Learning from Examples

Margrit Betke

CS 640

Fall 2023

First 10 slides based on Russell and Norvig, 3rd edition, Sections 1, 2, and 4.

Forms of Learning

AI system "learns" if it improves its performance *based on observations & feedback from its environment*

Unsupervised learning (=clustering):

Input: vector of attributes. No explicit feedback.

Supervised learning:

Input: vector of attributes. Feedback = output of continuous or discrete value(s) = labels of input examples.

Reinforcement Learning:

Actions are rewarded or punished.

In 440/640: Supervised Learning

Training set = N example input-output pairs

(x_1,y_1), (x_2,y_2), ..., (x_N,y_N)

where each y_j was generated by an unknown function f, such that $f(x) = y$. Function f needs to be learned.

- The AI system finds a function h that approximates f. For example, the AI system trains a neural net that computes $h(x_j)=y_j$ for all examples in the training set.
- There are no guarantees that new inputs $h(x_{new}) \approx f(x_{new})$.
- To measure accuracy (Is h ≈ f?), we use a test set of labeled examples = input-output pairs (≠ training set!):

A neural net is trained well if $h(x_{test}) \approx y_{test}$ for all test example pairs (x_{test}, y_{test}).

Classification versus Regression

Depending on the type of output, the learning problem is a

- Classification problem:

  Output values: number of classes (discrete, finite)

- Regression problem:

  Output values are numbers, e.g., tomorrow's temperature

Occham's Razor

= Law of succinctness

Which hypothesis among h_1, h_2, h_3 ... should the AI system choose?

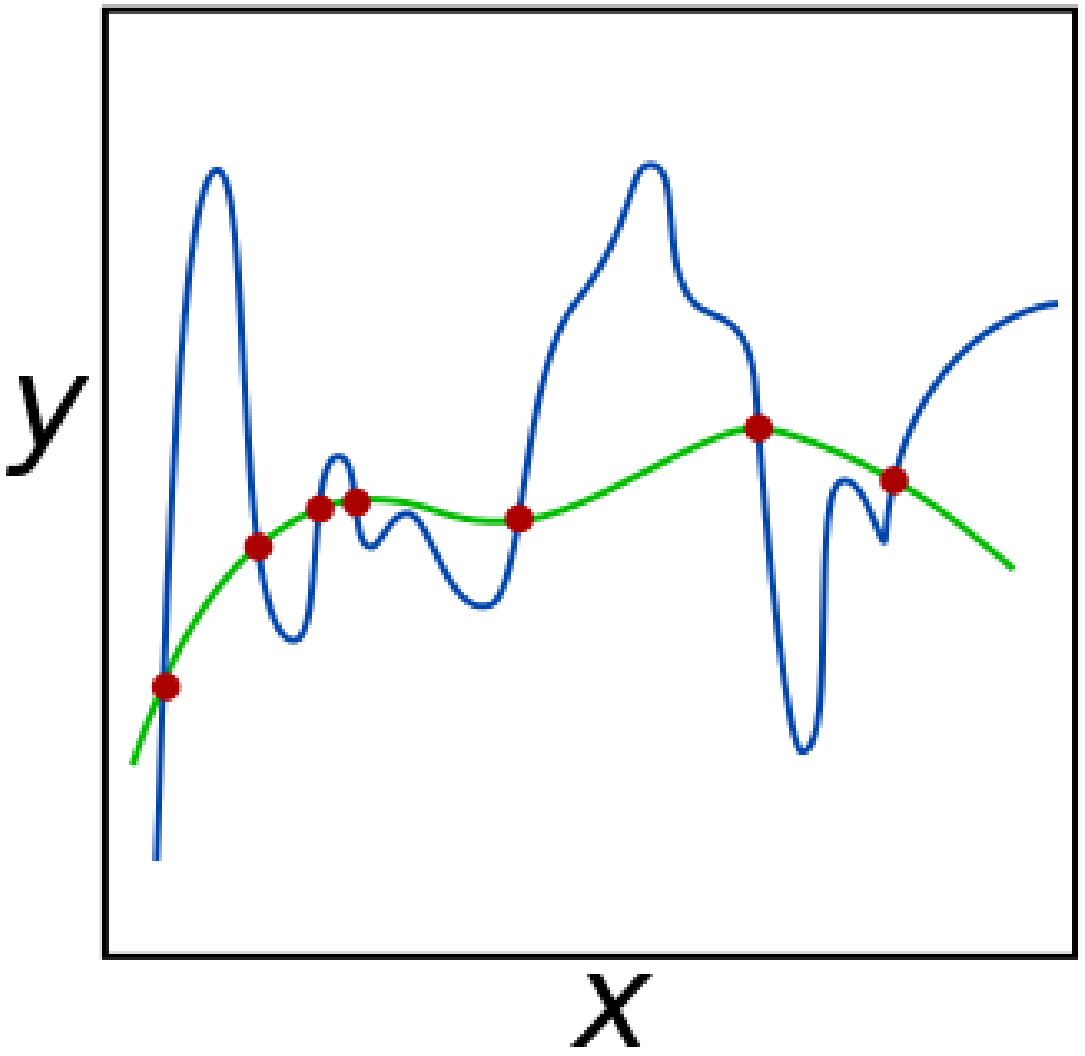
Choose the simplest hypothesis consistent with the data.

The simplest explanation will be the most plausible until evidence is presented to prove it false.

Example: Prefer a degree-1 polynomial (line) over a degree-7 polynomial

Trade-off between complex hypothesis that fit training data well and simpler hypotheses that may generalize better (and can typically be computed faster)

Occam's Razor: Choose green over blue model for h



Source: Wikipedia

Overfitting

- Avoid choosing an excessively complex learning system= model= hypothesis=neural net h.
- h is too complex if it has too many parameters relative to the number of observations.
- A model which has been overfitted will generally have poor predictive performance, as it can exaggerate minor fluctuations in the data.
- Higher-degree polynomials or complicated neural nets with many hidden layers and nodes fit the data better but may lead to overfitting.

Overfitting

- Avoid choosing an excessively complex learning system= model= hypothesis=neural net h.
- h is too complex if it has too many parameters relative to the number of observations.
- A model which has been overfit will generally have poor predictive performance, as it can exaggerate minor fluctuations in the data.
- Higher-degree polynomials or complicated neural nets with many hidden layers and nodes fit the data better but may lead to overfitting.

Solutions:

- Use “wrapper” to enumerate models h according to model size (e.g., number of nodes or layers in neural net). Select model with smallest error.
- Feature selection: Simplify model by discarding irrelevant attributes (dimensionality reduction). See below.
- Minimum description length: Select model with smallest number of bits required to encode program and data.

Loss Functions: SPAM Example

Loss value $L(y_{true}, y)$

= cost of misclassifying email:

A "false positive," e.g. hypothesize "non-spam" but it is truly "spam" L(spam,non-spam) = 1

Annoying but simply delete email.

A "false negative," e.g. hypothesize "spam" but it is truly "non-spam" L(non-spam, spam) = 10

Much worse, you may miss an important email.

Typical Loss Functions

- Absolute value loss: $L_1(y_{true},y) = |y_{true}-y|$
- Squared error loss = Euclidean loss:
$$L_2(y_{true},y) = (y_{true}-y)^2$$
- 0/1 loss: $L_{0/1}(y_{true},y) = 0$ if $y_{true} = y$, else 1

When training a classifier h:

Find h that minimizes the empirical loss

$$\text{EmpLoss}(h) = 1/N \sum L(y_{true,\,i}, h(x_i))$$

= mean error over a set of N examples $(x_i, y_{true,\,i})$

Cross-Validation

Holdout cross-validation =
Randomly split available (input,output) pairs into a training set to learn h and a test set to test the learned h.

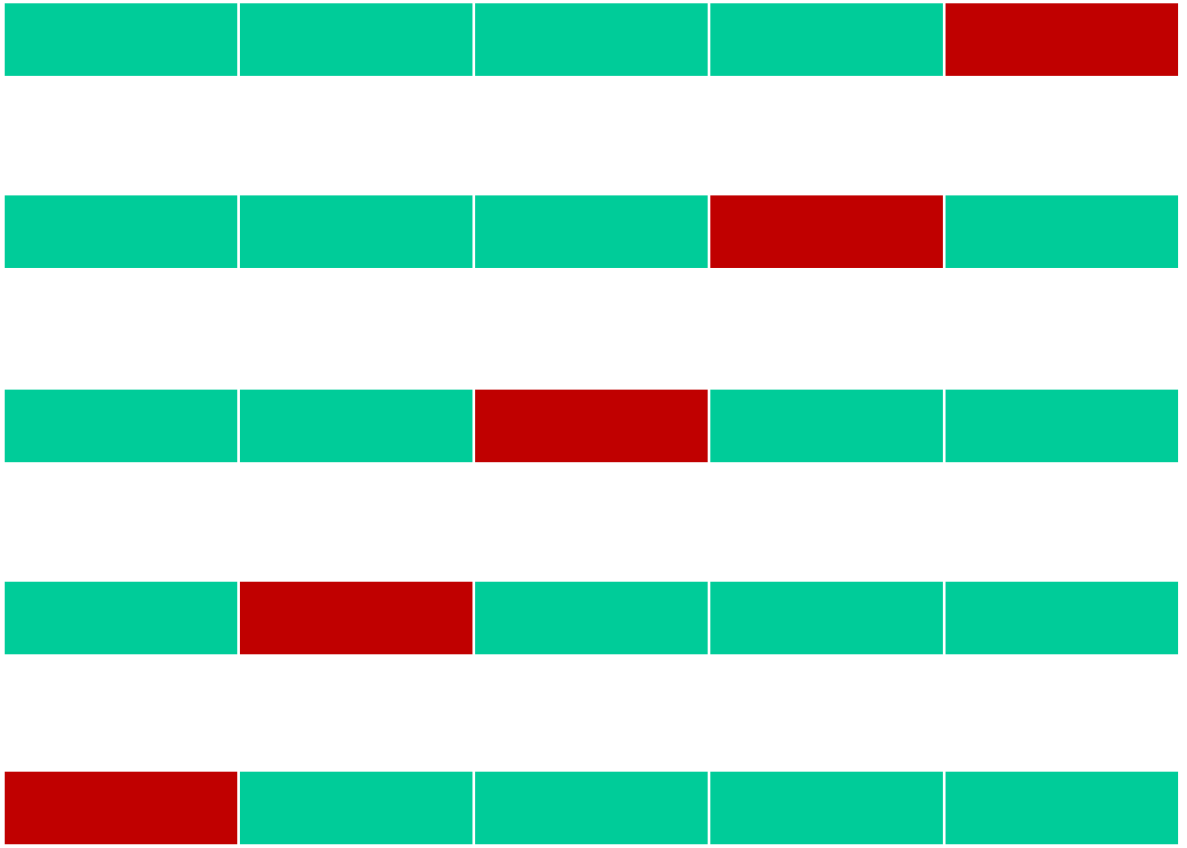
k-fold cross-validation =

- Split data into k equal subsets.
- Perform k rounds of learning. Each round leaves 1/k examples out of the training set that can then be used as the test set.
- The average test set score should be a better estimate than a single score (need to keep k h's around for prediction). Typically, k=5 or 10.

Leave-one-out cross validation: k=N.

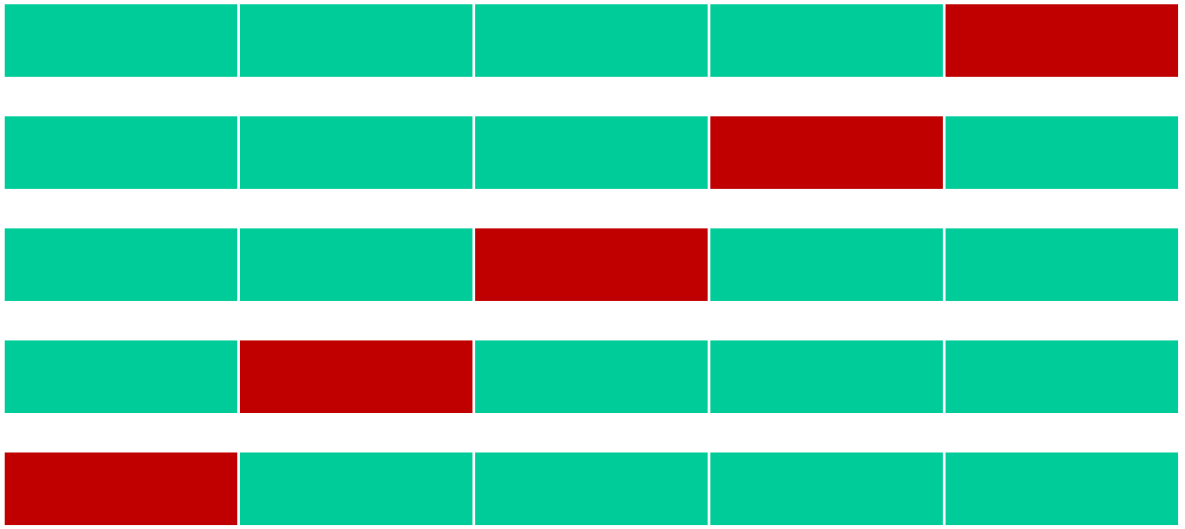
Simple 5-Fold Cross-Validation

5 Folds of Labeled Dataset: **Training** & **Testing**



Simple 5-Fold Cross-Validation

1. Create 5 folds of the labeled dataset for **training** & **testing**



2. Train 5 AI models
3. Conduct ROC analysis for each model
4. Report average performance (accuracy etc.)

5 Fold Cross-Validation with Random Selection of Folds

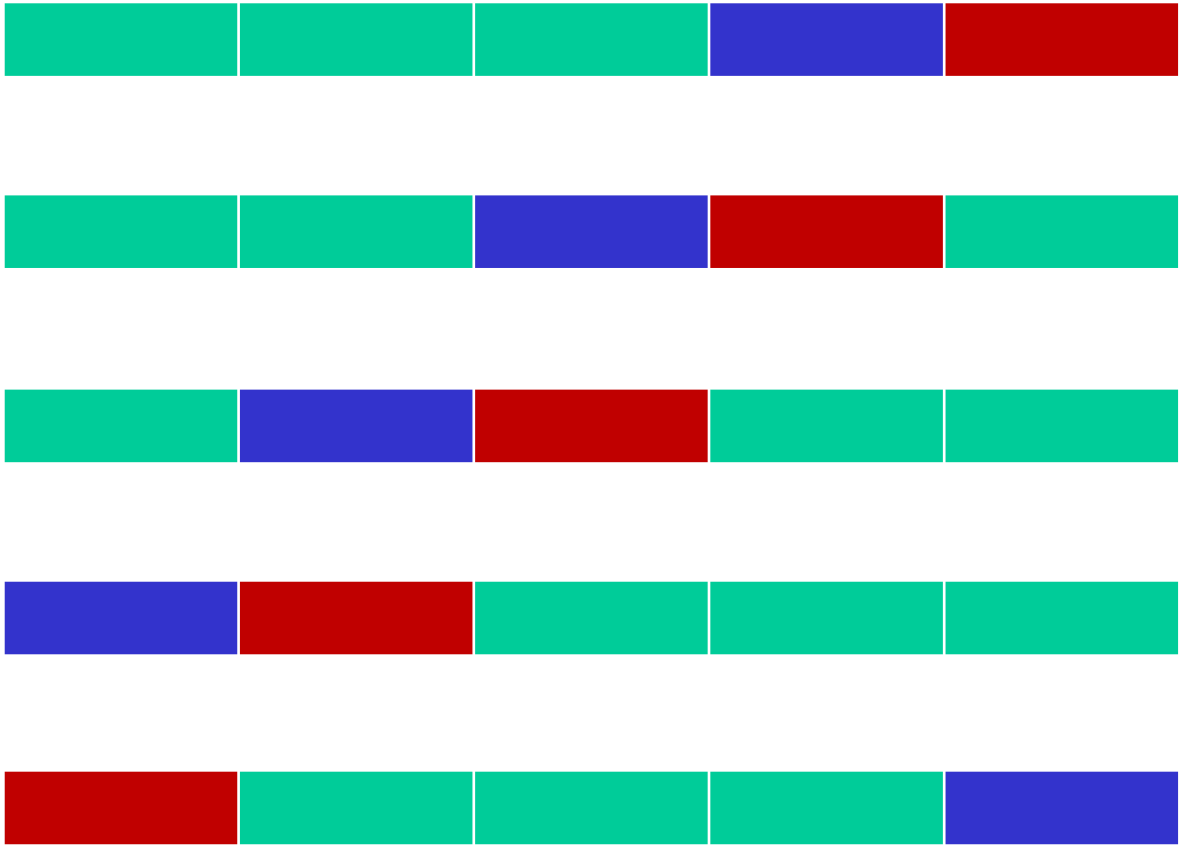
What if you were simply lucky in your **training** & **testing** folds?

2. Randomly create a different set of 5 folds & report average performance

3. Do this n times & report 1/n of average performance

Cross-Validation with Train/Validation/Test Sets

Training, Validation, & **Testing**



Why Validation?

With the validation data, we tune the hyperparameters of an AI model.

The validation process tells us whether training is moving in the right direction. It can be performed after each training epoch.

The validation process helps prevent our model from overfitting to the training data by challenging it on the unseen validation data.

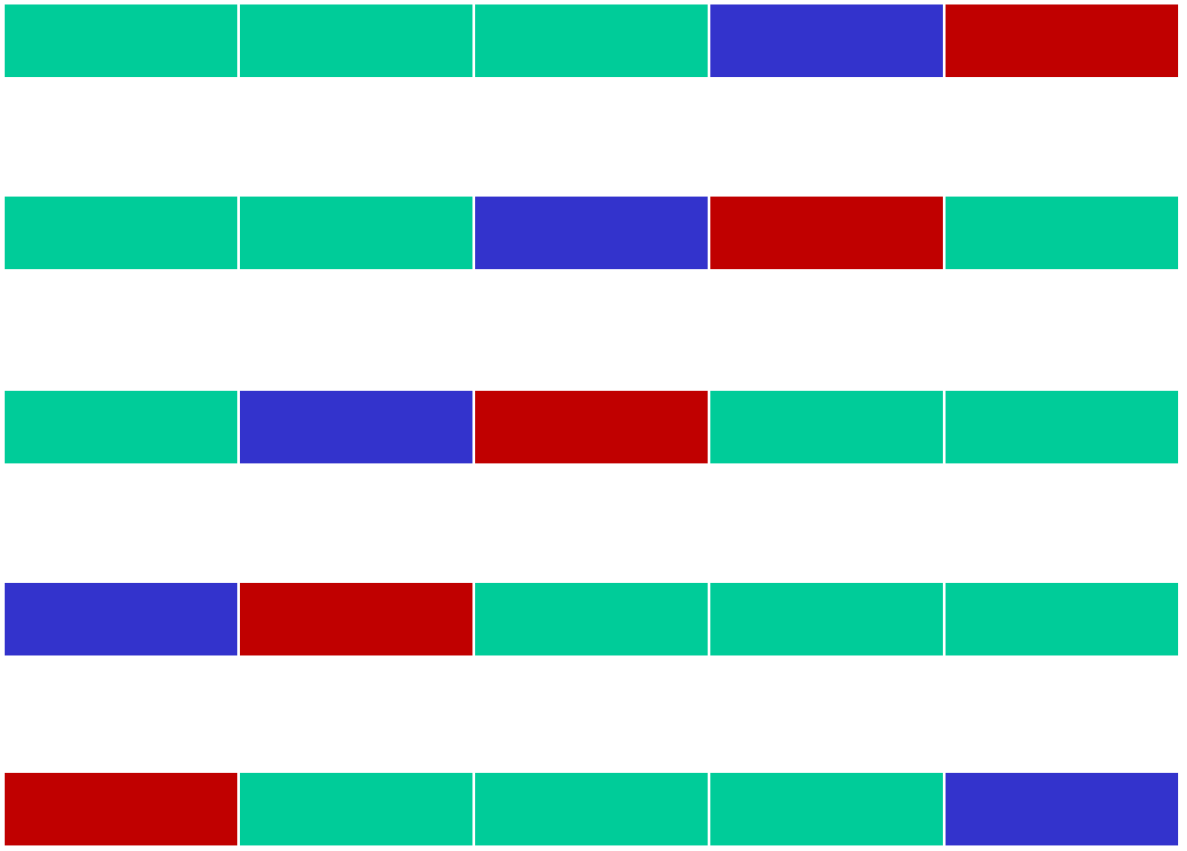
The validation data must therefore be separate from the training data (and of course from the test data).

What are common Train/Validation/Test Splits?

Training, Validation, & Testing

60%, 20%, 20% **or** **80%, 10%, 10%**

5 folds 10 folds



How to Train/Test when feature reduction is used to avoid overfitting

Feature selection: Simplify model by discarding irrelevant attributes (reduction to k features).

Common questionable practice for regressors:

1) For each feature, compare the vector of feature values for all samples in the dataset with the vector of desired output values.
2) Sort features by result of this comparison
3) Use highest ranked k features in subsequent usual cross-validation procedure

How to Train/Test when feature reduction is used to avoid overfitting

Feature selection: Simplify model by discarding irrelevant attributes (reduction to k features).

Common better practice for regressors:

1) For each feature, compare the vector of feature values for samples in the training dataset with the vector of desired output values.
2) Sort features by result of this comparison
3) Use highest ranked k features in subsequent nested cross-validation procedure

How to compare features and desired regression values

Normalized correlation coefficient:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

where

x_i = value of i^{th} feature, $\bar{x}$ mean feature value

y_i = value of i^{th} desired regression value (ground truth), $\bar{y}$ mean desired regression value

Nested Cross Validation

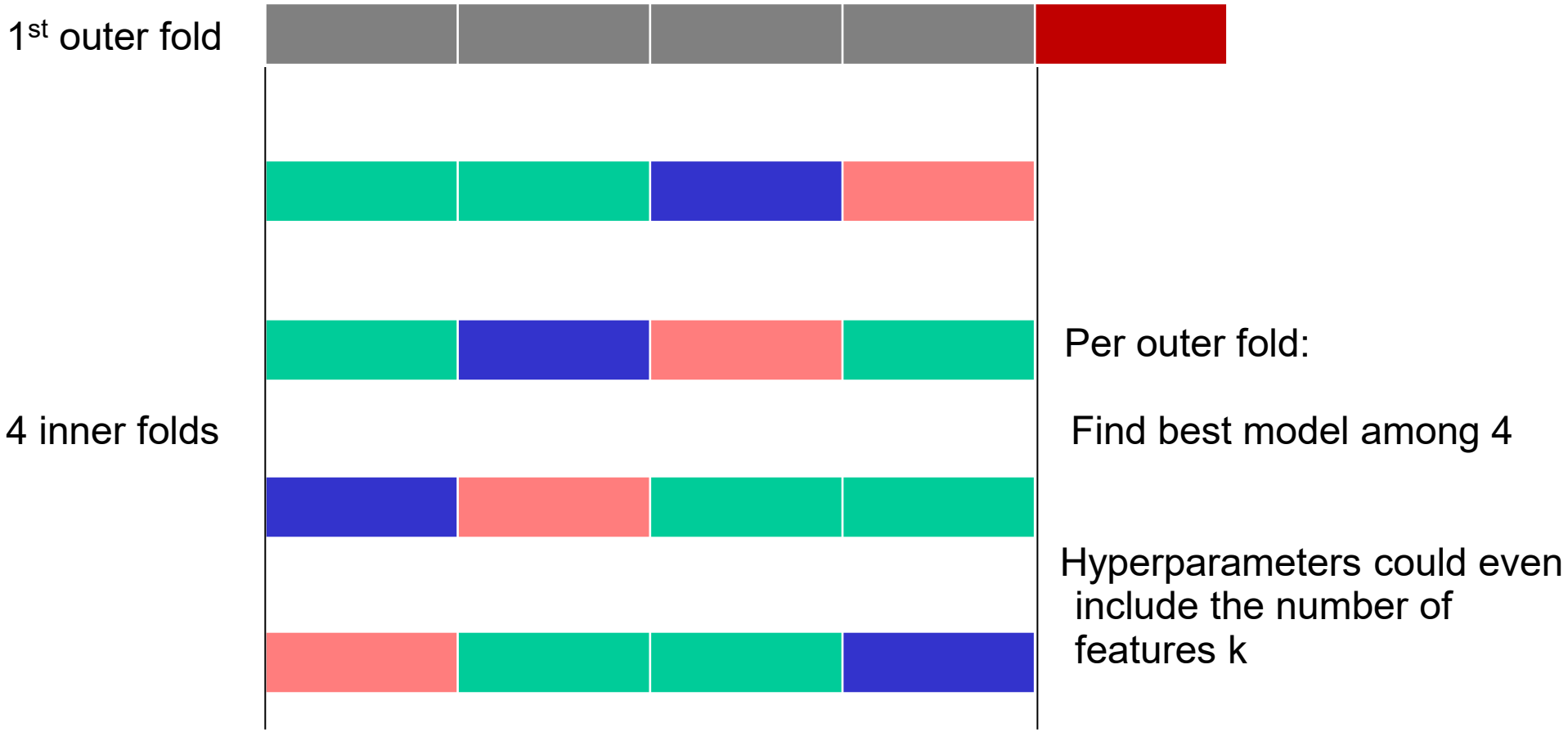
Training, Validation, & Testing

What if [[A|B] C] split was a bad choice? [[B|A]C]?

Nested Cross Validation

Here: 5 outer folds, 4 inner folds, report performance of 5 models

Training, Validation, & Testing



Research Paper with Nested Cross Validation:

Saurav Chennuri, Sha Lai, Anne Billot, Maria Varkanitsa, Emily J. Braun, Swathi Kiran, Archana Venkataraman, Janusz Konrad, Prakash Ishwar, and Margrit Betke. Fusion Approaches to Predict Post-stroke Aphasia Severity from Multimodal Neuroimaging Data. Accepted at the International Conference on Computer Vision Workshop on Computer vision for Automated Medical Diagnosis (ICCV CVAMD 2023). Paris, France, October 2, 2023. 10 pages.

Abstract:

This paper explores feature selection and fusion methods for predicting the clinical outcome of post-stroke aphasia from medical imaging data. Utilizing a multimodal neuroimaging dataset derived from 55 individuals with chronic aphasia resulting from left-hemisphere lesions following a stroke, two distinct approaches, namely Early Fusion and Late Fusion, were developed using Support Vector Regression or Random Forest regression models for prognosticating patients' functional communication skills measured by Western Aphasia Battery (WAB) test scores. A supervised learning method is proposed to reduce the number of features derived from each imaging modality. …

Saurav was a MS in AI student who graduated in May 2023.