

Ensemble Methods

TP 1 – An experimental protocol for supervised learning

Master MALIA

Guillaume Metzler

Institut de Communication (ICOM)
Université de Lyon, Université Lumière Lyon 2
Laboratoire ERIC UR 3083, Lyon, France

guillaume.metzler@univ-lyon2.fr

Abstract

This first practical session sets up the experimental protocol that we will use in all the following ones. On two simple families of models, Gaussian kernel SVMs and decision trees, we distinguish the empirical risk from the true risk, we see why and when the data must be normalized, we search hyperparameters with a validation set and with cross-validation, and we evaluate the final model once, on a test set kept aside. The last two parts open the programme of the course: combining many trees into a vote (ensemble methods, TP 2 and 3), and turning a test error into a *guarantee* on the true risk (generalization bounds and PAC-Bayes, TP 4 and 5).

About this session – TP 1: An experimental protocol

Goal. By the end of the session you should be able to explain the difference between empirical and true risk, decide when to normalize the features, tune hyperparameters by cross-validation without touching the test set, report a test error with its uncertainty, and say what ensemble methods and generalization bounds will bring.

To reread before the session. Sections 5 and 6.1 (risk and empirical risk minimization), Section 7.1 (how to train and tune a model), Section 8.3 (SVM and kernels), Section 8.7 (decision trees) and Sections 11.3–11.5 (preprocessing, splitting, tuning) of the lecture notes.

Tools. `numpy`, `matplotlib` and `scikit-learn` (`SVC`, `DecisionTreeClassifier`, `StandardScaler`, `make_pipeline`, `StratifiedKFold`, `cross_val_score`, `GridSearchCV`).

Data. A synthetic problem whose distribution is known (Part 1), and three datasets of the course loaded with `load(name)` of `tptools.py`, with labels in $\{-1, +1\}$: `wdbc` (breast cancer, 569×30), `pima` (diabetes, 768×8) and `australian` (credit, 690×14). The file `tp1_starter.py` contains the data generator and the imports.

Good practice Fix all the random seeds (`random_state=seed`). Never call `fit` on data that contains test examples, and put every preprocessing step in a

Pipeline so that cross-validation refits it on each training fold. Keep the test set in a variable that you use only once, at the end.

1 Empirical risk and true risk

The input X is uniform on $[-1, 1]^2$, the clean label is $c(x) = +1$ if $x_2 > \frac{1}{2} \sin(\pi x_1)$ and -1 otherwise, and the observed label Y is $c(X)$ flipped with probability $\eta = 0.1$. For a classifier h , the true risk is $R_{\mathcal{D}}(h) = \mathbb{P}(h(X) \neq Y)$ and the empirical risk on a sample $S = \{(x_i, y_i)\}_{i=1}^m$ is $\hat{R}_S(h) = \frac{1}{m} \sum_{i=1}^m \mathbb{1}[h(x_i) \neq y_i]$.

1. What is the Bayes classifier of this problem and its risk R^* ?
Check it on 50 000 examples.
2. Draw a training sample of size $m = 200$ and a sample of 50 000 examples used as a proxy for the true risk.
Plot the empirical (training) risk and the true risk of a decision tree as functions of its maximal depth (1 to 15), and of a Gaussian kernel SVM ($C = 10$) as functions of $\gamma \in [10^{-2}, 10^3]$, where the kernel is $k(x, x') = e^{-\gamma \|x - x'\|^2}$.
Identify underfitting and overfitting.
Plot the decision boundaries of the SVM for $\gamma \in \{0.01, 3, 100\}$.
3. Fix the tree of depth 5.
Draw 500 fresh samples of size $n \in \{50, 200, 1000, 5000\}$, compute its empirical risk on each, and compare their mean and standard deviation with $R_{\mathcal{D}}(h)$ and $\sqrt{R_{\mathcal{D}}(h)(1 - R_{\mathcal{D}}(h))/n}$.
Why is the empirical risk an unbiased estimate of the true risk here, but not in question 2?

2 Normalization of the data

1. Look at the standard deviations of the features of `wdbc` and `australian`.
What do you notice?
2. For 10 random splits (70%/30%), compute the test error of an SVM (`C=1`, `gamma="scale"`) and of a tree of depth 4, with and without standardization (`StandardScaler` fitted on the training set).
Explain the results.
3. Why must the scaler be fitted on the training set only?
How do you do it inside a cross-validation?

3 Hyperparameter search and cross-validation

We work on `pima`. Keep 25% of the data as test set (stratified, `random_state=0`); the remaining 75% is the learning set.

1. On the learning set, compute the 5-fold cross-validation error of the pipeline (standardization + SVM) for $C \in \{0.1, 1, 10, 100, 1000\}$ and $\gamma \in \{10^{-4}, \dots, 1\}$, and display it as a heat map.
Do the same for a tree, with `max_depth` $\in \{1, 2, 3, 4, 6, 8, \text{None}\}$ and `min_samples_leaf` $\in \{1, 5, 10, 20\}$.
Which hyperparameters do you select?
Retrain on the whole learning set and compute the test error.
2. **Hold-out or cross-validation?** Repeat the selection of (C, γ) 30 times, with different random partitions of the learning set, (a) with a single hold-out validation split (75%/25% of the learning set), (b) with 5-fold cross-validation.
For each procedure, count the number of distinct pairs selected, and compute the mean and standard deviation of the selected validation score and of the test error of the selected model.
Conclude.

4 The complete protocol, and the optimism of the best score

1. Write a function that, given a dataset and a seed, (i) splits it into a learning set (75%) and a test set, (ii) selects the hyperparameters of the SVM pipeline and of the tree by 5-fold cross-validation on the learning set (`GridSearchCV`), (iii) retrains them on the whole learning set and (iv) computes their test errors.
Run it on `wdbc`, `pima` and `australian` for 10 seeds, and compare the best cross-validation error with the test error.
2. To understand the difference, replace the labels of `pima` by random labels (independent of the features) and run the same protocol for the SVM.
What is the true risk of any classifier?
What do you observe?

5 Opening: what if we combine many trees?

1. On the synthetic problem ($m = 300$) and on `pima` (70%/30%), learn $B = 100$ fully grown trees, each one on a *bootstrap sample* of the training set (m indices drawn with replacement), and predict by the majority vote sign $(\sum_b h_b(x))$.
Compare the test error of the vote with that of one fully grown tree learned on the whole training set, and with the mean test error of the 100 trees.
2. Why can a vote be much better than its voters?

6 Opening: a first guarantee on the true risk

Let h be a classifier *fixed before* drawing a test sample T of n i.i.d. examples. Hoeffding's inequality gives, with probability at least $1 - \delta$ over T ,

$$R_{\mathcal{D}}(h) \leq \hat{R}_T(h) + \sqrt{\frac{\ln(1/\delta)}{2n}},$$

and a tighter version, based on the Chernoff bound, is $R_{\mathcal{D}}(h) \leq \bar{\text{kl}}^{-1}(\hat{R}_T(h), \frac{\ln(1/\delta)}{n})$, where $\bar{\text{kl}}^{-1}(q, \varepsilon) = \sup\{p \in [q, 1] : \text{kl}(q||p) \leq \varepsilon\}$ and $\text{kl}(q||p) = q \ln \frac{q}{p} + (1 - q) \ln \frac{1-q}{1-p}$ (`kl_inv_upper` in `common/tptools.py`). We take $\delta = 0.05$.

1. Prove the Hoeffding bound from $\mathbb{P}(R_{\mathcal{D}}(h) - \hat{R}_T(h) \geq \varepsilon) \leq e^{-2n\varepsilon^2}$.
2. Compute both bounds for the SVM selected in Part 4 (seed 0) on the three datasets.
Interpret them.
How many test examples would be needed for a Hoeffding term of 0.01?
3. On the synthetic problem, fix the SVM with $\gamma = 3$ learned on 200 examples, draw 1000 test sets of size $n \in \{50, 200, 1000\}$ and check that both bounds hold in at least 95% of the cases.
4. **Selection.** Why can the bound not be applied to the best cross-validation score of Part 4?
Simulate K useless classifiers (true risk $\frac{1}{2}$) whose errors on a common validation set of $n = 100$ examples are independent $\mathcal{B}(n, \frac{1}{2})/n$, select the best one, and compute how often the Hoeffding bound holds for it, for $K \in \{1, 25, 1000\}$.
Show that replacing δ by δ/K (union bound) repairs it.

7 Going further (★)

1. Prove that the Bayes classifier $h^*(x) = \text{sign}(2\eta(x) - 1)$ (either label when $\eta(x) = \frac{1}{2}$), with $\eta(x) = \mathbb{P}(Y = 1 \mid X = x)$, minimizes the true risk, and that $R^* = \mathbb{E}[\min(\eta(X), 1 - \eta(X))]$.
2. **Data leakage.** Draw $m = 100$ examples with $d = 2000$ Gaussian features and labels drawn uniformly in $\{-1, +1\}$, independently of the features.
 - (a) Select the 20 features most correlated with the label on the whole dataset, then estimate the accuracy of a logistic regression on these features by 5-fold cross-validation.
 - (b) Do the same, but perform the selection inside each fold, on the training part only.
Repeat 10 times and compare with the true accuracy.