Model Selection and Error Estimation: The Frequentist Approach

Greg Grudic

Lecture Goals

- Defn: Model Selection
  - Given a learning set, how do you decide which algorithms build the best models, and which learning parameters are "optimal" for the best algorithm.
- Error Prediction on Future Data
  - Without knowing what the future data is…

Building Supervised Learning Models

What are the outputs and inputs to a learning algorithm?

$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$ → Learning Algorithm → $M(\mathbf{x})$

Learning Parameters → Learning Algorithm

Model is used to make predictions! $\hat{y} = M(\mathbf{x})$

Learning Parameters

- These dictate how the learning algorithm will build a model
- Changing the learning parameters changes how good the model is
- Goal: Choose the learning parameters that produce the best model

Learning Parameters

- What are the learning parameters for the linear perceptron algorithm?
  - Learning rate.
- What are the learning parameters for the SVM algorithm?
  - C (or nu), kernel choice, kernel parameter values
- What are the learning parameters for the K Nearest Neighbor algorithm?
  - K

Which Model Is Best?

- OR – which learning parameters should I use?
- The ones that produce a model that gives the best accuracy results on data that was NOT used to build the model (sometimes called future data or **test data**).
- Test data does not appear in the learning set!
- So how do I pick the model parameters if I don't know what the test data is?
- Answer: create a *fake test set* called a validation set.

Measuring Model Accuracy: Regression

- Assume a set of data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_K, y_K)$
- Regression accuracy
  - Two commonly used metrics
    - Mean Square Error

$$error_{M(\mathbf{x})} = \frac{1}{K}\sum_{i=1}^{K}\left(y_i - M(\mathbf{x}_i)\right)^2 = \frac{1}{K}\sum_{i=1}^{K}\left(y_i - \hat{y}_i\right)^2$$

    - Relative Error

$$error_{M(\mathbf{x})} = \frac{\sum_{i=1}^{K}\left(y_i - M(\mathbf{x}_i)\right)^2}{\sum_{i=1}^{K}\left(y_i - \bar{y}\right)^2}$$

Measuring Model Accuracy: Classification

- Assume a set of data $(\mathbf{x}_1, y_1),\dots,(\mathbf{x}_K, y_K)$
- Classification accuracy

$$error_{M(\mathbf{x})} = \frac{1}{K}\sum_{i=1}^{K} c\left(\mathbf{x}_i, y_i, M(\mathbf{x}_i)\right)$$

Where $$c\left(\mathbf{x}_i, y_i, M(\mathbf{x}_i)\right) = \begin{cases} 0 & \text{if} \quad y_i = M(\mathbf{x}_i) \\ 1 & \text{otherwise} \end{cases}$$

Picking the Best Learning Parameters

- Partition learning data into disjoint sets
  - Training Set $(\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_T, y_T)$
    - Used to build the model
  - Validation Set $(\mathbf{x}_1, y_1),\ldots,(\mathbf{x}_V, y_V)$
    - Used to evaluate model
- Pick the Learning Parameters that give the lowest error on the Validation Set

$$error_{M(\mathbf{x})} = \frac{1}{V}\sum_{i=1}^{V} c\left(\mathbf{x}_i, y_i, M(\mathbf{x}_i)\right)$$

How Big Should the Training and Validation Sets Be?

- It Depends…
- If you have Lots of data for learning
  - Randomly putting half the data into each set is often sufficient
- If you only have a Small data set for learning
  - Usually do N-Fold Cross Validation

N-Fold Cross-Validation

- Partition the data $D_0 = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_M, y_M)\}$ into N disjoint sets $T_1, \ldots, T_N$
- For i from 1 to N, do
  - Use T_i for validation and the remaining S_i for training
    - Training Set: $S_i = \{D_0 - T_i\}$
    - Error on validation T_i : $error_{T_i}$
- Return the average error on validation sets

$$error_{M(\mathbf{x})} = \frac{1}{N} \sum_{i=1}^{N} error_{T_i}$$

Pick the learning parameters that minimize this error!

Does My Cross Validation Error Reflect the True Error of My Model?

- ***No***!!!!!!!!!!!!!!!!!!!
  - It is biased by the validation sets!
- To estimate the true error, we need to do randomized experiments
  - e.g. 100 experiments (or as many as you can)
  - Each experiment consist of random divisions of the data in learning and test sets. e.g.
    - 90% data for learning (***use cross validation on this set to pick learning parameters***)
    - 10% for testing
  - Report average test error over the 100 experiments

Recipe for Building Models From Data: The Frequentist Approach

1. Using K-Fold cross-validation, find the learning algorithm (and associated learning parameters) that minimize the error on the validation sets
2. Use this algorithm (and these learning parameters) on the entire dataset to build your final model.

Estimating the True Error of the Model: The Frequentist Approach

- Use randomized experiments:
  - Each experiment consist of random divisions of the data in learning and test sets. e.g.
    - 90% data for learning (***use cross validation on this set to pick learning parameters***)
    - 10% for testing – used to estimate the error
- Do as many random experiments as you have time for – the more random experiments, the better your estimate error

Pseudo Code for K-Fold Cross Validation

- Divide data into K folds
- For each algorithm and each set of learning parameter values
  - Build K models (each with a different validation set)
    - K-1 data folds for building model
    - Remaining fold for validation
  - Record average error of the K models on the associated validation sets for each algorithm and each set of learning parameter values
- Return algorithm and associated learning parameter values that gave minimum error on validation sets

Why do K-Fold Cross Validation?

- K-Fold CV is a principled way of picking the best learning algorithm and associated learning parameters for a dataset
- The learning algorithm and associated learning parameters are used on the entire dataset to build the final model
- K-Fold CV ***DOES NOT*** GIVE AN ESTIMATE OF ERROR ON FUTURE DATA!!!!!

Pseudo Code for Estimating the Error Rate of a Final Model on Future Data

- Pick a set of learning algorithms and associated learning parameters which K-fold CV will search over
  - These are usually obtained using a preliminary K-fold CV experiment, or some prior experience.
- Do N randomized experiments
  - Each takes a random subset of the data (say 90%) for training (D_trn), and the remainder for testing (D_tst)
  - Pass the training set (D_trn) to K-fold cross validation to obtain the best learning algorithm and associated learning parameters for this training set (D_trn)
  - Use this learning algorithm and these learning parameters to build a model using the training set (D_trn)
  - Calculate the error rate of this model on the test set (D_tst)
  - **<u>Return the average error of on all N test sets</u>**

What does the average error in the previous slide mean?

- The average returned is an unbiased estimate of error on future data obtained when the **final model is constructed as follows**:
  - All the training data is passed to a K-fold CV algorithm
    - The K-fold CV algorithm returns the learning algorithm and associated learning parameters that gave the best error rates
    - The search space of the K-fold CV algorithm **MUST** be the same as the search space used in the K-fold CV algorithm in the randomized test!
  - The final model is constructed by passing ALL the data to the learning algorithm chosen above, using the associated learning parameters

Compute Limited Pseudo Code for Estimating the Error Rate of a Model on Future Data

- Use K-fold CV **once** to pick a specific algorithm and a specific set of learning parameter values
- Do N randomized experiments
  - Each takes a random subset of the data (say 90%) for training (D_trn), and the remainder for testing (D_tst)
  - Use the above learning algorithm and learning parameters to build a model using the entire training set (D_trn)
  - Calculate the error rate of this model on the test set (D_tst)
  - **Return the average error of on the test sets**

What does the average error in the previous slide mean?

- The average returned error is an unbiased estimate of error on future data obtained when the **final model is constructed as follows**:
  - The final model is constructed by passing ALL the data to the learning algorithm and associated learning parameters chosen before the randomized experiments were started
- This is a valid error estimate… HOWEVER
  - By limiting yourself to specific learning parameter values, experimental evidence suggests that your error estimate may not be as accurate as if you have a range of values that you search over…

Typical values for K and N?

- K-fold CV: K = 5 or 10
  - The larger K, the better…
- For the Nearest Neighbor algorithm, the number of folds usually equals the number of training examples.
- N-random experiments: N = 50 or 100
  - The larger N, the better…

Rules of Thumb for K-fold CV

- **<u>Classification:</u>** if possible, each fold should have approximately equal proportion of classes
- **<u>Regression:</u>** if possible, each fold should have approximately equal range of output values

Quiz 6

1. What is the purpose of cross-validation
2. Does cross validation give a estimate of error on future data?
3. Can cross-validation be used for feature (input) selection? Remember, feature input selection is a way of deciding which features are relevant for maximizing the accuracy on future data.