Machine Learning for biology

V. Monbet

UFR de Mathématiques
Université de Rennes 1

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Linear model

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$$

Predictive ability

- The quality of a linear model is evaluated by Mean Square Error (MSE).

$$\text{MSE} = \text{bias}^2 + \text{variance}$$

- Linear regression has low bias (zero bias) but suffers from high variance. So it may be worth sacrificing some bias to achieve a lower variance.

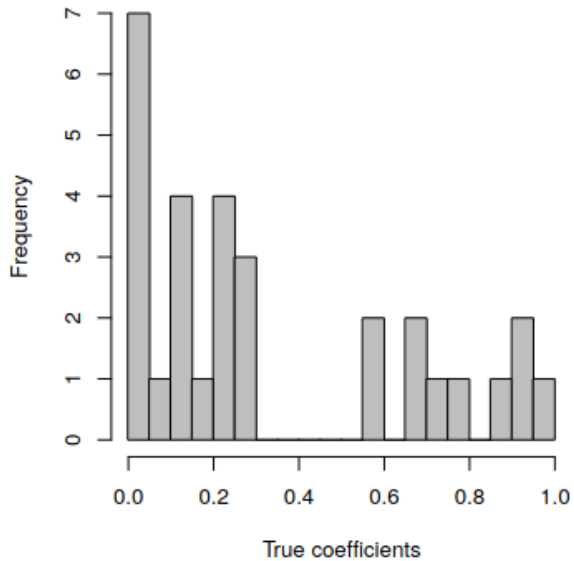
Interpretability

- with a large number of predictors, it can be helpful to identify a smaller subset of important variables. Linear regression doesn't do this.

Also, linear regression is not defined when $p > n$.

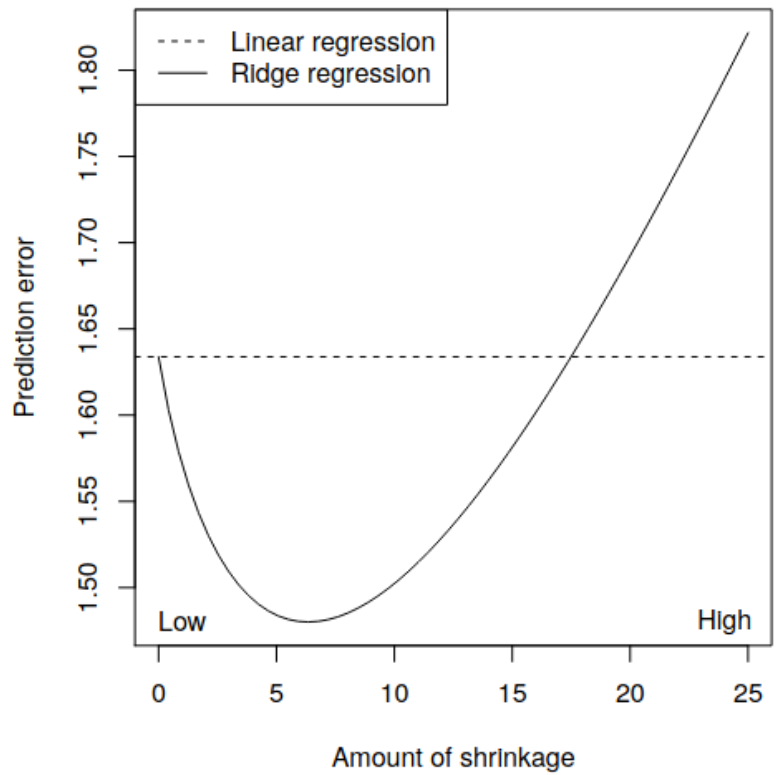
Linear model

Example : we have $n = 50$, $p = 30$, and $\sigma^2 = 1$. The true model is linear with 10 large coefficients (between 0.5 and 1) and 20 small ones (between 0 and 0.3). Histogram:



The linear regression fit:
Squared bias $\simeq 0.006$
Variance $\simeq 0.627$
Pred. error $\simeq 1 + 0.006 + 0.627 \simeq 1.633$

We reasoned that we can do better by shrinking the coefficients, to reduce variance.



Linear regression:
Squared bias $\simeq 0.006$
Variance $\simeq 0.627$
Pred. error $\simeq 1 + 0.006 + 0.627$
Pred. error$\simeq 1.633$

Ridge regression, at its best:
Squared bias $\simeq 0.077$
Variance $\simeq 0.403$
Pred. error $\simeq 1 + 0.077 + 0.403$
Pred. error $\simeq 1.48$

Overfitting

- Overfitting occurs when dimension is high and/or when inputs are correlated.
- It leads to unstable models with poor prediction properties.

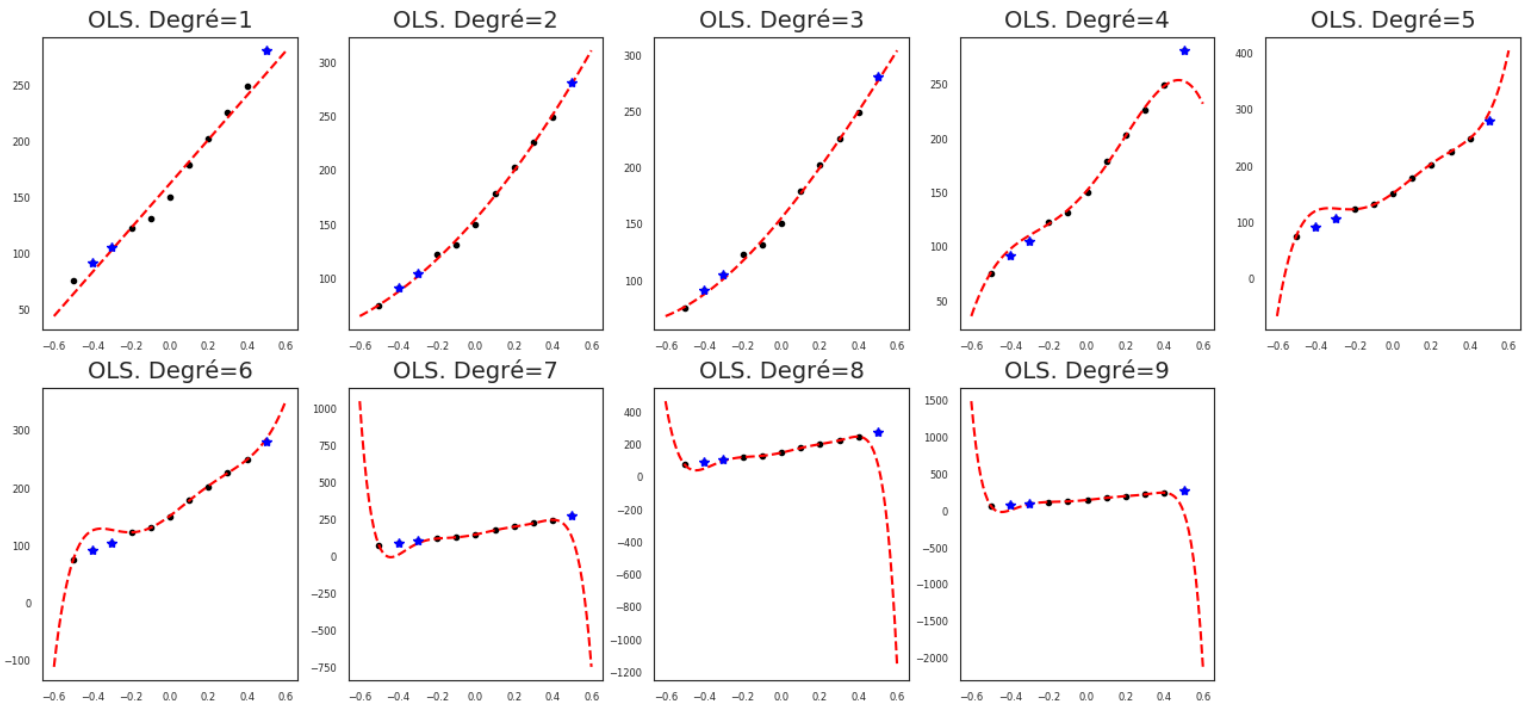
Overfitting is a common cause of poor machine learning algorithms.

- First solution: dimension reduction by creating new variables (PCA, t-SNE, etc).
- Second solution: select a subset of input variables, for instance by shrinking the coefficients of some variables.

Overfitting, a simple example

Population growth in US, 1900-2010

- Polynomial model $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_j x^j + \epsilon$
- The blue points are not in the learning dataset (n=9, p=j)



- If the number j of explanatory variable is too high, overfitting occurs (the red curve diverges).

Outline

Features selection: best subset selection

- Best subset regression finds, for fixed $q \in \{0, 1, 2, \cdots, p\}$, the subset of size q that gives smallest *RSS* or *RMSE*.
- Drawback: computational cost is huge (infeasible for $p > 40$).

- Rather than search through all possible subsets, we can seek a good path through them.

Feature selection: forward/backward selection

- Forward stepwise selection is a greedy algorithm, producing a nested sequence of models.
  It starts with the intercept, and then sequentially adds into the model the predictor that gives the greatest additional improvement to the fit.
  Computational advantage over best subset selection is clear.
  But there is no guarantee to find the best model.
- Backward stepwise selection starts with the full model and sequentially drops the predictor that improves the fit the least.

- Models are compared with the AIC or BIC cirterai. If the residuals are supposed to be Gaussian,

$$AIC = RSS + 2\frac{q}{n}\hat{\sigma}_\epsilon^2$$

$$BIC = n \log(RSS/n) + q \log n$$

- Computational advantage over best subset selection is clear.
  But there is no guarantee to find the best model.

Feature selection: hybride forward/backward selection

- In practice, hybride forward/backaward feature selestion is used: at each step the procedure is allowed to add and to remove a variable.

- Simulation example

$$Y = X_1 + X_2 + X_3 + \epsilon$$

where $X_j \sim \mathcal{N}(0, 1)$, $j = 1, \cdots, p$ independant variables.
$n = 50$ independant observations $(\mathbf{x}_i, y_i)$ are simulated. THe feature selection methods are run and the selection is compared to the truth.
The experiment is repeated 100 times.

Rate of good selection

p	Forw.	Back.	Hybr.
5	0.91	0.91	0.91
15	0.48	0.42	0.48
30	0.16	0.13	0.16
45	0.04	0.00	0.04

Conclusions
- Forward and hybride methods work better than backward method.
- The performance decreases when the ratio p/n increases.

Outline

Regularization: shrinkage methods

- The subset selection methods use least squares to fit a linear model that contains a subset of the predictors. As an alternative, we can fit a model containing all p predictors using a technique that constrains or regularizes the coefficient estimates, or equivalently, that shrinks the coefficient estimates towards zero.
- The **Ridge** regression coefficient estimates β^R are the values that minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

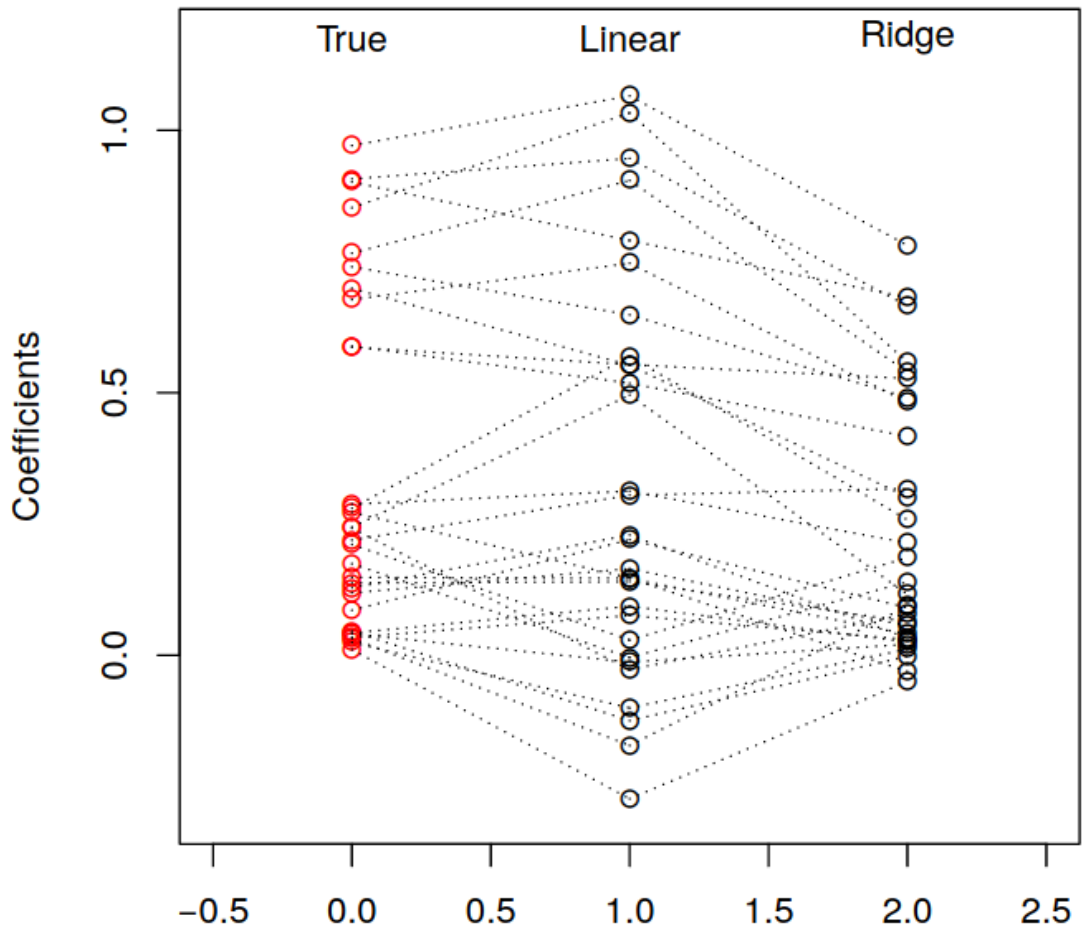
where $\lambda \geq 0$ is a tuning parameter, to be determined separately.

$$\beta_{Ridge} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbb{I})^{-1}\mathbf{X}^T y$$

- Here $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term. Note that:
  • When $\lambda = 0$, we get the linear regression estimate
  • When $\lambda = +\infty$, we get $\beta_{Ridge} = 0$
  • For λ in between, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients

Example: visual representation of ridge coefficients

Recall our last example ($n = 50$, $p = 30$, and $\sigma^2 = 1$; 10 large true coefficients, 20 small).
Here is a visual representation of the ridge regression coefficients for $\lambda = 25$:



The coefficients are shrinked to 0.

Important details

When including an intercept term in the regression, we usually leave this coefficient unpenalized. Otherwise we could add some constant amount c to the vector y, and this would not result in the same solution.
If we center the columns of X, then the intercept estimate ends up just being $\hat{\beta}_0 = \bar{y}$, so we usually just assume that y,X have been centered and don't include an intercept.

Also, the penalty term $\|\beta\|_2^2 = \sum_{j=1}^{p} \beta_j^2$ is unfair if the predictor variables are not on the same scale.
Therefore, if we know that the variables are not measured in the same units, we typically scale the columns of X (to have sample variance 1), and then we perform ridge regression

Bias and variance of ridge regression The bias and variance are not quite as simple to write down for ridge regression as they were for linear regression.
The general trend is:

- The bias increases as λ (amount of shrinkage) increases
- The variance decreases as λ (amount of shrinkage) increases

The choice of λ is important (it is discussed later).

Outline

Lasso regression

- The **Ridge** regression

$$\beta_{Ridge} = argmin \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

can have better prediction error than linear regression in a variety of scenarios, depending on the choice of λ.
But it will never sets coefficients to zero exactly, and therefore cannot perform variable selection in the linear model.

- The **Lasso** regression coefficient estimates β_{Lasso} are the values that minimize

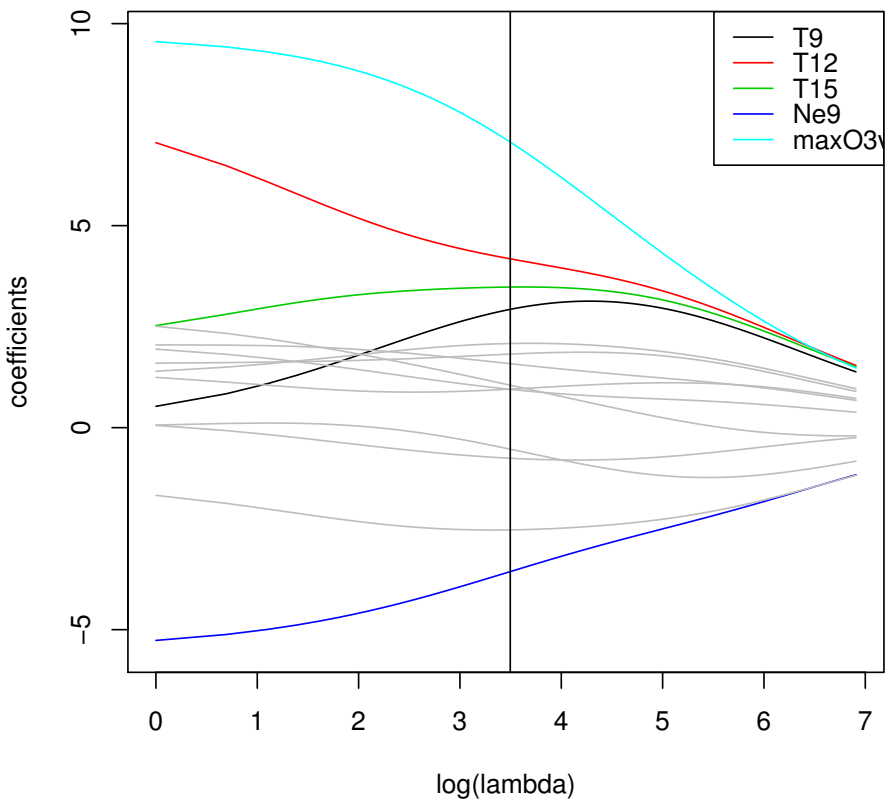
$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

The only difference between the lasso problem and ridge regression is that the latter uses a (squared) ℓ^2 penalty $\|\beta\|_2^2$, while the former uses an ℓ^1 penalty $\|\beta\|_1^2$. But even though these problems look similar, their solutions behave very differently.

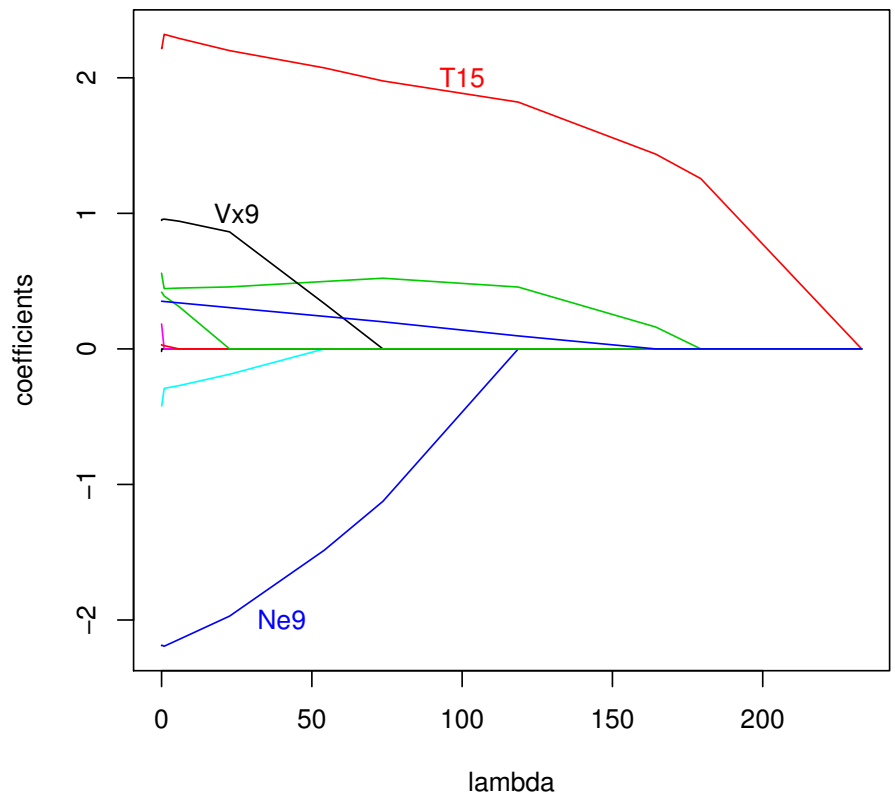
Ridge and Lasso coefficient paths for Daily Max Ozone dataset

Parameters values for different values of λ.
Lasso is to a proper feature selection method while Ridge only reduce the weights of the non informative inputs.

Ridge



Lasso



Lasso regression

$$\beta_{Lasso} = \underset{}{argmin} \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

The tuning parameter λ controls the strength of the penalty, and (like Ridge regression) we get

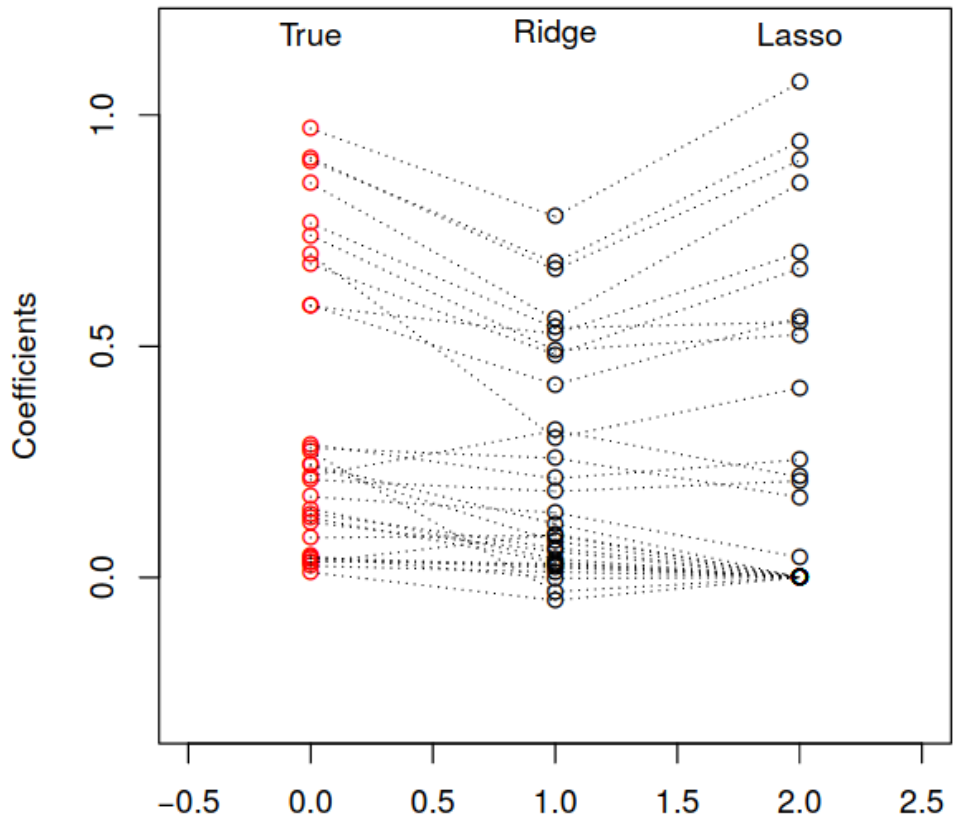
- β_{Lasso} = the linear regression estimate when $\lambda = 0$
- $\beta_{Lasso} = 0$ when $\lambda = \infty$
- For λ in between these two extremes, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients.

But the nature of the ℓ^1 penalty causes some coefficients to be shrunken to zero exactly.

This is what makes the lasso substantially different from ridge regression: it is able to perform variable selection in the linear model. As λ increases, more coefficients are set to zero (less variables are selected), and among the nonzero coefficients, more shrinkage is employed

Example: visual representation of lasso coefficients

Our running example from last time: $n = 50$, $p = 30$, $\sigma^2 = 1$, 10 large true coefficients, 20 small. Here is a visual representation of lasso vs. ridge coefficients (with the same degrees of freedom):



Important details

When including an intercept term in the model, we usually leave this coefficient unpenalized, just as we do with ridge regression.
As we've seen before, if we center the columns of X, then the intercept estimate turns out to be $\hat{\beta}_0 = \bar{y}$. Therefore we typically center y, X and don't include an intercept term.

As with Ridge regression, the penalty term $\|\boldsymbol{\beta}\| = \sum_{j=1}^{p} |\beta_j|$ is not fair is the predictor variables are not on the same scale. Hence, if we know that the variables are not on the same scale to begin with, we scale the columns of X (to have sample variance 1), and then we solve the lasso problem.

Bias and variance of Lasso regression The bias and variance are not quite as simple to write down for Lasso regression as they were for linear regression.
The general trend is:

- The bias increases as λ (amount of shrinkage) increases
- The variance decreases as λ (amount of shrinkage) increases

The choice of λ is important (it is discussed later).

In terms of prediction error (or mean squared error), the lasso performs comparably to ridge regression

Adaptive Lasso

- One of the drawback of the Lasso regularization is that the penalty is the same whatever the value of β_j.
- In adaptive Lasso the regularization constant is bigger if β_j seems to be close to zero. The a priori about the β_j value is given by $\hat{\beta}_j$.
- The adaptive Lasso estimates β^{AL} are the values that minimize

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\frac{|\beta_j|}{|\hat{\beta}_j|}$$

- The $\hat{\beta}_j$ are obtained from the regular linear regression estimate or a Ridge estimate or a Lasso estimate followed by a regular estimation gieven the selection.

Elasticnet

- Ridge regression does not allow to select feutures. But, it is known to lead to good predictions.
- Elasticnet combines Ridge and Lasso penalties.
- The Elasticnet estimates β^E are the values that minimize

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}(1-\alpha)\beta_j^2+\alpha|\beta_j|$$

Simulation example

- The true model is
$$Y = X_1 + X_2 + X_3 + \epsilon$$
where $X_j \sim \mathcal{N}(0, 1)$, $j = 1, \cdots, p$ independant variables.
- $n = 50$ independant observations $(\mathbf{x}_i, y_i)$ are simulated.
- The feature selection methods are run and the selection is compared to the truth.
- The experiment is repeated 100 times.
- Results

	Rate of good selection		
p	Hybr.	Lasso	Ad. Lasso
5	0.80	0.14	0.58
15	0.46	0.07	0.27
30	0.23	0.02	0.13
45	0.12	0.03	0.05

- Conclusions
-
-

Simulation example

- The true model is
$$Y = X_1 + X_2 + X_3 + \epsilon$$
where $X_j \sim \mathcal{N}(0, 1)$, $j = 1, \cdots, p$ independant variables.
- $n = 50$ independant observations $(\mathbf{x}_i, y_i)$ are simulated.
- The feature selection methods are run and the selection is compared to the truth.
- The experiment is repeated 100 times.
- Results

	Rate of good selection		
p	Hybr.	Lasso	Ad. Lasso
5	0.80	0.14	0.58
15	0.46	0.07	0.27
30	0.23	0.02	0.13
45	0.12	0.03	0.05

- Conclusions
  - The hybride selection method gives the best results.
  - Adaptive Lasso is much better than Lasso.

Simulation example

If our goal is only prediction (and not features selection).

- The prediction quality is measured by the MSE computed on an independant sample.
- Results

Mean Square Error (standard deviation)

p	Hybr.	Lasso	Ad. Lasso	Ad. Lasso+OMS[1]
5	1.1(0.2)	1.2(0.3)	1.2(0.3)	1.2(0.3)
15	1.2(0.3)	1.3(0.3)	1.2(0.3)	1.3(0.3)
30	1.4(0.4)	1.3(0.4)	1.4(0.4)	1.5(0.5)
45	1.6(0.6)	1.5(0.5)	1.7(0.6)	1.8(0.7)

- Conclusions
  -
  -
  -

[1] Ordinary Mean Square is sometimes run given the Lasso selection to remove the bias occurring in Lasso estimates.

Simulation example

If our goal is only prediction (and not features selection).

- The prediction quality is measured by the MSE computed on an independant sample.
- Results

Mean Square Error (standard deviation)

p	Hybr.	Lasso	Ad. Lasso	Ad. Lasso+OMS[2]
5	1.1(0.2)	1.2(0.3)	1.2(0.3)	1.2(0.3)
15	1.2(0.3)	1.3(0.3)	1.2(0.3)	1.3(0.3)
30	1.4(0.4)	1.3(0.4)	1.4(0.4)	1.5(0.5)
45	1.6(0.6)	1.5(0.5)	1.7(0.6)	1.8(0.7)

- Conclusions
  - The prediction based on the hybride method and Lasso lead to the smallest errors.
  - The OMS fit performed after an Adaptive Lasso selection tends to increase the prediction error.
  - The errors increase with p.

[2] Ordinary Mean Square is sometimes run given the Lasso selection to remove the bias occurring in Lasso estimates.

Simulation example

If our goal is only prediction (and not features selection).

- The prediction quality is measured by the MSE computed on an independant sample.
- Results

Mean Square Error (standard deviation)

p	Hybr.	Lasso	Ad. Lasso	Ad.Lasso+OMS	Ridge	Elasticnet
5	1.1(0.2)	1.2(0.3)	1.2(0.3)	1.2(0.3)	1.1 (0.2)	1.1(0.2)
15	1.2(0.3)	1.3(0.3)	1.2(0.3)	1.3(0.3)	1.4(0.4)	1.3 (0.3)
30	1.4(0.4)	1.3(0.4)	1.4(0.4)	1.5(0.5)	2.0(0.4)	1.4(0.4)
45	1.6(0.6)	1.5(0.5)	1.7(0.6)	1.8(0.7)	2.3(0.6)	1.5 (0.4)

- Conclusions
  - Ridge leads to the worse predictions in this examples (with independant variables). - Elaticnet has the same errors as Lasso.

Choice of the regularization constant

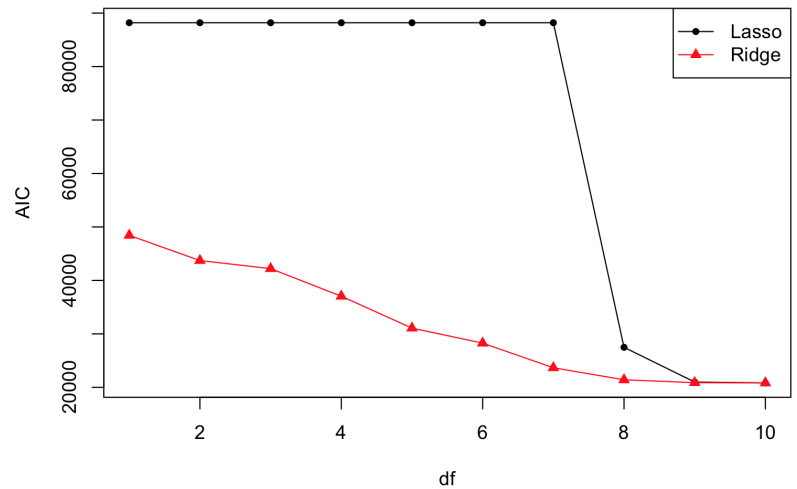
- Main difficulty with Ridge and Lasso methods is to choose the regularization constant λ

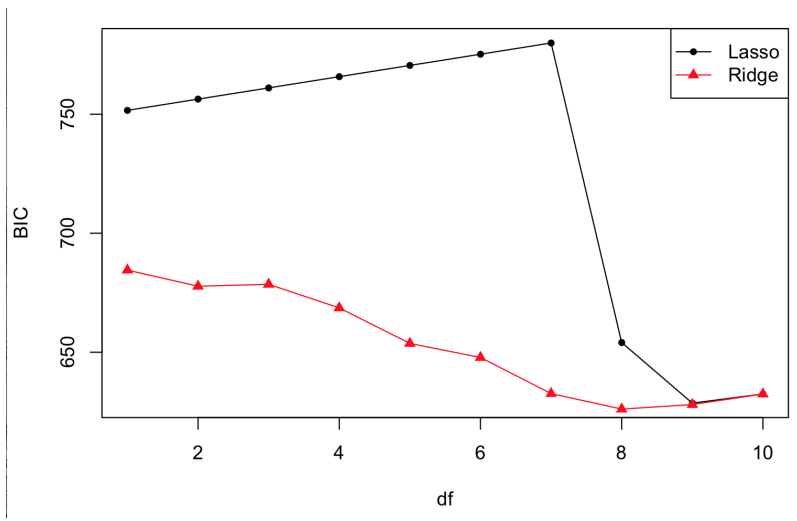
$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

- If λ is small the shrinkage is low and we keep a lot of variables in the model $\rightarrow$ high variance
- If λ is large the shrinkage is strong and we do not keep enough variables in the model $\rightarrow$ high bias.
- Methods to choose λ
  - RMSE computed by CV, expensive but accurate
  - AIC or BIC criteria, cheap but less accurate

BIC and AIC for Ozone data

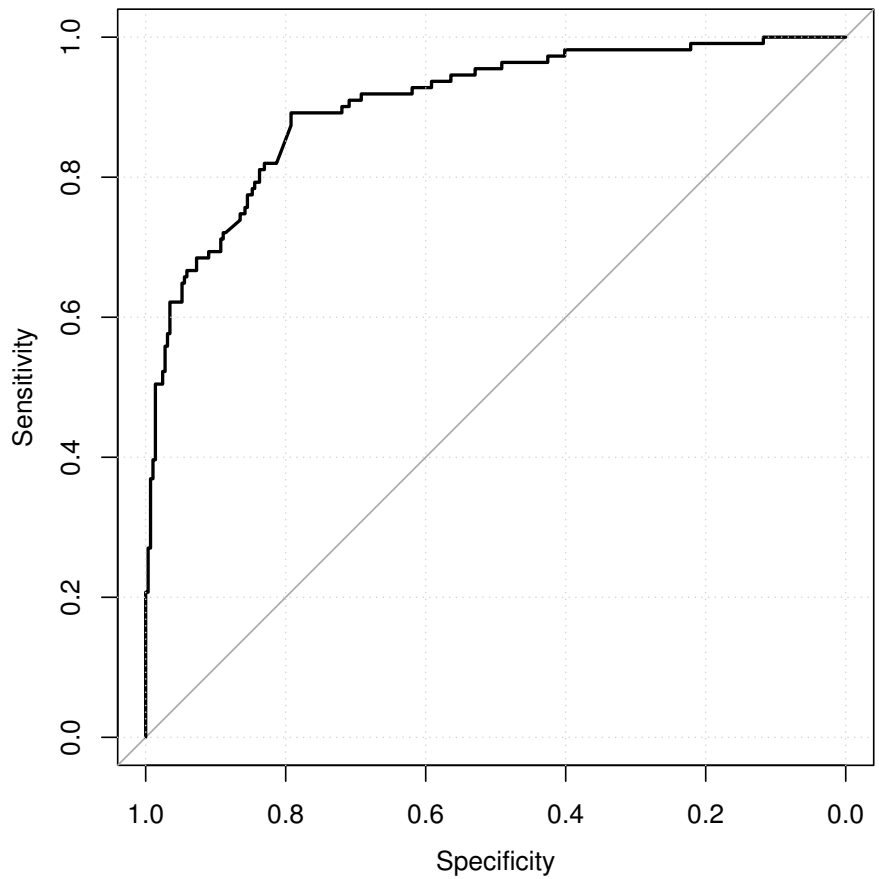
- BIC usually leads to more parsimonious models than AIC.
- BIC keeps more inputs than CV's RMSE.
  BIC: 8-9, CV Ridge: around 4, CV Lasso: 6-7





Logistic regression, LASSO

- All the selections methods can be used in the context of logistic regression.
- Lasso model based on the genes with highest absolute correlation with class AML/ALL ($|\rho| > .5$)
- n.train = 50%, AUC = 0.90



Recap

Ridge (resp. Lasso regression) minimizes the usual regression criterion plus a penalty term on the squared ℓ^2 (resp. ℓ^1) norm of the coefficient vector. As such, it shrinks the coefficients towards zero. This introduces some bias, but can greatly reduce the variance, resulting in a better mean-squared error

The amount of shrinkage is controlled by λ, the tuning parameter that multiplies the penalty. Large λ means more shrinkage, and so we get different coefficient estimates for different values of λ. Choosing an appropriate value of λ is important, and also difficult. It should be done by cross-validation.

Ridge regression performs particularly well when there is a subset of true coefficients that are small or even zero. It doesnâĂŹt do as well when all of the true coefficients are moderately large; however, in this case it can still outperform linear regression over a pretty narrow range of (small) λ values.

The Lasso estimates are generally biased, but have good mean squared error (comparable to Ridge regression). On top of this, the fact that it sets coefficients to zero can be a big advantage for the sake of interpretation.

QUIZZ - Question no 1

Imagine a linear model with 100 input features: 10 are highly informative. 90 are non-informative. Assume that all features have values between -1 and 1. Which of the following statements are true?

1. L1 regularization may cause informative features to get a weight of exactly 0.0.
2. L1 regularization will encourage many of the non-informative weights to be nearly (but not exactly) 0.0.
3. L1 regularization will encourage most of the non-informative weights to be exactly 0.0.

QUIZZ - Respons no 1

Imagine a linear model with 100 input features: 10 are highly informative. 90 are non-informative. Assume that all features have values between -1 and 1. Which of the following statements are true?

1. L1 regularization may cause informative features to get a weight of exactly 0.0.
   Be careful–L1 regularization may cause the following kinds of features to be given weights of exactly 0:
   - Weakly informative features.
   - Strongly informative features on different scales.
   - Informative features strongly correlated with other similarly informative features.
2. L1 regularization will encourage many of the non-informative weights to be nearly (but not exactly) 0.0.
3. L1 regularization will encourage most of the non-informative weights to be exactly 0.0.
   L1 regularization of sufficient lambda tends to encourage non-informative weights to become exactly 0.0. By doing so, these non-informative features leave the model.

QUIZZ - Question no 2

Imagine a linear model with 100 input features: 10 are highly informative. 90 are non-informative. Which type of regularization will produce the smaller model?

1. Lasso regularization.
2. Ridge regularization.

QUIZZ - Respons no 2

Imagine a linear model with 100 input features: 10 are highly informative. 90 are non-informative. Which type of regularization will produce the smaller model?

1. Lasso regularization.
   L1 regularization tends to reduce the number of features. In other words, L1 regularization often reduces the model size.
2. Ridge regularization.

QUIZ - Question no 3

Imagine a linear model with two strongly correlated features; that is, these two features are nearly identical copies of one another but one feature contains a small amount of random noise. If we train this model with L2 regularization, what will happen to the weights for these two features?

1. Both features will have roughly equal, moderate weights.
2. One feature will have a large weight; the other will have a weight of almost 0.0.
3. One feature will have a large weight; the other will have a weight of exactly 0.0.

QUIZ - Respons no 3

Imagine a linear model with two strongly correlated features; that is, these two features are nearly identical copies of one another but one feature contains a small amount of random noise. If we train this model with L2 regularization, what will happen to the weights for these two features?

1. Both features will have roughly equal, moderate weights.
   L2 regularization will force the features towards roughly equivalent weights that are approximately half of what they would have been had only one of the two features been in the model.
2. One feature will have a large weight; the other will have a weight of almost 0.0.
3. One feature will have a large weight; the other will have a weight of exactly 0.0.

Linear model, concluding remarks

Linear models

- Approximate the relation between $\mathbf{X}$ and y by a linear form

y is quantitative,	y is qualitative,
$E[y] = \beta_0 + \boldsymbol{\beta}\mathbf{x}$,	$E[y] = P(Y = 1) = g(\beta_0 + \boldsymbol{\beta}\mathbf{x})$,
No numerical optimization for fitting	Numerical optimization for fitting
Validation score: rmse	Validation score: classification error, ROC curve

- Linear models are elementary blocks for other ML methods.

For all models/methods

- Use (Monte Carlo) cross validation to validate the models.
- Variable selection is important to prevent overfitting.
- Different approaches for features selection: BIC, greedy algorithms, regularization. The hybride forward-backward method is often the best for features selection. The regularization methods are easiest to use in some framwork as for instance neural networks, mixture models, ...

Outline