582631 — 5 credits

Introduction to Machine Learning

Lecturer: Teemu Roos
Assistant: Ville Hyvönen

Department of Computer Science
University of Helsinki

(based in part on material by Patrik Hoyer and Jyrki Kivinen)

November 1st–December 16th 2016

Lecture 13:
Resampling and Ensemble Methods

December 16, 2016

Performance and Generalisation

- A fundamental issue in machine learning is that we build models based on training data, but really care about performance on new unseen test data
- *Generalisation* refers to the learned model's ability to work well also on unseen data
  - good generalisation: what we learned from training data also applies to test data
  - poor generalisation: what seemed to work well on training data is not so good on test data

Resampling: Not Only Supervised Learning

- So far, we've considered supervised learning: learning to predict Y given X
  - resampling (cross-validation) can be used to obtain a number of train–test splits
  - averaging reduces variance of the test error estimate
- However, we can apply the same ideas for estimating any parameter (accuracy, coefficient, probability)
- For example:
  - estimate the variance of the least squares estimate of a regression coefficient (see the Lab in the textbook, pp. 195–197)
  - obtain confidence interval of the median of a variable (see the additional material on bootstrap confidence intervals on the course homepage)
  - combine different estimates, such as predictions or even hierarchical clustering solutions, etc.

Performance and Generalisation (2)

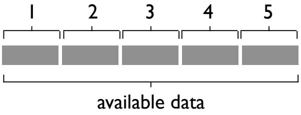
Important to notice:

- The test error rate (on a test/validation set that is separate from the training set) is a valid estimator of the error rate
- The purpose of cross-validation is just to obtain multiple train–test splits
- Hence, resampling is not *necessary* to estimate performance – it simply helps to improve estimation accuracy!

Cross-validation

Recall that cross-validation gives us K (e.g., $K = 10$) train–test splits

1. Divide the data into K equal-sized subsets:



2. For j goes from 1 to K:
   2.1 Train the model(s) using all data except that of subset j
   2.2 Compute the resulting validation error on the subset j

3. Average the K results

When $K = N$ (i.e. each datapoint is a separate subset) this is known as *leave-one-out* cross-validation.

Bootstrap

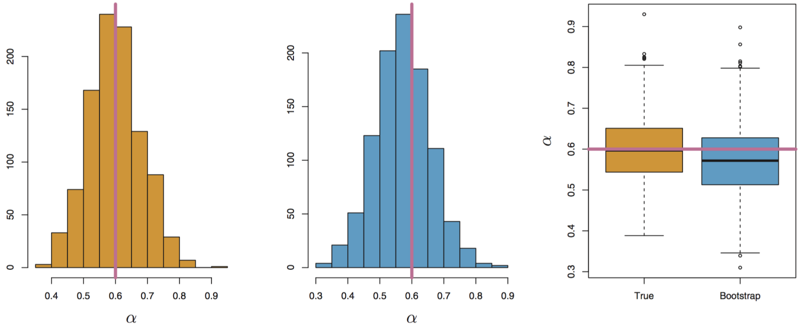
- Another popular resampling method is *bootstrap*
- The idea is to reuse the "training" set to obtain multiple data sets
- Not restricted to supervised learning (hence "training")
- These **bootstrap samples** can be used to estimate the variability of an estimate of parameter θ
- Bootstrap:
  1. Let $D = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ be the actual data
  2. Repeat $j = 1, \ldots, K$ times:
     2.1 Create D_j^* by drawing n objects from D *with replacement*
     2.2 Obtain estimate $\hat{\theta}_j^*$ from D_j^*
  3. Use the bootstrap estimates $\hat{\theta}_1^*, \ldots, \hat{\theta}_K^*$ to estimate variability

Bootstrap (2)

- Let F be the true underlying distribution
- Denote by F^* the empirical distribution corresponding to the actual data D
  - For example, if $D = (a, a, b, a)$, then $F^*(a) = 0.75$ and $F^*(b) = 0.25$
- The bootstrap samples D_j^* are drawn from F^*
- The bootstrap principle (assumption):
  - The empirical distribution F^* is a good approximation of the true distribution F
  - The bootstrap distribution of the estimator $\hat{\theta}^*$ is a good approximation of the sampling distribution of $\hat{\theta}$
- The bootstrap principle implies that we can treat $D_1^*, \ldots, D_K^*$ as K replicates from the same distribution as D

Bootstrap (3)

Example (p. 189 in the textbook):



Source: (James et al., 2013)

Left: Histogram of estimates from 1000 simulated data sets (from F)

Right: Histogram of estimates from 1000 bootstrap samples (from F^*)

Ensemble Method for Supervised Learning

- Having several training samples, $D_1, \dots, D_K$, would clearly be nice also for supervised learning
- We can combine the learned models $\hat{f}_1, \dots, \hat{f}_K$ into an aggregate model $\hat{f}_{\mathrm{agg}}$
- The aggregate model will have lower variance than the individual models
- If a learning method has high variance (but low bias), then $\hat{f}_{\mathrm{agg}}$ may be a very good model
- **Bagging** = bootstrap aggregation: $\hat{f}_j = \hat{f}_j^*$ obtained from bootstrap (see textbook Sec. 8.2.1)

Bagging

1. Bootstrap to obtain $D_1^*, \ldots, D_K^*$
2. Learn models (classifiers or regression models) $\hat{f}_1^*, \ldots, \hat{f}_K^*$ from the bootstrap samples
   - for example, unpruned regression/decision trees (high variance, low bias)

3a. For regression, combine by averaging:

$$\hat{f}_{\mathrm{bag}}(\mathbf{x}) = \frac{1}{K} \sum_{j=1}^{K} \hat{f}_j^*(\mathbf{x})$$

3b. For classification, combine by voting:

$$\hat{f}_{\mathrm{bag}}(\mathbf{x}) = \mathrm{majority}\left(\hat{f}_1^*(\mathbf{x}), \ldots, \hat{f}_K^*(\mathbf{x})\right)$$

Random Forests

- For decision trees, bagging tends to improve somewhat
- However, the trees are highly dependent in cases where the splits that maximize the gain are clearly better than the 2nd best splits
  - for example, always split according to X_3 first, then X_4, etc.
- The trees can be forced to use different features by only allowing splits based on a *random sample of the features*
- For example, only consider about $\sqrt{p}$ of all features at each split: the feature of with maximum gain is usually outside this set
- Trees constructed in bagging or random forests are usually not pruned since averaging a large number of trees reduces overfitting

Stacking and Boosting (not required for the exam)

The bagging approach is to combine the "base-learners" by averaging or voting. We can usually do better by either

- **stacking**: the idea in stacking is to apply a meta-level machine learning algorithm to learn how to best combine the base-learners
- **boosting**: boosting is an iterative method where new learning problems are constructed based on the errors made by the earlier solutions

Summary of Resampling and Ensemble Methods

Resampling methods

- Boosting and other resampling methods are generic statistical techniques that reuse the data to simulate repeated sampling
- Bootstrap can be used for various statistical estimation tasks such as obtaining confidence intervals

Ensemble Methods

- In supervised machine learning, ensemble methods build multiple hypotheses from multiple training sets obtained by resampling
- Examples:
  - cross-validation
  - bagging
  - random forests (a specific variant of bagging for decision trees)
  - stacking, boosting, ...