Evaluation of Machine Learning Models

Shrey Gupta

Applied Machine Learning (HOUSECS 59-01), Duke University

September 19, 2018

Evaluation

- How can we evaluate the performance of the models we create?
- Various performance measures for regression, classification, and clustering.
    - Depending on various "goals" and "priorities", different measures used.

Regression

- Metrics: mean absolute error, mean squared error, and R^2 value.

Mean Absolute Error

$$\ell(f(x), y) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

- Intuitive loss function for regression: penalize the *distance* between the predicted and actual outputs.

Mean Squared Error

$$\ell(f(x), y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- Recall: least-squares error has a closed form solution in regression, and hence is most commonly used.
- In comparison to mean absolute error, larger absolute errors are relatively penalized *more*, and smaller absolute errors *less*.

R^2 Value

$$R^2 = \frac{explained\ \ variation}{total\ \ variation} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

- "Goodness-of-fit": the greater $0 \leq R^2 \leq 1$, the stronger the model according to the data.
- Note: a low R^2 does *not* imply a poor model, but rather unexplainable variation with respect to the features.

Classification

- Metrics: misclassification error (and classification accuracy), precision, recall, F1-score, confusion matrices, and ROC curves (and associated AUC).

Misclassification Error

$$error = 1 - accuracy$$

$$accuracy = \frac{true\ \ positives + true\ \ negatives}{positives + negatives}$$

- Most commonly used metric for classification.
- Insightful when the number of positive points is approximately equal to the number of negative points.

Precision and Recall

$$precision = \frac{true\ positives}{predicted\ positives}$$

$$recall = \frac{true\ positives}{positives}$$

- Precision: "fraction of relevant instances among the retrieved instances" (Wikipedia).
- Recall: "fraction of relevant instances that have been retrieved over the total amount of relevant instances" (Wikipedia).

F1-score

$$score = 2\left(\frac{precision \times recall}{precision + recall}\right)$$

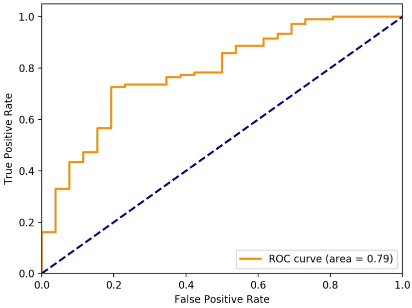
- Most applications require a balance between precision and recall (as there is a trade-off between the two).

Confusion Matrices

	y = +1	y = -1
$\hat{y}$ = +1	*true positives* (*TP*)	*false positives* (*FP*)
$\hat{y}$ = -1	*false negatives* (*FN*)	*true negatives* (*TN*)

- Provides a concise presentation of the predictive power of a model.

ROC Curves



- As we shift the classification barrier, for a given *FPR* (false positive rate), what is the respective *TPR* (true positive rate) we can achieve?

ROC Curves

- Ends of curve signify classifying *nothing* as positive and *everything* is positive.
- Note: curve must be monotonically increasing.
- *AUC* (area under curve) signifies the area under the ROC curve; larger values are preferred.
- Effectively measures the *sensitivity* of a classifier.

Clustering

- Metrics: purity, Rand measure, and F1-score.

Purity

$$\frac{1}{n} \sum_{clusters} \max_{classes} |class \in cluster|$$

- Degree to which each cluster contains a single class (calculated with labeled points).
- Note: does not work well for imbalanced data, and does not penalize having a large number of clusters.

Rand Measure

$$measure = \frac{true\ \ positives + true\ \ negatives}{positives + negatives}$$

- Similar to accuracy measure for classification, and requires labeled points.

F1-score

$$score = 2\left(\frac{precision \times recall}{precision + recall}\right)$$

- Precision and recall are calculated with labeled points, similar to classification.
- Similar to F1-score for classification, and requires labeled points.

Training and Testing

- When training on a particular data set, we can no longer use the accuracy (or other metric) on that set as an effective evaluation.
- Solution: train on a portion of the data (perhaps 70%), and "test" (i.e. compute the evaluation metric) on the remaining portion.

Training and Testing

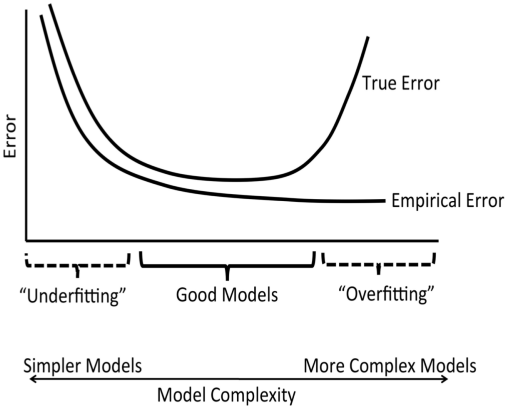


Image source: Cynthia Rudin

Training, Testing, and Validation

- A similar problem arises when tuning hyperparameters.
- Solution: create a validation set (e.g. using a 7-2-1 split).
  - Train with certain hyperparameters on the training set, evaluate on the validation set, and rotate to determine the best set of hyperparameters.
  - Finally, evaluate algorithm performance on the unused testing set.

Cross-validation

- Divide the data into k folds (e.g. a common value is $k = 10$).
- Train the algorithm on the first 8 folds, validate on the next fold, and test on the last fold.
- Rotate the folds, and repeat.
- Calculate the mean, standard deviation, and other statistics over the evaluation metric across the folds.
- Ensures data is "symmetrically" chosen.

Notebook

- Today's notebook will work through an example of cross-validation and evaluation metrics for regression and classification.