Evaluation

Alipio Jorge (DCC-FCUP)

November 2021

Model Evaluation

The credit office of the bank now has to decide whether to give the loan or not to a specific client. Management feels that the current credit decision procedure is not efficient and that the bank can make more money and secure more good clients with a better process The bank has an historical record of loans. Some were conceded and went well. Other were not conceded or had a bad outcome.

- What is the **machine learning goal**?
  - To obtain a **classification model** for predicting whether the loan is going to be payed or not.
- Which are the **success criteria**?
  - **Machine Learning**: Is the model predicting correct classes?
  - **Business**: Is the model increasing profit?

Model Evaluation: estimating real performance

- We want to **estimate** how well the model performs with **unknown** cases
  - makes good predictions (we do not know the future)
  - makes good diagnoses (we do not know them yet)
  - assigns correctly an email to a folder (not previously assigned)

Model Evaluation: model selection

- We have a classification problem
  - Is Naive Bayes **better** than KNN or Decision Trees?
- **Model Selection**
  - Obtain a model with each one (on the same data)
  - Identify the model that performs better (on the same data)

Model Evaluation: hyperparameter tuning

- What should the number of neighbours be for kNN?
- **Compare** 1NN, 3NN, 5NN and 15NN
    - this is also model selection

Holdout

- **Holdout** evaluation method
  - separate data set in **train** and **test**
  - use train to **learn** the model, and test to **assess** it
  - why **separate**?

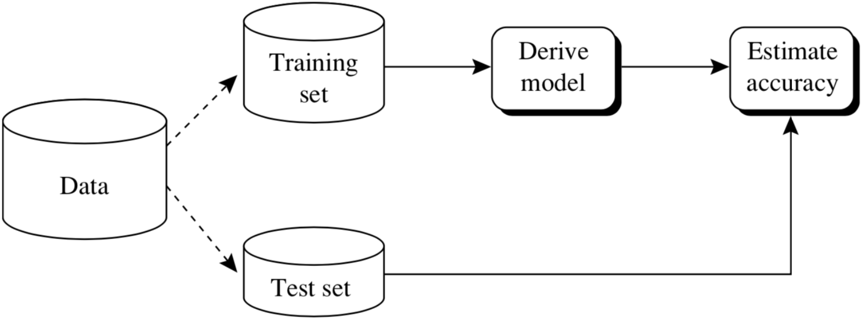


Figure 1: fig from han et al.

Metrics for Evaluating Classifier Performance

- How to **assess** a classifier?
    - many metrics
- Popular ones
    - Accuracy
    - Recall
    - Precision
    - F1
    - Sensitivity
    - Specificity

Accuracy

- **Accuracy**
  - the test asks *Total* questions
  - each question is equally important
  - the model gets *Right* questions right
  - the proportion of right answers

$$Accuracy = \frac{Right}{Total}$$

- **Error**
  - $Error = 1 - Accuracy$

The confusion matrix

- My credit decision model has an accuracy of 64% (or 0.64, we can say either way)
- How good is it on each class?
  - No idea, accuracy **amalgamates** all the classes
- The **Confusion Matrix**
  - where are the errors?
  - we can see that 24 loans are wrongly given
  - and 12 are wrongly denied

classified as $\rightarrow$	loan	no loan
loan	43	12
no loan	24	21

Binary classification

- When we are learning a **concept** of interest
  - e.g. a good credit client, the presence of a disease
- We have
  - **positive examples**
  - **negative examples**
- Depending on how a model classifies a test case

classified as	loan	no loan	
loan	**True Positives**	**False Negatives**	Positives
no loan	**False Positives**	**True Negatives**	Negatives

Redefining Accuracy

$$Accuracy = \frac{TP + TN}{P + N}$$

- In other words
  - The main diagonal divided by the sum of all the matrix

Other interesting measures: Recall

- Are we missing good clients?
  - If $Recall = 1$ we get them all

$$Recall = \frac{TP}{P}$$

- A.k.a.
  - **True Positive Rate**
  - **Sensitivity**
    - How sensitive is the model to the positive class?

Other interesting measures: Precision

- Are all our decisions good?
  - If $Precision = 1$ we only say right things

$$Precision = \frac{TP}{TP + FP}$$

- **Note the following**
  - Management wants to get more clients and asks for a model with higher **recall**.
  - Data scientists say: "we risk getting more bad clients too"
  - When Recall increases Precision **tends** to go down
  - and vice-versa

Other interesting measures: F_1

- How to combine recall and precision?
  - calculate the **harmonic mean**

$$F_1 = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

- F_1 is
  - also known as **F-score**
  - low if **either** recall or precision are low
  - equal to Recall and Precision if $Recall = Precision$
  - generalised by F_β

Other interesting measures: Specificity

- Are we excluding all the bad clients?
  - If *Specificity* $= 1$ we identify all bad clients (and hopefully some good ones)

$$Specificity = \frac{TN}{N}$$

- A.k.a.
  - **True Negative Rate**
  - Used in medical applications
    - negatives are patients without the disease

An example

- The bank wants a model to identify good clients...

classified as $\rightarrow$	loan	no loan
loan	43	12
no loan	32	21

- **Accuracy** is $(43 + 21)/(32 + 12 + 43 + 21) = 0.59$
- **Recall** ('loan' as positive) is $43/(43 + 12) = 0.78$
  - the bank identifies 78% of the good clients, but 22% are missed
- **Precision** is $43/(43 + 32) = 0.57$
  - 57% of the loans given would fail
- **Specificity** is $21/(21 + 32) = 0.40$
  - the bank only detects 40% of the 'bad' clients
- **F1** is $2 \times 0.78 \times 0.57/(0.78 + 0.57) = 0.66$
  - there is a relatively good balance of recall and precision

An extreme example

The bank has a model for detecting fraud in the use of their credit card. The confusion matrix of the model is the following

classified as	fraud	clean	
fraud	**16**	**8**	Positives
clean	**300**	**1200**	Negatives

- accuracy $= 0.80$
  - this looks good. **Is it?**
- what is the accuracy of the **majority class rule**?
  - 0.98 (higher than the model)
- recall $= 0.67$
  - not so bad to get 67% of the fraud cases
- precision $= 0.05$
  - many 'innocents' will have their card blocked
- **Conclusion**: it depends on the **cost** of missing a fraud and annoying a clean client (check later for **cost matrices**)

Estimating evaluation metrics

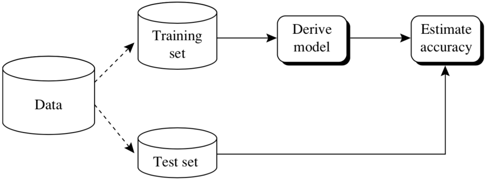
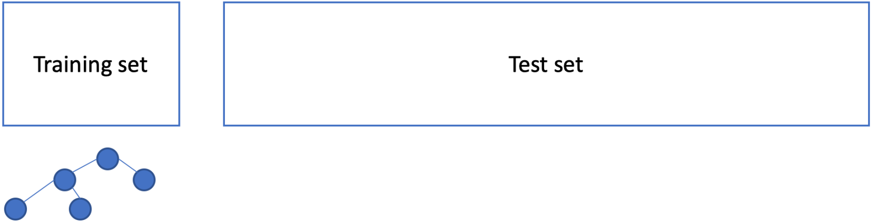


Figure 2: fig from han et al.

- **Option 1**: Estimate accuracy on the training examples
  - Estimate tends to be **optimistic**
  - Very bad choice
- **Option 2**: Isolate a subset for testing (**holdout**)
  - Estimate is more **reliable**
  - Estimate tends to be **pessimistic**
  - Depends on **sampling**
  - Depends on the **sizes** of the train and test set

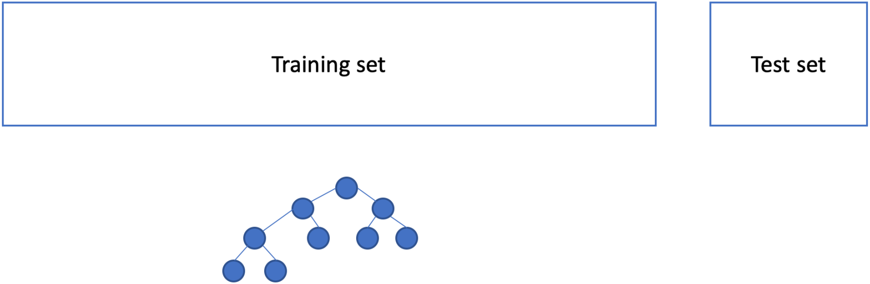
Holdout

- **Option2.1**: Small training set, large test set
  - Model will be **less robust** than if more data had been used
  - Testing is more representative



Holdout

- **Option2.2**: Large training set, small test set
  - Model is **more robust** (closer to training with the whole data)
  - Testing may be less representative (we have to make sure it is)



Holdout

- **Given** a dataset D
  - **Randomly** split D int **Train** and **Test**
    - e.g.: 80% - 20%
  - **Learn the model** from Train
  - **Assess the model** with Test
- The train and test sets are **independent**
- Different samples may produce **different** results (variance)
- If D is small, then Holdout produces unreliable estimates

Example with a data set

- Using the iris data set with 3NN
- 10-90 split, different runs
  - 0.925, 0.918, 0.829, 0.985, 0.948, 0.896, 0.896, 0.962
- 90-10 split, different runs
  - 1.000, 1.000, 0.933, 1.000, 0.867, 0.933, 0.867, 0.933

Dealing with assessment variance

(*note: I will mostly assume we are estimating accuracy but the concepts are applicable to other metrics*)

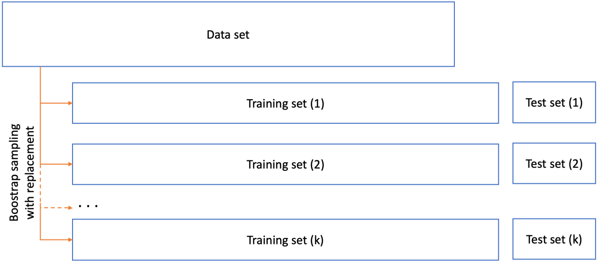
- A problem with holdout is **variance** of the evaluation estimates
  - Accuracy is also a **random variable**
  - We do not know its **true value** or **distribution**
- Solutions:
  - Testing with **more data**
    - reduces variance
    - if there is more data
  - **Repeating** the train-test cycle
    - reduces variance
    - enables **studying the distribution** of the measure (e.g. accuracy)
    - but we need a different sample for each iteration

Repeating train-test: Random subsampling

- **Random Subsampling**
  - k train-test subsamples from the **same** data
  - obtain an accuracy estimate for each subsample
  - calculate average and variance of the k estimates

Repeating train-test: Bootstrapping

- **Bootstrapping** samples the data set with replacement
  - k bootstrap subsamples from the **same** data
    - same size as data, can have **repeated examples**
  - k test sets with the examples left out in each bootstrap
    - on average 63.2% of the data set
  - obtain an accuracy estimate for each subsample
  - calculate average and variance of the k estimates



Subsampling methods: analysis

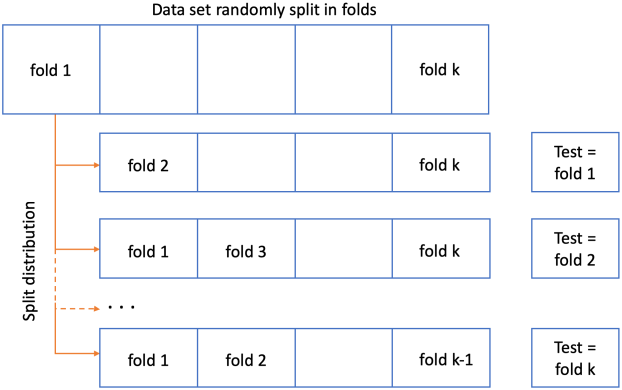
- Bootstrapping and Random subsampling
  - Samples lack **independence**
  - Exploit **small data**
  - Accuracy estimators are still **not very robust**
  - Better than simple holdout
- Which one is **preferable**?
  - not very clear
  - bootstrapping promotes variance in the model
    - under represented example categories may be **amplified**
    - e.g. 5 vintage mansions in the house market

Repeating train-test: k-fold Cross-validation

- Two random test sets may share examples
  - not ideal for test independence
- **Cross-validation**
  - Split the data in k folds of the same size
  - Use each fold as a separate test set
    - no shared examples

Repeating train-test: k-fold Cross-validation

- In each iteration $i \in 1 \ldots, k$
  - use fold i for testing
  - use the other $k - 1$ folds for training
  - obtain Acc_i
- Study the distribution of the Acc_i



Repeating train-test: k-fold Cross-validation

- Each sample is used **exactly once**
- Test sets are **independent**
  - Training sets are not
- What if we have a **small class**?
  - **stratified cross validation** to avoid under representation
- If we have **very few examples**?
  - **leave one out** cross validation
  - also leave p out for other (small) values of p
- **How many** folds should we use?
  - typical: 10 fold cross validation (10 fold CV)
  - (Demsar 2006) 2 times 5 fold CV
- **Shufling** before splitting may be a good idea

Validation, Test and deployment

- We can use cross-validation for **tuning**
  - hyperparamater tuning, pre-processing decisions, ...
- In after tuning (and other decisions) use a **new test set**
- The **internal test set** is the **validation set** (or sets)

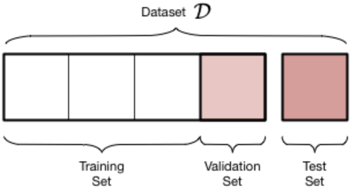


Figure 3: fig from mc.ai

Validation, Test and deployment

- **Why** do we need **validation and test ?**
  - using the test set to improve results leads to overly **optimistic** results
    - it is like **using the future to make predictions**
    - or **knowing the exam questions** when you study
    - but we do it in the lab as long as comparisons are fair
  - the test set should be for **testing only**
- Which model we use in **deployment** ?
  - We use the approach and hyperparameters that had best test results
  - We can **then** use the whole data to train the model
    - if data is scarce. We can use less data too

Measuring statistical significance

- We compare two algorithms **A** and **B**
  - **A** has 0.8832 accuracy
  - **B** has 0.8845 accuracy
- Is this difference **important**?
  - **Statistical significance**
    - the difference occurs most of the time
    - how likely is it to observe this difference or larger?
  - **Usefulness**
    - st. significant does not imply useful
    - does it save more lifes with fewer secondary effects?

Measuring statistical significance

- Use **Hypothesis testing**
- 10 fold CV example with t-test
  - obtain two samples of the accuracies: Acc_A and Acc_B
  - each sample has size 10
  - calculate the means $mean_A$ and $mean_B$
  - we assume that the accuracy values follow a **t-distribution** with $k-1$ degrees of freedom
  - we can use a paired **t-test**
  - H_0 **or Null hypothesis** is that the difference of means is zero
  - the **paired t-test** checks if we can reject H_0
  - the test **assumes independence** of the samples (not true)

Measuring statistical significance: Example

- **Question**: is 1NN better than 5NN for iris classification?
- Apply 10 fold cross validation
  - Split the data set in 10 folds
  - **Acc_1NN** = [1, 0.93, 1, 0.93, 0.87, 1, 0.87, 1, 1, 1]
  - **Acc_5NN** = [1, 0.93, 1, 1, 0.87, 0.93, 0.93, 1, 1, 1]
  - **mean_1NN** = 0.96
  - **mean_5NN** = 0.966667
- Use the **paired t-test**
  - **p-value** = 0.59
  - this is the probability of observing this difference of accuracy (or higher) assuming that **H_0 is true**
    - we cannot reject H_0 for a level of significance of 5% (0.59>0.05)

Measuring statistical significance: Example

- **Question**: is 5NN better than 100NN for iris classification?
- Apply 10 fold cross validation
    - Split the data set in 10 folds
    - **mean_5NN** = 0.966667
    - **mean_100NN** = 0.67
- Use the **paired t-test**
    - **p-value** = 8.5e-9
    - the probability of observing this difference of accuracy (or higher) assuming that **H_0 is true** is **very low** (below 0.05)
        - we **reject** H_0 for a **level of significance** of 5%

More on statistical significance tests

- t-test with cross validation should be avoided
  - the independence of the values does not exist
  - it gives some information though
- if we have enough data to promote independence
  - t-test is acceptable
- To compare two algorithms on multiple datasets (e.g. 30)
  - cross-validate on each dataset
  - choose a **level of significance** (typically 1% to 5%)
  - if assumptions hold use a parametric test (**t-test**)
  - if not, use non-parametric **wilcoxon signed rank** test
- To compare many algorithms on multiple datasets
  - use **Friedman** test and post-hoc **Nemenyi** with **critical distances**
- Be **careful with multiple comparisons**
  - sometimes we have to **adjust** the p-values

Misclassification Costs

- giving a loan to a bad client has a **higher cost** than not giving a loan to a good client
- Accuracy is insensitive to cost
  - We can use a misclassification **cost matrix** combined with the confusion matrix

Conf	loan	not
loan	43	12
not	24	21

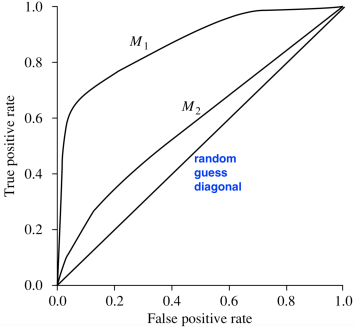
Cost1	loan	not
loan	0	1
not	10	0

Cost2	loan	not
loan	0	1
not	2	0

- With Cost1 we have $Cost = 24 \times 10 + 12 = 252$
- With Cost1 we have $Cost = 2 \times 10 + 12 = 32$

ROC curves

- We can compare the performance of classifiers on the whole spectrum of misclassification costs
- **Receiver operating characteristic curves**
  - plot relation between TPR and FPR
  - the **AUC**, area under the curve, is an assessment measure



References

- Books
  - Han, Kamber & Pei, Data Mining Concepts and Techniques, Morgan Kaufman.
  - Jake VanderPlas, Data Science Handbook, O'Reilly
- Papers
  - Janez Demsar, Statistical Comparisons of Classifiers over Multiple Data Sets, JMLR, 2006.
- Blog articles -https://towardsdatascience.com/introduction-to-na%C3%AFve-bayes-classifier-fa59e3e24aaf