Machine learning Part 2

Jérôme Dockès

QLS course 2021-07-30

Recap of part 1

Supervised learning

- Regression: least-squares linear regression
- Classification: logistic regression

Regularization

- ℓ_2 a.k.a. ridge regularization

Model evaluation and selection

- Out-of-sample generalization; independent test set
- Performance metrics:
    - regression: mean squared error
    - classification: accuracy, ROC curve
- Cross-validation

Don’t remember? watch Part 1 again!

Notation & vocabulary

Supervised learning framework

$$Y = f(X) + E \tag{1}$$

- $Y \in \mathbb{R}$: output (a.k.a. target, dependent variable) to predict
- $X \in \mathbb{R}^p$: features (a.k.a. inputs, regressors, descriptors, independent variables)
- $E \in \mathbb{R}$: unmodelled noise
- f: the function we try to approximate

Example: linear regression

$$\begin{aligned} Y &= \beta_0 + \langle X, \beta \rangle + E & (2) \\ &= \beta_0 + \sum_{j=1}^{p} X_j \, \beta_j + E & (3) \end{aligned}$$

"learning" = estimating $\beta_0 \in \mathbb{R}$ and $\beta \in \mathbb{R}^p$

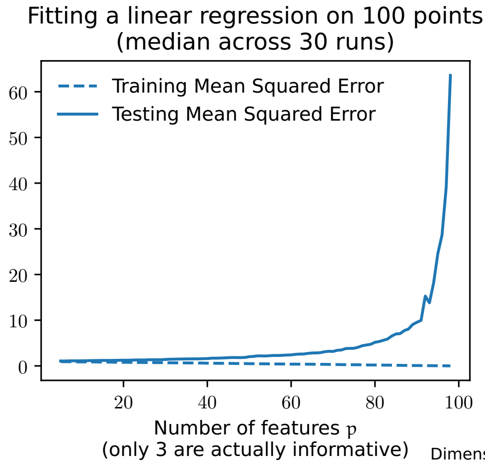
Dimensionality reduction

Problems when the number of features p becomes large

- Bigger errors on test data (larger variance of predictions)
- Numerical stability issues
- Computational cost and memory usage

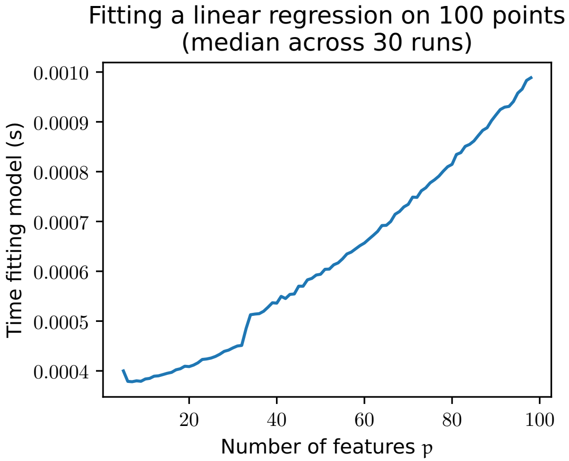
Model complexity: overfitting

- Model complexity increases with dimension.
- Example: a linear model in dimension p can fit exactly (0 training error) any set of $p + 1$ points.
- Risk of overfitting: fitting exactly training data but failing on test data



Cost of fitting many parameters

- Many algorithms require polynomial time in p
- Implementations often make copies of the design matrix (e.g. for centering & rescaling)



Univariate feature selection

- a.k.a. feature screening, filtering . . .
- Check features (columns of X) one by one for association with the output y
- Keep only a fixed number or percentage of the features

Simple (linear) association criteria

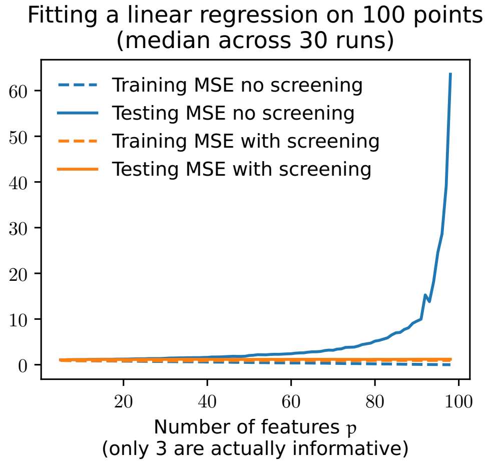
- for regression: correlation
- for classification: ANalysis Of VAriance

Read more in the scikit-learn user guide

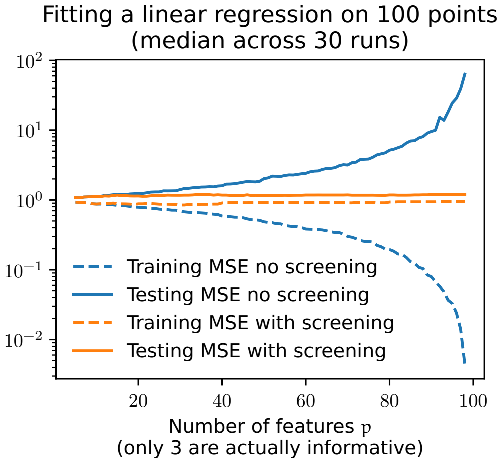
https://scikit-learn.org/stable/modules/feature_selection.html#feature-selection

Univariate feature selection

Keeping only the 10 best features (most correlated with y)

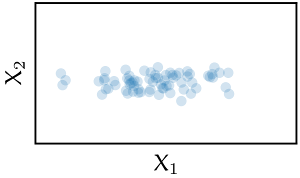


Same plot in log scale

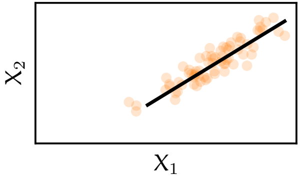


Linear decomposition methods

Maybe OK to drop X_2:



Data low-dimensional but no feature can be dropped:



Find a better referential in which to represent the data

Linear regression: projection on the column space of X

Approximate y as a combination of the columns of X

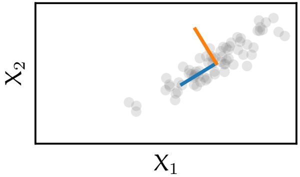
$$\hat{\mathbf{y}} = \mathbf{X}\,\hat{\beta} \in \mathbb{R}^n \tag{4}$$

- The columns of X are a family of p n-dimensional vectors
- When p is high or the columns of X are correlated, we want to use a family of $k < p$ instead
- Feature selection: drop some columns, keep only k
- Could we build a better family of k vectors?

Principal Components Regression

- Approximation of X of rank k: find a family of k basis vectors and approximate each column of X as a mixture of these k vectors
- Find the family that gives the best approximation: the one with the smallest Frobenius norm of the reconstruction error.
- This is the same as finding the k orthogonal directions in which X varies the most

Principal Components: feature space



Ridge regression and PCA

- Both ridge regression and PC regression compute the coordinates of y in the basis given by the SVD of X
- ridge shrinks the coefficients of sv d_j by a factor $d_j^2/(d_j^2+\lambda)$
- PC regression sets the coefficient to 0 for all but the k largest d_j

Other decomposition methods

- Take y into account
- Different criteria (sparsity, independence, . . .)

Nested cross-validation: setting hyperparameters

How can we choose:

- Number of features or PCA components k?
- The ridge hyperparameter λ?

Try a few and pick the best one... But measure its performance on separate data!

Some common pitfalls with cross-validation

- Ignoring dependencies between samples
- Ignoring dependencies between CV scores
- Over-interpreting good CV scores

Two sources of variance: training data and test sample

Don’t use Leave-One-Out Cross-validation

fMRI decoding

- Describe data and task

The decoding pipeline

- Masking: extracting voxels that are inside the brain
- Feature selection with ANOVA
- Classifier: logistic regression

Implementation: in class