Applied Data Science Lunch Lecture

Regularization

Maarten Cruyff

Content

1. Prediction
2. Regularization
   - ridge
   - lasso
3. Example

Prediction

Training

Prediction

1. Train the model on data at hand
2. Predict unknown outcome on future data

Examples

- diagnosis of disease based on symptoms
- spam based on content of email

GLM

Model

- generalized linear model

$$g(y) = \boldsymbol{x}'\beta + \epsilon$$

Parameter estimation

- find $\hat{\beta}$ that minimizes MSE / deviance

Under- and overfitting

Too few predictors in the model

- relevant predictors are missing
- parameter estimates are **biased**
- poor predictions on new data

Too many predictors in the model

- capitalization on chance, spurioussness, multicollinearity
- parameter estimates have **high variance**
- poor predictions on new data

Bias versus Variance

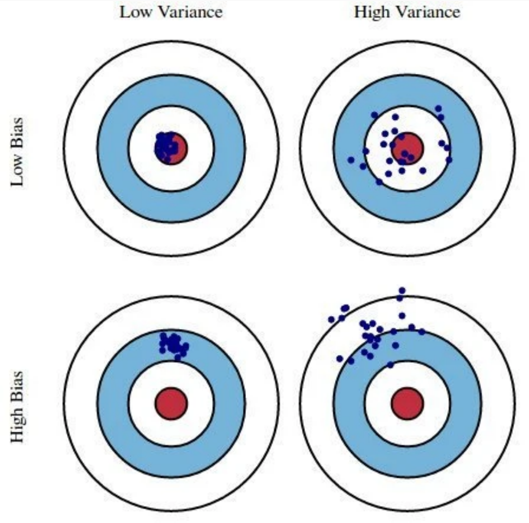


Figure 1: Hitting the bull's eye.

Bias-Variance Tradeoff

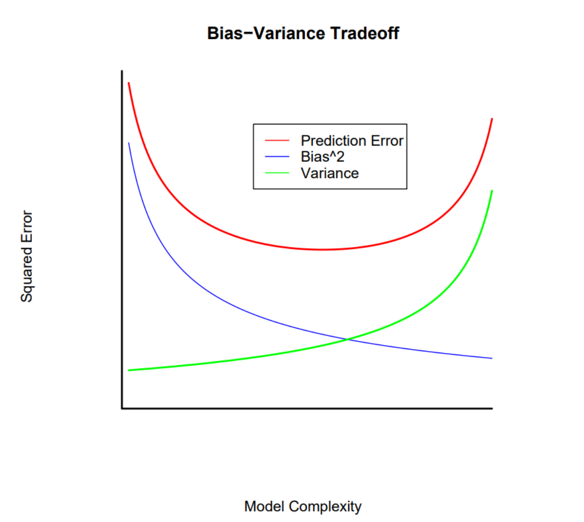


Figure 2: Optimal prediction is compromise between bias and variance.

How to find the optimum?

Data science techniques

- stepwise procedures (AIC/BIC)
- **regularization**
- GAM's
- trees
- boosting/bagging
- support vector machines
- deep learning

Regularization

Lasso and ridge

Regularization

- penalizing MSE/deviance with size parameter estimates

Lasso defined by ℓ_1 penalty $\lambda \sum_{j=1}^{p} |\beta_j|$

- shrinks parameters **to** 0

Ridge defined by ℓ_2 penalty: $\lambda \sum_{j=1}^{p} \beta_j^2$

- shrinks parameters **towards** 0

- λ controls amount of shrinkage
- predictors are standardized

Regularization vs Stepwise

Stepwise procedures

- penality on **number** of parameters (AIC/BIC)
- no hyperparameter to be estimated

Regularization

- penality on **size** of parameters
- optimal shrinkage parameter to be estimated

Train/dev/test

1. Partition the data in training/test set
2. Cross validate λ's on train/validation set
3. Choose λ with smallest averaged deviance (or $+1$ SD)
4. Compare deviance test with competing models

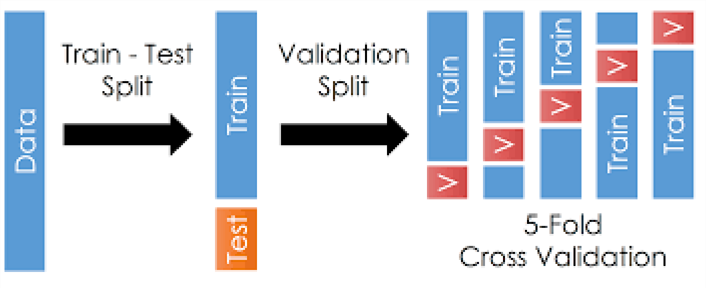


Figure 3: Train/dev/test

R package `glmnet`

`glmnet()`

- fast algorithm to compute shrinkage for sequence λ
- plot parameter shrinkage as function λ

`glmnet.cv()`

- performs k-fold cross validation to determine optimal λ
- plot averaged deviance as function λ

Example

Spam filter

Classify email as spam/nonspam

Response variable

- 2788 mails classified as “nonspam”
- 1813 mails classifed as “spam”

57 standardized frequencies of words/characters, e.g.

- !, $, (), #, etc.
- make, all, over, order, credit, etc.

The model

Logistic regression model

$$\text{logit}(\pi) = \boldsymbol{x}'\boldsymbol{\beta}$$

where π is the probability of spam.

Testing for interactions:

- 2-way: 1596 additional parameters
- 3-way: 29260 additional parameters

Restrict models to 2-way

Model comparisons

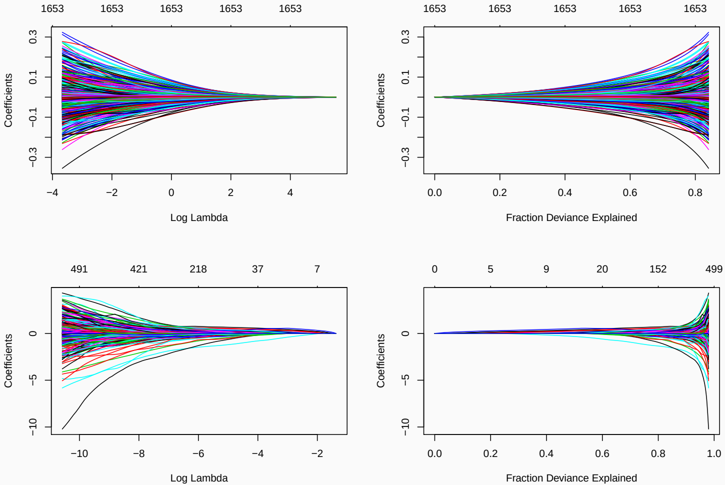
Models

- main-effects with `glm()`
- stepwise with `step()`
- ridge with `glmnet()`
- lasso with `glmnet()`
- full 2-way with `glm()`

Which model has lowest deviance on test set?

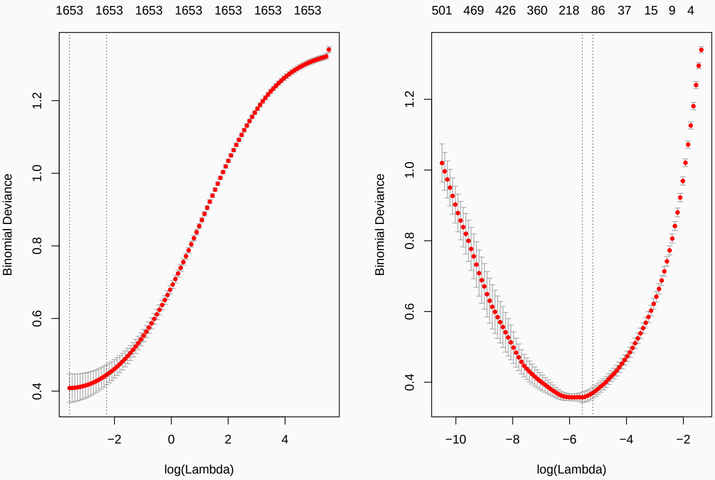
Shrinkage ridge (top) and lasso (bottom)

Results for training set (no cross validation)



Averaged deviance ridge (left) and lasso (right)

Results cross validation



Results on test set

	Deviance	Error rate	#pars	L1-norm
main effects	269.7	6.3	58	104.9
ridge	246.7	7.2	1653	39.3
lasso	213.1	6.3	108	14.6
stepwise	572.9	7.7	129	3554.1

- lasso

	nonspam	spam
nonspam	665	32
spam	40	414

- main

	nonspam	spam
nonspam	666	31
spam	41	413

Conclusions

Regularization

- reduces variance without substantially increasing bias
- ability to handle large number of predictors
- fast algorithm

Extensions

- mixing ℓ_1 and ℓ_2 penalties (e.g. elastic net)
- grouped lasso (e.g. hierarchical models)
- similarities with Bayesian models

Thanks for your attention!