

Week 10 - Predictive modeling

L10-04. Predictive modeling in practice – model building

Lada Kohoutová

30 April 2021



- Code available at https://github.com/cocoanlab/interpret_ml_neuroimaging
- **Matlab**
 - Statistics and Machine Learning toolbox
 - Signal Processing toolbox
- **CanlabCore Tools**
(<https://github.com/canlab/CanlabCore>)
- **Statistical Parametric Mapping (SPM) toolbox**
(<https://www.fil.ion.ucl.ac.uk/spm/>)



Useful functions

- CanlabCore Tools
- `fmri_data` object

```
obj =  
  
    fmri_data with properties:  
  
        source_notes: 'Info about image source here'  
            X: []  
            mask: [1x1 fmri_mask_image]  
        mask_descrip: 'REMOVED: CHANGED SPACE'  
    images_per_session: []  
            Y: []  
            Y_names: []  
            Y_descrip: 'Behavioral or outcome data matrix.'  
        covariates: []  
        covariate_names: {''}  
    covariates_descrip: 'Nuisance covariates associated with data'  
    history_descrip: 'Cell array of names of methods applied to this data, in order'  
    additional_info: [0x0 struct]  
        dat: [352328x1 single]  
        dat_descrip: []  
        volInfo: [1x1 struct]  
    removed_voxels: 0  
    removed_images: 0  
        image_names: 'brainmask.nii'  
        fullpath: '/Users/ladakohoutova/Desktop/CocoanLab/gitrep/CanlabCore2/CanlabC  
        files_exist: 1  
        history: {1x4 cell}
```



Useful functions

- CanlabCore Tools
- fmri_data object
- predict function
 - Works on fmri_data objects
 - Can run various algorithms (e.g., multiple regression, SVM, SVR, PCR, etc.)
 - Cross-validation
 - Bootstrap

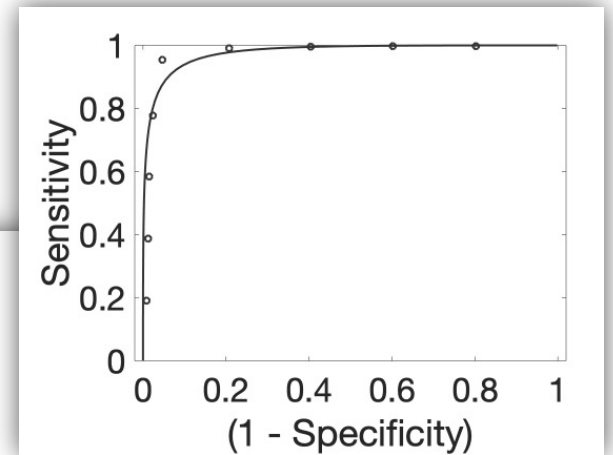
```
out =  
  
    struct with fields:  
  
        Y: [62x1 double]  
        algorithm_name: 'cv_svr'  
        function_call: '@(xtrain, ytrain, xtest, cv_assignment) cv_svr(xtr  
        function_handle: [function_handle]  
        yfit: [62x1 double]  
        err: [62x1 double]  
        error_type: 'mse'  
        cverr: 125.2375  
        nfolds: 'nfolds'  
        cvpartition: [1x1 struct]  
        teIdx: {1x5 cell}  
        trIdx: {1x5 cell}  
        other_output: []  
        other_output_descrip: 'Other output from algorithm - trained on all data'  
        other_output_cv: []  
        other_output_cv_descrip: 'Other output from algorithm - for each CV fold'  
        mse: 125.2375  
        rmse: 11.1910  
        meanabserr: 9.0577  
        pred_outcome_r: 0.5219  
        weight_obj: [1x1 statistic_image]  
        WTS: []
```



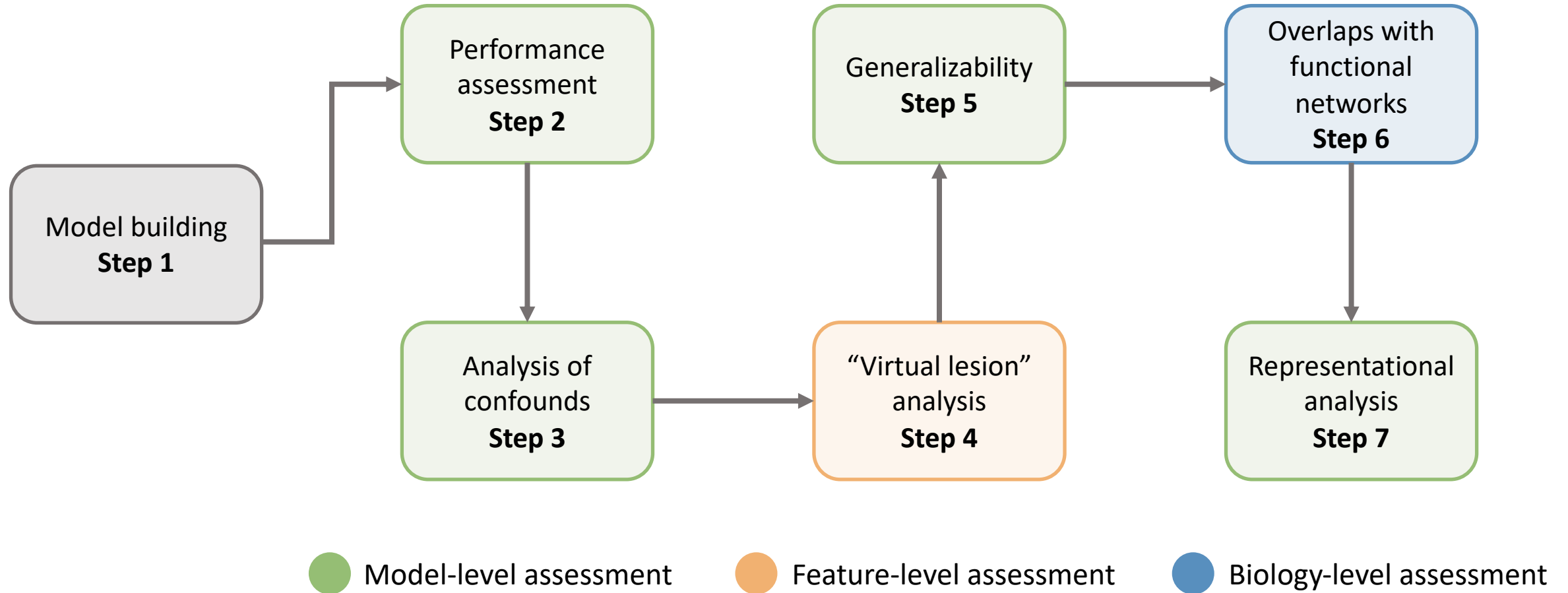
Useful functions

- CanlabCore Tools
- fmri_data object
- predict function
- roc_plot function
 - Calculates accuracy, sensitivity, and specificity
 - Draws ROC curve for input values and binary outcome
 - Various threshold options (e.g., 'twochoice')
- apply_mask function
- canlab_pattern_similarity function

```
ROC_8fold =  
  
  struct with fields:  
  
      baserate: 0.5000  
      all_vals: [1x1 struct]  
      class_threshold: 0  
      sensitivity: 0.9576  
      specificity: 0.9534  
      AUC: 0.9782  
      AUC_descrip: 'Numerically integrated, nonparametric area under curve'  
      threshold_type_for_misclass: 'A priori threshold'  
      observations: [1x1 struct]  
      PPV: 0.9536  
      accuracy: 0.9555  
      N: 944  
      accuracy_p: 0  
      accuracy_se: 0.0067  
      Gaussian_model: [1x1 struct]  
      line_handle: [1x2 Line]  
      sensitivity_ci: [0.9385 0.9748]  
      specificity_ci: [0.9343 0.9718]  
      PPV_ci: [0.9353 0.9722]  
      Binned_output: [1x1 struct]
```



Tutorial workflow



Example data

- fMRI data from Woo et al., *Nat Commun*, 2014
- **Binary classification** – heat vs. rejection
- Support Vector Machine (SVM)

