter Plot

In blue, setosa Iris; in green, versicolor Iris; in orange, virginica Iris



Iris example

- First linear classifier (versicolor respect to setosa):

$$8.1 + (-7 \quad -7.2 \quad 14 \quad 18)x = 0$$

- Second linear classifier (virginica respect to setosa):

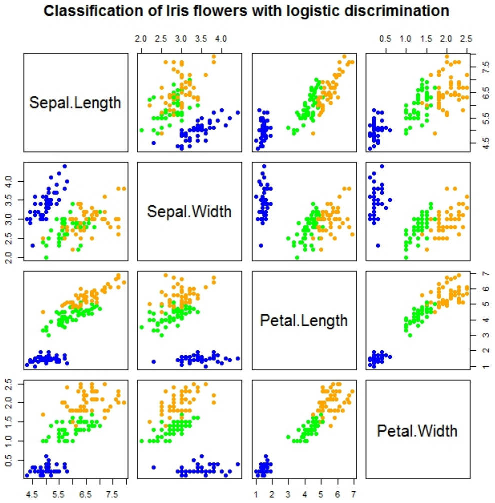
$$-34.5 + (-9.5 \quad -13.9 \quad 23.4 \quad 36.3)x = 0$$

- Posterior probabilities for 120-th flower:
  $p_{\text{setosa}} = 0$, $p_{\text{versicolor}} = 0.08$, $p_{\text{virginica}} = 0.92$

- Hence, predict observation 120-th is virginica

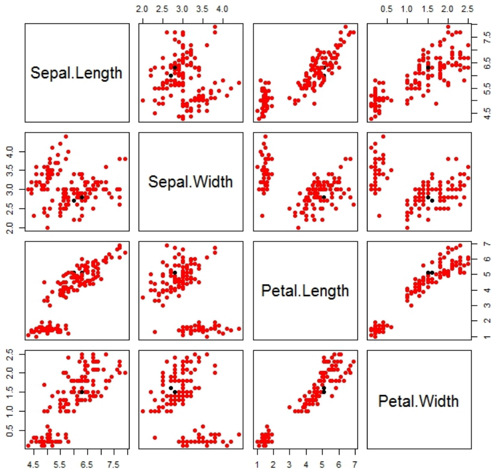
- But in general, how to select the threshold probability to classify?

Iris: Logistic regression



Iris: Logistic regression

Bad (in black) classifications for Iris flowers with logistic discrimination



Classification: performance measures

- For a given threshold probability, we have the following table (confusion matrix):

	Classify in P_1	$\cdots$	Classify in P_G
Belongs to P_1	n_{11}	$\cdots$	n_{1G}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
Belongs to P_G	n_{G1}	$\cdots$	n_{GG}

  where n_{ij} is the amount of observations coming from partition i that are classified in partition j

- The **error** is then: $\frac{n_{12}+\cdots+n_{G,G-1}}{n}$, the sum of off-diagonal elements

- The **accuracy** is the sum of the diagonal terms

- The confusion matrix should be computed using cross-validation or testing sets (not training ones)

Performance measures

- The in-sample or training error may be too optimistic, due to overfitting
- We need more realistic errors:
- Use cross-validation: exclude from the sample one observation, estimate parameters to build the classification rule, classify the excluded observation, and finally check the error. Hence, n classifications are performed
- Use a training and a validation set: let's say, 80% of the data to estimate (train or learn) the model and the other 20% to validate. Repeat many times
- These are known as out-of-sample or testing errors
- Finally, note from confusion matrix other performance measures can be obtained: Accuracy, Kappa, Sensitivity (TPs), Specificity (TNs), etc.

Iris example: logistic regression performance

- In sample performance:

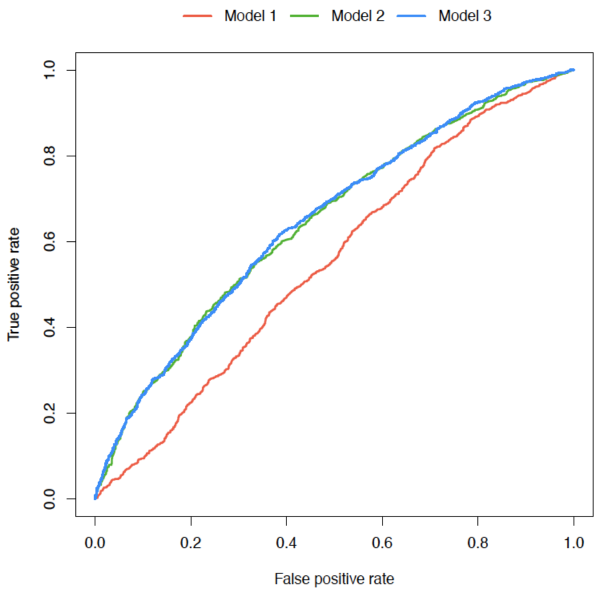
```
predictions  setosa versicolor virginica
setosa           50          0         0
versicolor        0         49         1
virginica         0          1        49
```

- In-sample error = 1.3% (small, but optimistic)
- Cross-validation: error = 2% (a bit worse, but more realistic)
- Out-of-sample or testing: error = 3% (a bit worse, but more realistic)

ROC curves

- Used for binary problems by varying the cutoff to assign groups (Bayes' rule)
- Compare false positive rates, $P(\hat{y} = 1 | y = 0)$, with true positive ones, $P(\hat{y} = 1 | y = 1)$
- The true positive rate is also called the sensitivity and 1 minus the false positive rate is also called the specificity
- The receiver operating characteristic (ROC) curve is created by varying the cutoff from 0 to 1

ROC curves


Model 1
Model 2
Model 3
True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

AUC

- In previous graph, models 2 and 3 have higher true positive rates than model 1
- But difficult to compare models 2 and 3
- A useful summary of the overall quality is the area under the curve (AUC)
- An AUC of 0.5 indicates a random guessing while an AUC of 1 indicates perfect classification

The end… Attained objectives

- Relax some of the assumptions in classical linear regression (normality, loss functions, etc.)
- Deal with the curse of dimensionality in high-dimensional problems
- Handle the R language for advanced regression (including caret package)

“Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise.”

John W. Tukey

Afi Escuela de Finanzas