Statistical Natural Language Processing

Machine learning: evaluation

Çağrı Çöltekin

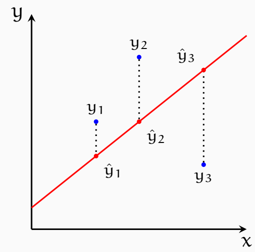
University of Tübingen
Seminar für Sprachwissenschaft

Summer Semester 2020

Measuring success/failure in regression

Root mean squared error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2}$$

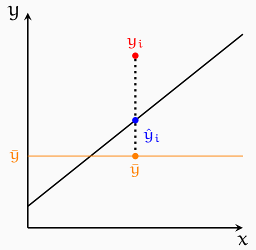


- Measures average error in the units compatible with the outcome variable

Measuring success/failure in regression

Coefficient of determination

$$R^2 = \frac{\sum_i^n(\hat{y}_i - \bar{y})^2}{\sum_i^n(y_i - \bar{y})^2} = 1 - \frac{\text{MSE}}{\sigma_y^2}$$



- r^2 is a standardized measure in range $[0, 1]$
- Indicates the ratio of variance of y explained by x
- For single predictor it is the square of the correlation coefficient r

Measuring success in classification

Accuracy, Precision, recall, F-score

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

$$F_1\text{-score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

		true value	
		positive	negative
predicted	pos.	TP	FP
	neg.	FN	TN

Measuring performance outside the training data

We want our models to perform well on unseen (test) data.

- *Overfitting* occurs when the model learns the idiosyncrasies of the training data
- *Underfitting* occurs when the model is not flexible enough for solving the problem at hand

We want simpler models, but not too simple for the task at hand.

Bias and variance

Bias of an estimate is the difference between the value being estimated, and the expected value of the estimate

$$B(\hat{\boldsymbol{w}}) = E[\hat{\boldsymbol{w}}] - \boldsymbol{w}$$

- An *unbiased* estimator has 0 bias

Variance of an estimate is, simply its variance, the value of the squared deviations from the mean estimate

$$\text{var}(\hat{\boldsymbol{w}}) = E\left[(\hat{\boldsymbol{w}} - E[\hat{\boldsymbol{w}}])^2\right]$$

$\boldsymbol{w}$ is the parameter (vector) that defines the model

Bias–variance relationship is a trade-off: models with low bias result in high variance.

Bias–variance, underfitting–overfitting

- Bias and variance are properties of estimators
- We want estimators with low bias, low variance
- Complex models tend to overfit – and exhibit high variance
- Simple models tend to have low variance, but likely to have (high) bias

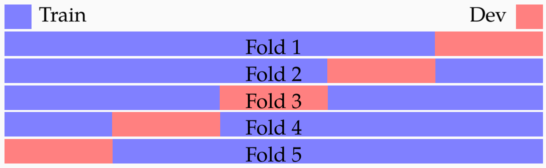
Model selection & hyperparameter tuning

- Our aim is to reduce the test error
- We can estimate the test error on a *development* set (*validation* or *held-out* data):
  - Split the data at hand as *training* and *development* set
  - Train alternative models (different hyperparameters) on the training set
  - Choose the model with best development set performance

Cross validation

- To avoid overfitting, we want to tune our models on a *development set*
- But (labeled) data is valuable
- Cross validation is a technique that uses all the data, for both training and tuning with some additional effort
- Besides tuning hyper-parameters, we may also want to get 'average' parameter estimates over multiple folds

K-fold Cross validation



- At each fold, we hold part of the data for testing, train the model with the remaining data
- Typical values for k is 5 and 10
- In *stratified* cross validation each fold contains (approximately) the same proportions of class labels.
- A special case, when k is equal to n (the number of data points) is called *leave-one-out cross validation*

The choice of k in k-fold CV

- Increasing k
  - reduces the bias: the estimates converge to true value of the measure (e.g., accuracy) in the limit
  - increases the variance: smaller held-out sets produce more varied parameter estimates
  - is generally computationally expensive
- 5- or 10-fold cross validation is common practice (and found to have a good balance between bias and variance)

Comparing with a baseline

- The performance measures are only meaningful if we have something to compare against

random: does the model do anything useful at all?
majority class: does the classifier better than predicting the majority class all the time?
state-of-the-art: how does your model compare against known (non-trivial) models?

- In comparing different models we use another split of the data, *test set*
- Ideally test set is used only once – we want to avoid tuning the system on the test data
- Differences between models are reliable only if the same test set is used
- Differences are reliable if your test set size is large enough
- Use statistical tests when comparing different models/methods

Summary

The first principle is that you must not fool yourself and you are the easiest person to fool. – Richard P. Feynman

- The measures of success in ML systems include
  - RMSE / r^2
  - Accuracy
  - Precision / recall / F-score
- We want models with low bias and low variance
- Evaluating ML system requires special care:
  - Never use your test set during training / development
  - Tuning your system on a development set
  - Cross-validation allows efficient use of labeled data

Next:

- Introduction to artificial neural networks