Computational Data Analysis (02582)

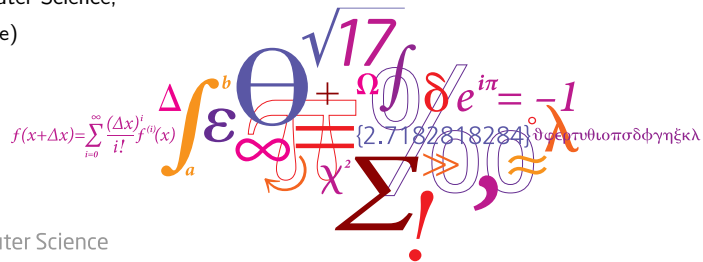
MODEL SELECTION & ASSESSMENT

Sneha Das

Assistant Professor

Department of Applied Mathematics and Computer Science,

Technical University of Denmark (DTU Compute)



DTU Compute

Department of Applied Mathematics and Computer Science

Outline I

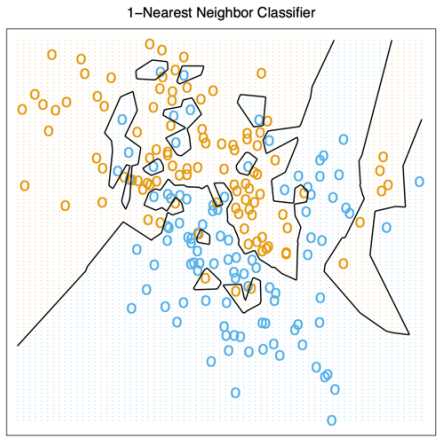
- Part I Method recap: K-nearest neighbours (KNN)
- Recap
- Part II - How do we choose & Justify models
  - Overview: Model Assessment and Validation
  - Model selection
  - Model Assessment

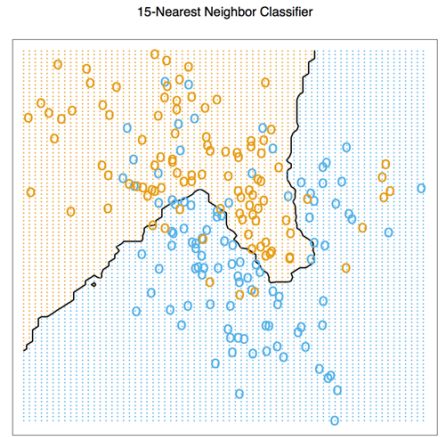
KNN classification

Classify observations according to the **majority class** of the K nearest neighbors.

- Define a distance measure of proximity between observations, eg Euclidean distance.
- It is general practice to standardize each variable to mean zero and variance 1.
- K is a positive integer of your choice. Small values give low bias, large values will give low variance.

KNN classification example - ESL





K-Nearest-Neighbors regression

We can use the same technique for regression!
Estimate the response of an observation as the **average response** of the K nearest neighbors.

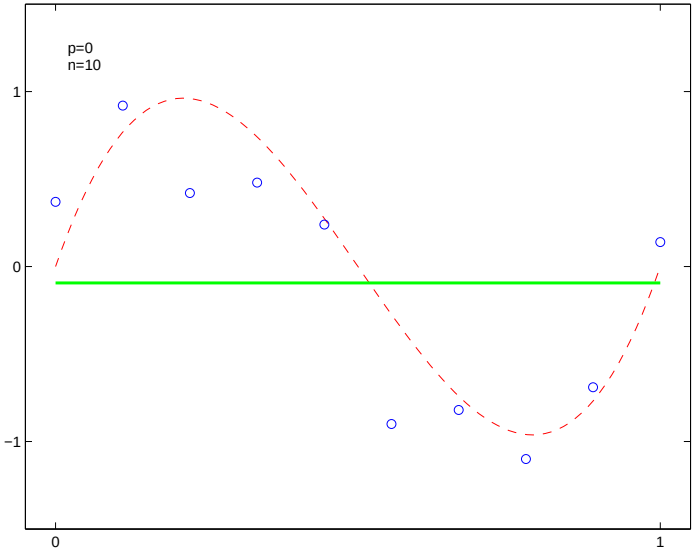
- Define a distance measure of proximity beqween observations, eg Euclidean distance.
- It is general practice to standardize each variable to mean zero and variance 1.
- K is a positive integer of your choice. Small values give low bias, large values will give low variance.

Over- and underfitting

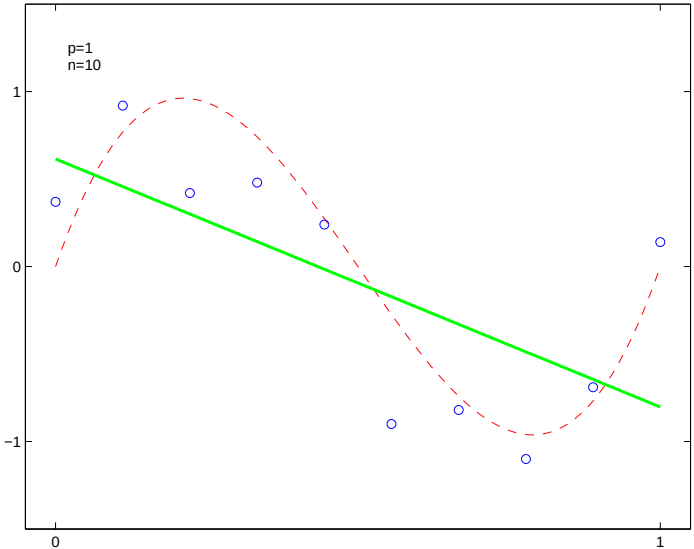
- We prefer a simple model.
- We prefer models that work (low EPE).
- These properties may contradict

Too simple: Our data set will not be accurately described. Model assumptions are wrong.
Too complex: The model becomes too flexible. We fit noise in data. We need lots of data to support such a model.

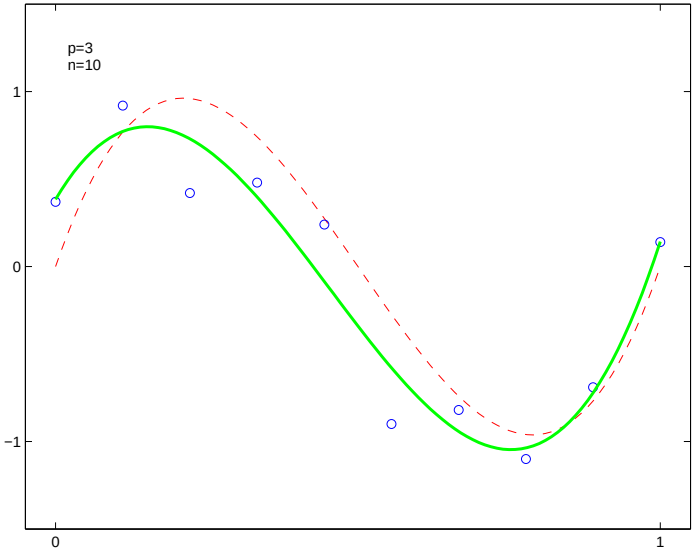
0th order polynomial



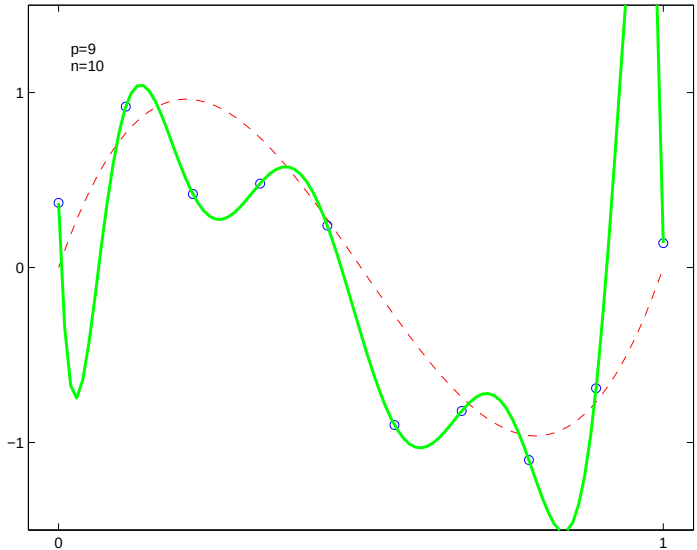
1st order polynomial



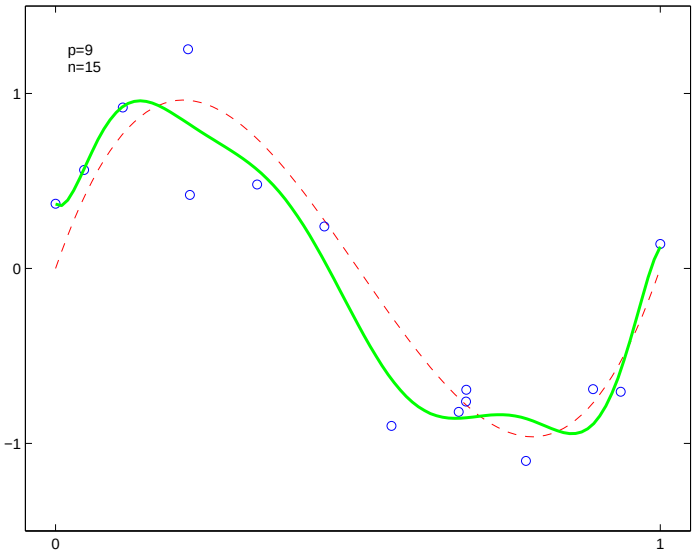
3rd order polynomial



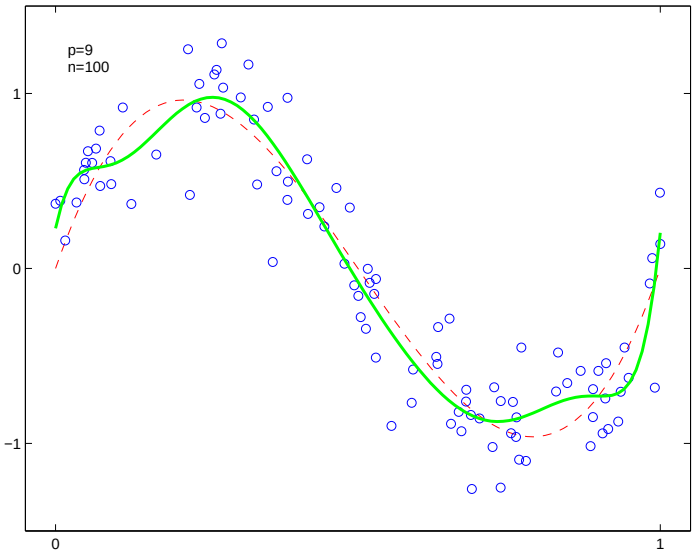
9th order polynomial



Data set size



Data set size



Ridge regression

We can use a flexible model (such as the 9th order polynomial before) and **avoid overfitting via regularization**.

Quadratic norm,

$$\beta_{Ridge} = \arg\min_{\beta} ||Y - X\beta||_2^2 + \lambda||\beta||_2^2$$

with simple solution,

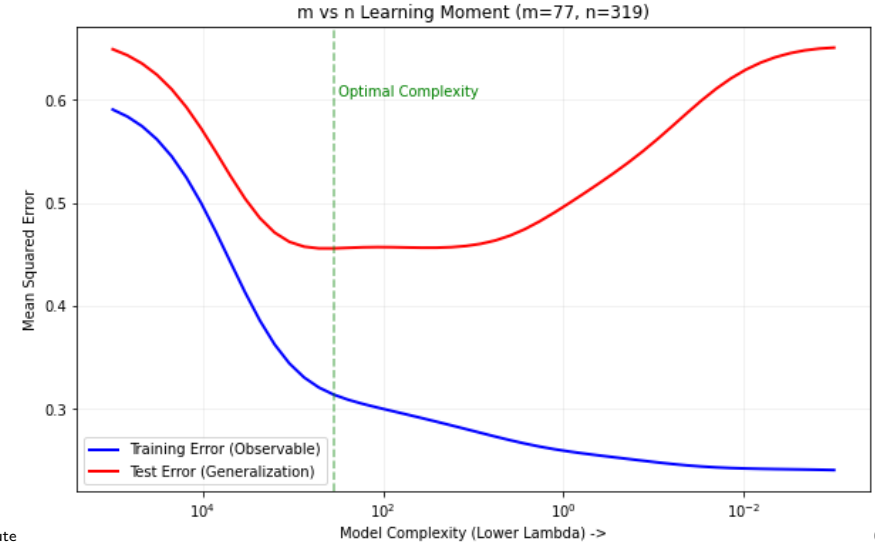
$$\beta_{Ridge} = (X^T X + \lambda I)^{-1}(X^T Y)$$

Regularization parameter λ,

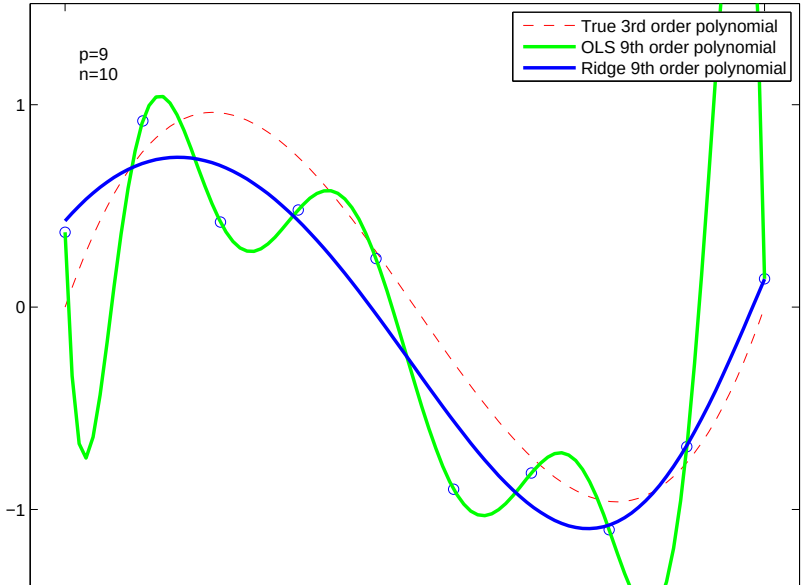
- $\lambda = 0$ gives $\beta_{Ridge} = \beta_{OLS}$
  - No bias
  - High variance
- $\lambda \to \infty$ gives $\beta_{Ridge} \to 0$
  - High bias
  - No variance

Ridge regression

Error with respect to complexity ($M \approx N$)



Optimal ridge model



Controlling the Complexity 'Knob'

How do we move along the Bias-Variance curve?

Method	Complexity Knob	High Complexity
OLS	Fixed (m)	Large m
Ridge	Regularization λ	Small $\lambda \to 0$

- **Regularization:** Deliberately adds bias to reduce variance.

Model selection and assessment

The more complex the model, the better we will fit the training data. Often we overfit to the training data.
Overfit models can perform poorly on test data - high variance.
Underfit models can perform poorly on test data - high bias.
We perform

1. **Model selection:** To choose a value of a tuning parameter or to choose between models.
2. **Model assessment:** To assess our chosen model, e.g. estimate the future prediction ability of the chosen model.

For both of these purposes, the best approach is to evaluate the procedure on an independent test set.
If possible one should use different test data for (1) and (2) above. A **validation set** for (1) and a **test set** for (2).

Model selection

For well-conditioned problems, divide data into three parts

Training set

Used to build different models, e.g. with different values of the tuning parameter λ.

Validation set

Used to calculate $E\hat{P}E$ for the different models and select the best, i.e. used to tune parameters, select features and similar *modeling decisions*. Also called **dev** (development) set, or **hold-out cross validation** set.

Test set

Used to estimate/test how well the resulting model performs, i.e. *evaluate performance* of algorithm, but make no decisions based on this data - Only report.

Model selection

Andrew Ng's *Machine Learning Yearning* (a good read for practitioners)

- Dev and test set should reflect data you expect to get in the future (i.e. the goal of the modeling)
- Dev and test sets must come from the same distribution
- Alternatively use a dev$_{train}$ and a dev$_{test}$ set

Making sure the split isn't critical

`for m=1:R` repetitions
- Randomize data (permute)
- Split data in 3 (train, validation, test)
- Train model on range of tuning parameters using train data
- Select best model based on validation data
- Test model to estimate the error on test set

`end` Calculate mean and std error over R test errors

Cross-validation

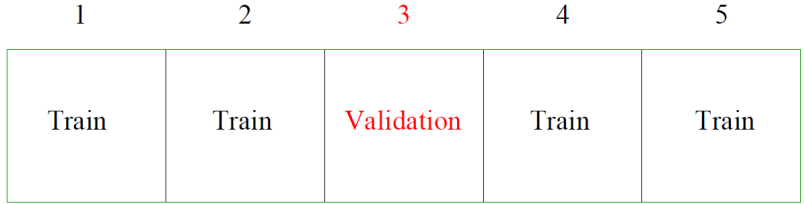
Often there is insufficient data to create a separate validation and test set. In this instance use cross validation instead.

Cross-validation:

- The primary method for estimating a tuning parameter, e.g. λ.
- No underlying assumptions
  - Except, that observations are assumed to be independent.
- Is simple, intuitive and easy to implement

K-fold cross-validation

- Split randomized data into K (roughly) equal parts



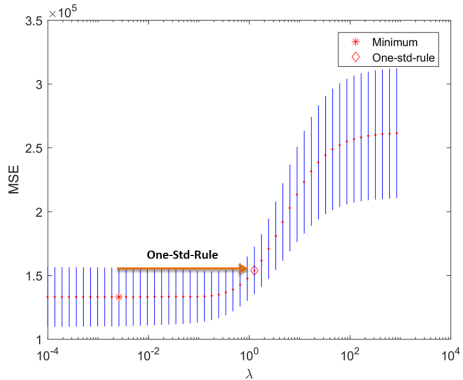
- For each $k = 1, 2, ..., K$, fit the model with parameter λ to the other $K - 1$ parts, giving $\hat{\beta}^{\neg k}(\lambda)$ and compute its error in predicting the kth part:

$$Err_k(\lambda) = \sum_{i \in k\text{th part}} (y_i - X_i \hat{\beta}^{\neg k}(\lambda))^2$$

This gives the cross-validation error

$$CV(\lambda) = \frac{1}{K} \sum_{k=1}^{K} Err_k(\lambda)$$

CV Model Selection



- Estimated prediction error curves, computed via cross validation.
- Bars indicate estimated standard errors.
- Vertical line chosen by 'one standard error rule'.

One standard error rule: Choose the smallest model whose error is no more than one standard error above the error of the best model. This compensates that the CV procedure chooses somewhat too complex models.

Variance estimates in K-fold cross-validation

We often use the standard error to give us a confidence about the mean (the cross-validation error from the previous slide)

$$CV(\lambda) = \frac{1}{K}\sum_{k=1}^{K} Err_k(\lambda)$$

$$S.E.(\lambda) = \frac{1}{\sqrt{K}}\sqrt{\frac{1}{K}\sum_{k=1}^{K}(Err_k(\lambda) - CV(\lambda))^2}$$

NOTE: This is a **biased variance estimate** as the observations are correlated! The variance is underestimated - thus be careful in trying to test for significant differences beteween two models (Bengio 2004).

Exercise: The One-SE Rule (5 mins)

The Data:
Your 10-fold CV results for Ridge Regression:

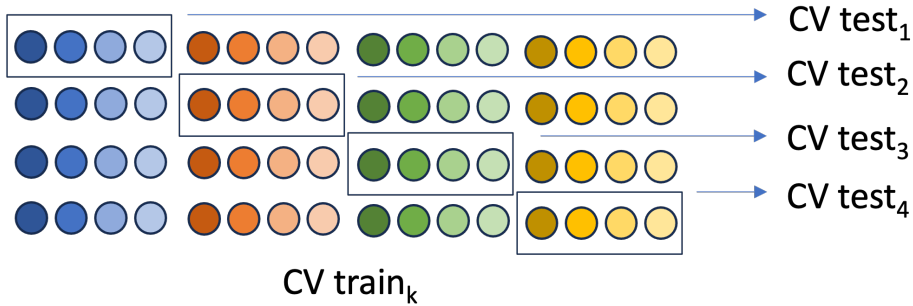
- **Minimum Error:** 1.2 ($\lambda = 0.1$)
- **Standard Error (SE):** 0.2
- **Model B Error:** 1.35 ($\lambda = 1.0$)
- **Model C Error:** 1.50 ($\lambda = 10.0$)

Group Task:
Calculate the selection threshold. Which λ setting is the most 'Scientific' choice?

Vevox Poll Open

The Choice?
Pick the λ value that an auditor would justify.

Leave-one-group-out cross-validation



When would you use Leave-k-groups-out?

Considerations

Observations are supposed to be independent. Otherwise information will leak from validation set to training set and we will overfit.

- Permute data before splitting in folds - the data set might be sorted in some way.
- Normalize training data separately for each fold.
- Impute missing values separately for each training set.
- If observations are sampled in groups, let each group go into the same fold.
- Be extremely careful with data sampled from dynamic systems (time-dependent data).
- Be careful to perform ALL pre-processing within the CV folds.

Be careful using leave-one-out CV ($K = N$), the folds are too similar. Use $K = 5$ or 10 as a good compromise between bias and variance.

Validation and test set issues

- Important to have both cross-validation and test sets, since we often run CV many times with different parameters. This can bias the CV results.
- A separate test set provides a convincing, independent assessment of a models performance (use only once!).
- Test set results might still overestimate actual performance, as a real future test set may differ in many ways from todays data.

Exercise: Spot the Leak! (5 mins)

The Investigation:
A model for clinical diagnosis follows this workflow:

Step 1 Normalization: Subtract mean/std of the *entire* dataset.

Step 2 Split: 80% Training, 20% Test.

Step 3 Tuning: Use 10-fold CV on Training set to find best λ.

Step 4 Predict: Evaluate on the Test set. Result: **99% Accuracy**.

Group Task:
Find the 'Data Leakage.' How did the test set bleed into the training?

Vevox Word Cloud

Identify the Step
Which step or action is the source of the leakage?

Information criteria

- Optimism and training error
- Information criteria
  - C_p-statistic
  - Akaike Information Criterion, AIC
  - Bayes Information Criterion, BIC

Optimism of the training error

The training error for an OLS regression is

$$\textbf{training-error}\ , \overline{err} = \frac{1}{N}\sum_{i=1}^{N}(y_i - x_i\hat{\beta})^2$$

If for each x_i can get a *new measurement* y_i^0 (Predicting *new* outcomes y_i^0 at the *same* locations x_i). What would the error for these new y_i^0 be?

$$\textbf{in-sample-error}\ , Err_{in} = \frac{1}{N}\sum_{i=1}^{N}(y_i^0 - x_i\hat{\beta})^2$$

Selection based on $\overline{err}$ leads to choosing the model that is best at memorizing noise, not signal.

Selecting the model with the smallest in-sample-error would be an effective model selection tool.

Optimism of the training error I

The difference between the **in-sample-error** and the **training-error** is called **optimism**.

$$op \equiv Err_{in} - \overline{err}$$

It is possible to show that quite generally

$$\textbf{expected optimism}, \omega \equiv E_y(op) = \frac{2}{N} \sum_{i=1}^{N} Cov(\hat{y}_i, y_i).$$

The more we overfit data the greater $Cov(\hat{y}_i, y_i)$ will be and thereby increasing the optimism.

- **High Complexity ($d \uparrow$):** $\hat{y}_i$ follows y_i more closely $\rightarrow$ Covariance increases $\rightarrow$ Optimism explodes.

Optimism of the training error II

In the Linear Case: With d variables and noise σ_ϵ^2:

$$E[Err_{in}] = E[\overline{err}] + 2 \cdot \frac{d}{N}\sigma_\epsilon^2$$

Therefore we have that

$$\textbf{expected in-sample-error} = \textbf{expected training-error} + 2\frac{d}{N}\sigma_e^2$$

To get an honest estimate, we must penalize the training error by the "cost of complexity" (d). **This is the foundation of C_p, AIC, and BIC.**

The C_p Statistic (estimate of in-sample error)

To find the 'best' model without splitting data, we apply a penalty to the training error.

The C_p Formula

$$C_p = \underbrace{\overline{err}}_{\text{Training Error}} + \underbrace{2 \cdot \frac{d}{N} \cdot \hat{\sigma}_e^2}_{\text{Complexity Penalty}}$$

- $\overline{err}$ **Expected training error,**: use the actual training error $\frac{1}{N}\sum_{i=1}^{N}(y_i - x_i\hat{\beta})^2$ as an estimate.
- d **(Complexity):** Number of variables in the current model.
- $\hat{\sigma}_e^2$ **(Noise Floor):** The MSE of a **low-bias model** (e.g., OLS with all features). This represents the irreducible noise in the system.
- **Selection Rule:** Choose the model that **minimizes** C_p.

The Trade-off:

Adding a variable ($d \uparrow$) always decreases $\overline{err}$, but it *increases* the penalty.

C_p identifies the point where the benefit of a new feature is outweighed by the cost of its complexity.

Akaike Information Criterion

AIC is similar to C_p, but useful when a log-likelihood loss function (logL) to estimate the error term (maximized log-likelihood)

$$AIC = -\frac{2}{N}\text{logL} + 2\frac{d}{N}$$

For the Gaussian case with a tuning parameter λ we define

$$AIC(\lambda) = \overline{err}(\lambda) + 2\frac{d(\lambda)}{N}\hat{\sigma}_e^2.$$

- For the Gaussian model C_p and AIC are identical.
- $d(\lambda)$ is the effective number of parameters in the model tuned with λ.

Effective number of parameters

For a linear fitting method

$$\hat{Y} = SY$$

we can calculate the effective number of parameters as

$$df(S) = trace(S)$$

ie the sum of the diagonal elements of S.
Both OLS and ridge regression are linear fitting methods,

$$\begin{aligned}\hat{Y} &= X\hat{\beta} \\ &= X(X^T X)^{-1} X^T Y \\ &= SY\end{aligned}$$

Bayes Information Criterion

The Bayes Information Criterion, BIC, is like AIC based on a log likelihood loss function. It is motivated from a Bayesian approach selecting the model with the largest posterior probability.

$$BIC = -2\text{logL} + \log(N)d$$

Under the Gaussian model and squared error loss,

$$BIC(\lambda) = \frac{N}{\hat{\sigma}_e^2}\left(\overline{\text{err}}(\lambda) + \log(N)\frac{d(\lambda)}{N}\hat{\sigma}_e^2\right).$$

Again, select the model that has the lowest value.

AIC and BIC

For a given set of models (including the true model).

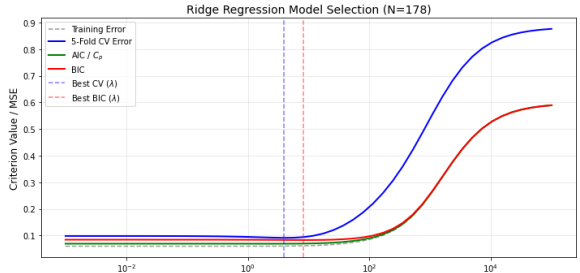
- As $n \to \infty$ the probability that BIC selects the correct model approaches one, whereas AIC tends to select a model which is too complex.
- For small sample sizes BIC tends to select a model which is too simple.

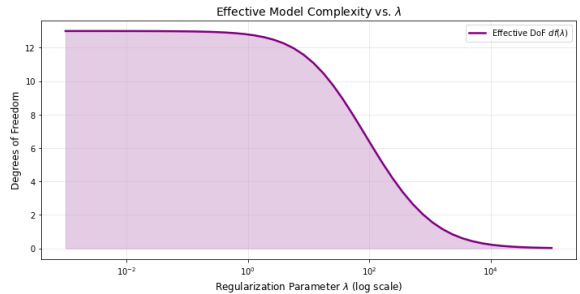
Relationship between AIC and CV

- Stone(1977) showed that AIC and leave-one-out cross-validation are asymptotically equivalent.
- Often leave-one-out cross-validation and AIC tend to choose models which are too complex.

Stone M. (1977) An asymptotic equivalence of choice of models by cross-validation and Akaike's criterion. Journal of the Royal Statistical Society Series B. 39, 44-7.

Example: CV, AIC, BIC for ridge regression on wine dataset





Model assessment

- Nested cross validation
- Bootstrapping

The Selection-Induced Bias

- **Scenario:** You test 100 different λ values using 10-fold CV.
- You pick the λ with the **absolute minimum** CV error.
- **Question:** Is that minimum error an unbiased estimate of future performance?

The Problem: Optimization Overfitting

By picking the best λ based on the validation folds, you have "spent" the independence of that data. The resulting error is **optimistically biased**.

Insight: We didn't just fit the model; we fitted the hyperparameter to the noise in the CV folds.

The Solution: Nested Cross-Validation

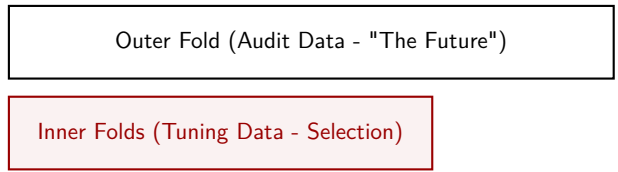
To get an honest audit, we must separate **Model Selection** from **Model Assessment**.

Inner Loop: Selection

- Used to tune λ.
- Finds the 'best' configuration for a specific training set.

Outer Loop: Assessment

- Used to audit the **entire procedure**.
- Estimates how well the 'Selection + Training' pipeline generalizes.



The Nested Mechanism

1. **Outer Split:** Split data into K_{outer} folds.
2. **For each Outer Fold j (Test):**
   - Take the remaining data as the 'Training Set.'
   - **Inner Loop:** Perform K_{inner}-fold CV on this set to find the best λ^*.
   - Train the final model with λ^* on the **entire** Training Set.
   - Evaluate on the held-out Outer Fold j.
3. **Final Report:** Average the K_{outer} scores.

Computational Cost

Total fits $= K_{outer} \times (K_{inner} \times N_{\text{lambdas}} + 1)$.
Example: $10 \times 10 \times 100 = 10,000$ *model fits. Is it worth it?*

Group exercise: The Nested Detective (10 mins)

The Case File:
An AI researcher for a surgical robot uses Nested CV.

- Outer Loop (K=5): General Error = 12%.
- In each outer fold, the "Best λ" chosen by the inner loop was different.

Group Discussion (Vevox):

1. Does the fact that λ changed mean the audit failed?
2. Which λ do you use for the final robot deployed in the hospital?
3. If the Outer Error (12%) is much higher than the Inner Error (5%), what does this tell the Auditor?

The Auditor's Insight

Tuning vs. Assessment
Nested CV audits the **methodology**, not a specific single model.

The Bootstrap

Bootstrap is a general method for **assessing statistical accuracy**.

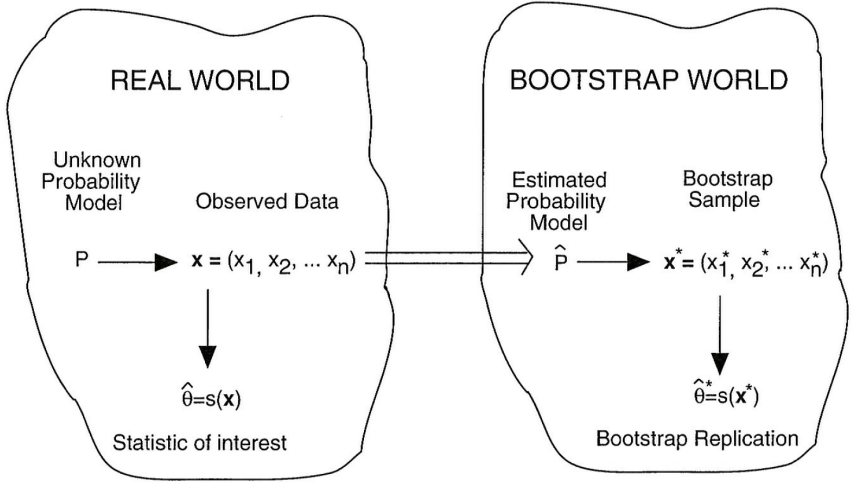
- Term from the story of Baron von Munchhausen 'Pulling oneself up by the bootstraps' (though in fact it was by the hair)
- 'Pull oneself over a fence by one's bootstraps', to mean an absurdly impossible action.
- Bootstrap estimates can be thought of as Monte-Carlo estimates.

The statistical bootstrap is also an absurdly simple and effective way of solving apparently hard problems.

Understanding bootstrap

David Freedman's terminology

We do not know P, therefore are we operating in the world where $\hat{P}$ is the truth. Use it as a mirror copy of the real world!



Bootstrap Method

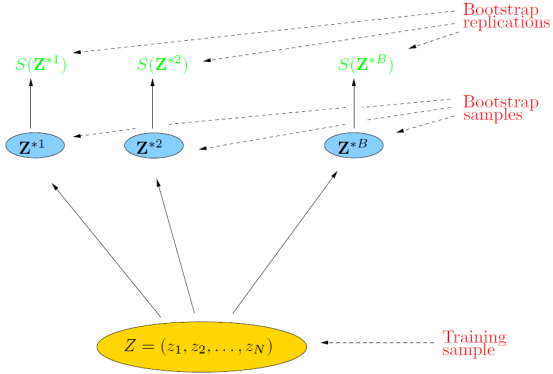
1. Given a training set $Z = (z_1, z_2, ..., z_N)$ where $z_i = (x_i, y_i)$, the basic idea is to randomly draw data sets with replacement from the training data, each sample the same size as the original training set.
2. This is done B times ($B = 100$ say), producing B bootstrap data sets. Then refit the model to each of the bootstrap data sets, and examine the behavior of the fits over B replications.

Instead of repeating the whole data collection 100 times we just draw observations with replacement from our training data and use them as if they were new experiments.

Example

$S(Z)$ is any quantity (read 'parameter') computed from Z.

The variance of a parameter, S, is estimated by taking the variance over B bootstrap replications (the value of the parameter for the specific bootstrap samples).



$$\widehat{\mathsf{Var}}[S(Z)] = \frac{1}{B-1}\sum_{b=1}^{B}(S(Z^{*b}) - \bar{S}^*)^2, \quad \bar{S}^* = \frac{\sum_b S(Z^{*b})}{B}$$

Remarks

1. How many bootstrap replicates do we need?
   - For standard deviation, a couple of hundreds.
   - For confidence intervalls, 1000 - 2000.

   Try different numbers and see if it affects the result.
2. Bootstrap does not work so well for 'tail statistics'. It works better for 'in the middle' of data.
3. Tibshirani: Do not use bootstrap for model selection. It is intended for other problems.

Exercise: Bootstrap (10 mins)

The Evidence:
You ran a Bootstrap ($B = 2000$) to assess the reliability of a feature (Age) in your model.

- **Mean coefficient:** 0.5
- **95% Confidence Interval:** $[-0.1, 1.1]$

Group Task:
As a clinical auditor, do you include this feature in the final medical report? Why or why not?

Vevox Poll Open

Your Verdict?
Trust, Reject, or Collect More Data?

Classifier performance

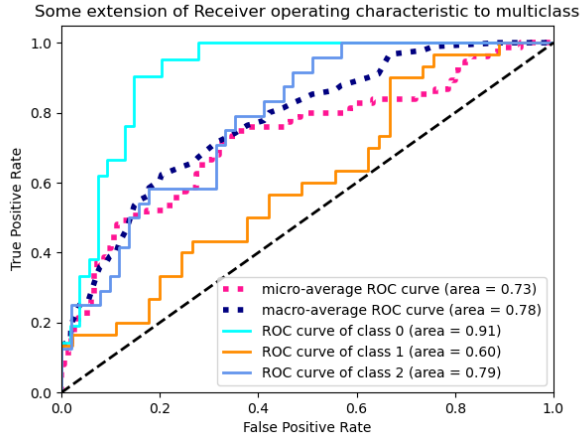
- Confusion Matrix
- ROC curves

Confusion matrix

		Predicted condition			
	Total population $= P + N$	**Predicted positive**	**Predicted negative**	Informedness, bookmaker informedness (BM) $= TPR + TNR - 1$	Prevalence threshold (PT) $= \frac{\sqrt{TPR \times FPR} - FPR}{TPR - FPR}$
Actual condition	**Real Positive (P)** [a]	**True positive** (TP), hit[b]	**False negative** (FN), miss, underestimation	True positive rate (TPR), recall, sensitivity (SEN), probability of detection, hit rate, power $= \frac{TP}{P} = 1 - FNR$	False negative rate (FNR), miss rate type II error [c] $= \frac{FN}{P} = 1 - TPR$
	Real Negative (N)[d]	**False positive** (FP), false alarm, overestimation	**True negative** (TN), correct rejection[e]	False positive rate (FPR), probability of false alarm, fall-out type I error [f] $= \frac{FP}{N} = 1 - TNR$	True negative rate (TNR), specificity (SPC), selectivity $= \frac{TN}{N} = 1 - FPR$
	Prevalence $= \frac{P}{P + N}$	Positive predictive value (PPV), precision $= \frac{TP}{TP + FP} = 1 - FDR$	False omission rate (FOR) $= \frac{FN}{TN + FN} = 1 - NPV$	Positive likelihood ratio (LR+) $= \frac{TPR}{FPR}$	Negative likelihood ratio (LR−) $= \frac{FNR}{TNR}$
	Accuracy (ACC) $= \frac{TP + TN}{P + N}$	False discovery rate (FDR) $= \frac{FP}{TP + FP} = 1 - PPV$	Negative predictive value (NPV) $= \frac{TN}{TN + FN} = 1 - FOR$	Markedness (MK), deltaP (Δp) $= PPV + NPV - 1$	Diagnostic odds ratio (DOR) $= \frac{LR+}{LR-}$
	Balanced accuracy (BA) $= \frac{TPR + TNR}{2}$	F_1 score $= \frac{2\,PPV \times TPR}{PPV + TPR} = \frac{2\,TP}{2\,TP + FP + FN}$	Fowlkes–Mallows index (FM) $= \sqrt{PPV \times TPR}$	*phi* or Matthews correlation coefficient (MCC) $= \sqrt{TPR \times TNR \times PPV \times NPV} - \sqrt{FNR \times FPR \times FOR \times FDR}$	Threat score (TS), critical success index (CSI), Jaccard index $= \frac{TP}{TP + FN + FP}$

Sources: [12][13][14][15][16][17][18][19] view • talk • edit

ROC - receiver operating characteristics curve



Recap: The Performance Scorecard

Classification Audit

Accuracy: Total correct. *Dangerous for imbalanced data.*

Sensitivity: Recall of the positive class. (e.g: detecting cracks).

Specificity: Recall of the negative class. (e.g: avoiding false diagnosis).

AUC-ROC: General performance across all classification thresholds.

Note

If prevalence is low (e.g: 0.1%), ignore Accuracy. Use **Precision-Recall** curves.

Regression Audit

MSE/RMSE: Outlier sensitive. Useful for safety-critical audits. $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

MAE: Direct physical interpretation. Robust to outliers. $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$

R^2: Fraction of variance explained. *Relative, not absolute.*

Residuals: The final 'sanity check' for structural patterns.

Note

If the **Residual Plot** shows a pattern, the model is incomplete regardless of MSE.