Guidelines for Machine Learning Experiments

Vignesh L S

A. Aim of the Study

- We need to start by stating the problem clearly, defining what the objectives are. In machine learning, there may **be several possibilities.**
- As we discussed before, we may be interested in assessing the expected error (or some other response measure) of a learning algorithm on a particular problem and check that, for example, **the error is lower than a certain acceptable level**.
- These can be two different algorithms, or one can be a proposed improvement of the other, for example, **by using a better feature extractor.**
- In the general case, we may have more than two learning algorithms, and we may want **to choose the one with the least error, or order them in terms of error, for a given dataset**.

B. Selection of the Response Variable

- We need to decide on what we should use as the quality measure.
- Most frequently, error is used that is the misclassification error for classification and mean square error for regression.
- We may also use some variant; for example, generalizing from 0/1 to an arbitrary loss, we may use a risk measure.
- In information retrieval, we use measures such as precision and recall. In a cost-sensitive setting, not only the output but also system parameters.

C. Choice of Factors and Levels

- If we fix an algorithm and want to find the best hyper parameters, then those are the factors.
- If we are comparing algorithms, the learning algorithm is a factor. If we have different datasets, they also become a factor. The levels of a factor should be carefully chosen so as not to miss a good configuration and avoid doing unnecessary experimentation.
- For example, in optimizing k of k-nearest neighbor, one can try values such as 1, 3, 5, and so on, but in optimizing the spread h of Parzen windows, we should not try absolute values such as 1.0, 2.0, and so on, because that depends on the scale of the input; it is better to find some statistic that is an indicator of scale
- For example, the average distance between an instance and its nearest neighbor—and try h as different multiples of that statistic.

D. Choice of Experimental Design

- It is always better to do a factorial design unless we are sure that the factors do not interact, because mostly they do. Replication number depends on the dataset size; it can be kept small when the dataset is large; we will discuss this in the next section when we talk about resampling.
- However, too few replicates generate few data and this will make comparing distributions difficult; in the particular case of parametric tests, the assumptions of Gaussianity may not be tenable.
- Generally, given some dataset, we leave some part as the test set and use the rest for training and validation, probably many times by resampling. How this division is done is important. In practice, using small datasets leads to responses with high variance, and the differences will not be significant and results will not be conclusive.It is also important to avoid as much as possible toy, synthetic data
- and use datasets that are collected from real-world under real-life circumstances. Didactic one- or two-dimensional datasets may help provide intuition, but the behavior of the algorithms may be completely different in high-dimensional spaces.

E. Performing the Experiment

- Before running a large factorial experiment with many factors and levels, it is best if one does a few trial runs for some random settings to check that all is as expected.
- In a large experiment, it is always a good idea to save intermediate results (or seeds of the random number generator), so that a part of the whole experiment can be rerun when desired. All the results should be reproduceable.
- In running a large experiment with many factors and factor levels, one should be aware of the possible negative effects of software aging.
- It is important that an experimenter be unbiased during experimentation. In comparing one’s favorite algorithm with a competitor, both should be investigated equally diligently.
- In large-scale studies, it may even be envisaged that testers be different from developers. One should avoid the temptation to write one’s own “library” and instead, as much as possible, use code from reliable sources; such code would have been better tested and optimized.
- As in any software development study, the advantages of good documentation cannot be underestimated, especially when working in groups.
- All the methods developed for high-quality software engineering should also be used in machine learning experiments.

F. Statistical Analysis of the Data

- This corresponds to analyzing data in a way so that whatever conclusion we get is not subjective or due to chance. We cast the questions that we want to answer in the framework of hypothesis testing and check whether the sample supports the hypothesis.
- For example, the question "Is A a more accurate algorithm than B?" becomes the hypothesis "Can we say that the average error of learners trained by A is significantly lower than the average error of learners trained by B?"
- As always, visual analysis is helpful, and we can use histograms of error distributions, whisker-and-box plots, range plots, and so on.

G. Conclusions and Recommendations

- Once all data is collected and analyzed, we can draw objective conclusions. One frequently encountered conclusion is the need for further experimentation.
- Most statistical, and hence machine learning or data mining, studies are iterative.
- We should always remember that statistical testing never tells us if the hypothesis is correct or false, but how much the sample seems to concur with the hypothesis.
- There is always a risk that we do not have a conclusive result or that our conclusions be wrong, especially if the data is small and noisy. When our expectations are not met, it is most helpful to investigate why they are not.
- For example, in checking why our favorite algorithm A has worked awfully bad on some cases, we can get a splendid idea for some improved version of A.
- All improvements are due to the deficiencies of the previous version; finding a deficiency is but a helpful hint that there is an improvement we can make!
- But we should not go to the next step of testing the improved version before we are sure that we have completely analyzed the current data and learned all we could learn from it. Ideas are cheap, and useless unless tested, which is costly.