Artificial Intelligence (CS303)

Lecture 6: Performance Evaluation for Machine Learning

Hints for this lecture

- **Learning is more than models and training algorithms, performance evaluation is crucial.**

Outline of this lecture

- **Why?**
- **Performance Metrics**
- **Estimating the Generalization**

Why Performance Evaluation is Important?

- Suppose you are an engineer responsible for developing a computing system, say for surveillance purpose.

- What you have is a set of data provided by your client.

- Upon deliver of the system, you need to make him/her believe that you'   ve done a good job. What would you say?

Why Performance Evaluation is Important?

- A typical statement: my system has achieved a **score of 99** in terms of **XXX** (e.g., accuracy) in **5 seconds**.

- This statement sounds great but is highly risky because the environments for development and deployment might be **different**, e.g.,
    - What the client really want in his business is not consistent with XXX. (then no matter how you improve your score on XXX, things might be hopeless)
    - The data you got only reflect the reality partially. (Score of 99 does not hold in reality)
    - The hardware used to run the system might be different. (5 seconds does not hold in reality)

- Any of the above case happens → your reputation as an engineer will decrease → you might lose your job

General Remedy

- Be careful when choosing your objective function, two principles:
  - Consistent with the user requirements?
  - Existing easy-to-use algorithm to optimize it (to train the model)?

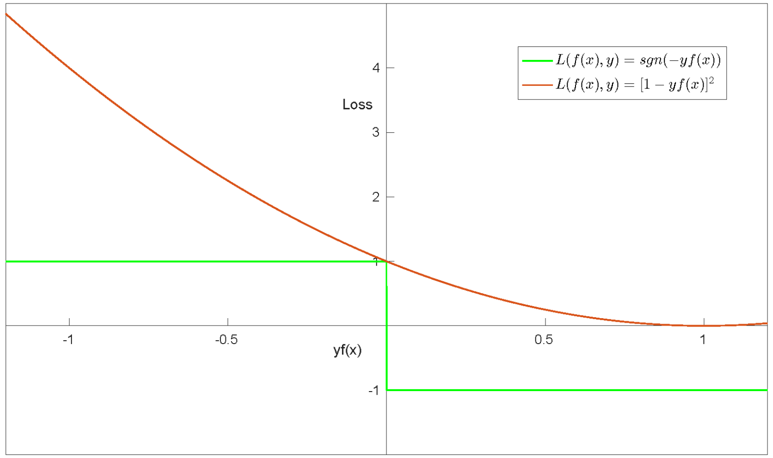
- Do internal tests as much as possible
  - estimate the generalization performance as accurate as possible.

- Can only reduce rather than remove risk. There is no guarantee in life.

I. Performance Metrics

Performance Metrics

- For practical considerations, objective function for training could be **different from** the performance metric **we truly care**.



Green: Consistent but difficult to optimize
Red: Easy to optimize but inconsistent

Instance	Label	Classifier f	Classifier g
x_1	1	$f(x_1) = -0.1$	$g(x_1) = 4$
x_2	-1	$f(x_2) = 0.1$	$g(x_2) = -2$

Performance Metrics

- **There are many performance metric (i.e., XXX), e.g., for even for binary classification**

	Predicted Positive	Predicted Negative
Positive	True Positive rate	False Negative rate
Negative	False Positive rate	True Negative rate

$$accuracy = \frac{TPR \times N^{+} + TNR \times N^{-}}{N^{+} + N^{-}}$$

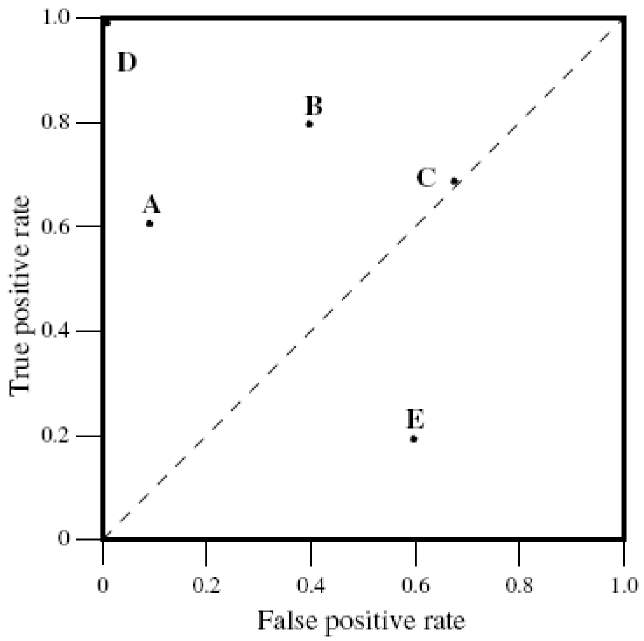
$$precision = \frac{TPR \times N^{+}}{TPR \times N^{+} + FPR \times N^{-}}$$

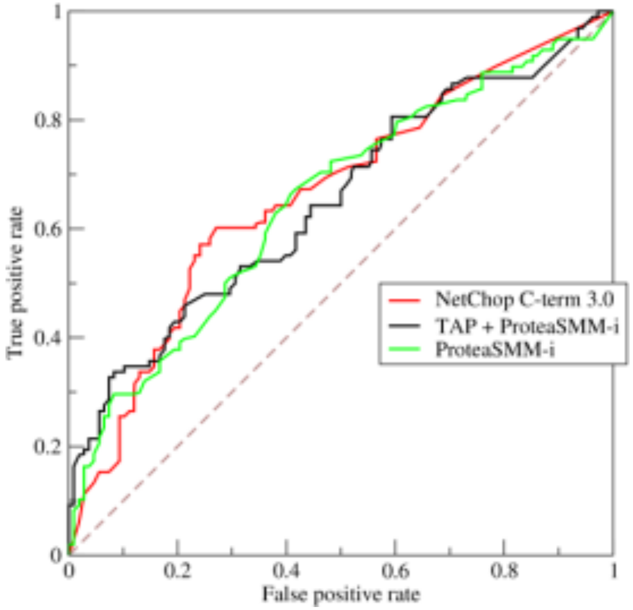
$$recall = \frac{TPR \times N^{+}}{TPR \times N^{+} + FNR \times N^{+}}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recl}$$

Performance Metrics

- Receiver Operating Characteristic (ROC) analysis (for binary classification)
  - Mapping your classifier into the ROC space
  - Tune a threshold to get a set a points, connect them to get a ROC curve.





II. Estimating the Generalization

Estimating the Generalization

- Generalization performance is a **random variable**.

- Split the data in hand into training and testing subsets.
  - Random Split
  - Cross-validation
  - Bootstrap

- Collecting the test performance for many times, calculate the average and standard deviation.

- Do statistical tests (check your textbook on statistics).

Summary

- **If there is only one lecture that you could remember about learning, this should be the one.**

To be continued