PRoNTo for developers & ongoing developments

Tong Wu

Course 2018

May 14th- 15th

UCL, London





Outline

- PRoNTo architecture
- Module 1: Data & Design
- Module 2: Prepare feature set
- Module 3: Specify Model
- Module 4: Run Model/Model estimation
- Module 5: Compute weights
- Ongoing developments



PRoNTo architecture

- Modular design: 5 independent modules with different functionalities
- User interfaces: GUI and Batch
- Core functions: implement functions of different modules





PRoNTo architecture

PRoNTo ::			Batch Editor	A BasicIO PRONTO ¥
Pattern Recogniti data	on for Neuroimaging Toolbox		Module List Data & Design ^ Feature set/Kern Specify model	Current Module: Data & Design Help on: Data & Design Directory <x Groups <x< th=""></x<></x
Main steps	Review options		Run model Compute weights	Masks <-X Review No
Data & Design	Review data	User interfaces		
Prepare feature set	Review kernel & CV			
Specify model	Display results			
Run model	Display weights		< +	
Compute weights	Batch		Data & Design Specify the data ar	nd design for each group (minimum one group).
	Credits		This branch contain * Directory	ns 4 items:
GU	l and			Batch and
GUI specif	fic functions		Batch	specific functions
		(Shared) Core functions	<u> </u>	



PRoNTo architecture - GUI





PRoNTo architecture - Batch

承 Batch Editor		
File Edit View SPM	BasicIO PRoNTo	لا ر
D 🛩 🖬 🕨		
Module List	Current Module: Feature set/Kernel	
	Help on: Feature set/Kernel Load PRT.mat Feature/kernel name Modalities Use one kernel per modality	<-X <-X <-X No
		-
	Current Item: Load PRT.mat	
		Â
	Specify	
Load PRT.mat Select data/design s	tructure file (PRT.mat).	





Data & Design – PRT structure



• User inputs:

✓ groups (Dementia, NonDementia)
 ✓ Subjects (Dementia: S1, S2, NonDementia: S1, S2)
 ✓ Image types (modalities – white/gray matter)
 ✓ Covariates, designs, masks (WholeBrain.nii)

PRT.group			
gr_name	subject		
Dementia	1x2 struct		
NonDementia	1x2 struct		

PRT.masks				
mod_name	WhiteMatter			
fname	/WholeBrain.nii			



Prepare feature set

Main inputs



Main outputs

- A file array (.dat binary file)
 in 1st-level mask
- A kernel matrix (.mat file)
 use 2nd-level mask
- Updated masks and/or atlases
- New fields in PRT structure:
 ✓ PRT.fas
- ✓ PRT.fs



Prepare feature set



Current Folder	$\overline{\mathbf{O}}$
🗋 Name 🔺	
🕂 Fastest.mat	
Feature_set_Run1.dat	
Feature_set_Run2.dat	
🕂 PRT.mat	

PRT 🗶 PRT.fs	≫ PRT.fs.id_mat ×	
1x1 struct with 6 fi	elds	
Field 🔺	Value	
👍 dir	'C:\Users\sony\Desktop	
🔁 group	1x1 struct	
Η design 🛛 []		
🔁 masks	1x2 struct	
🔁 fs	1x1 struct	
圭 fas	1x2 struct	



- PRT.fas (file array structure): saves all features in the 1st-level mask.
- Modality (v2.1): runs/sessions, or Gray matter/White matter.





• PRT.fs (feature set/kernel): use 2nd-level mask or atlas







fMRI experiment - concatenate by samples





MRI modalities - combine modalities



Atlas 2nd level mask (ROI) Feature set Review Kernel, Model, CV

The ID matrix in PRT.fs : PRT.fs.id_mat and PRT.fs.id_col_names

group	subject	modality	condition	block	scan
1	1	1	0	0	1
1	1	1	0	0	2
1	1	1	6	1	3
1	1	1	6	1	4
1	1	1	0	0	5
1	1	1	0	0	6
1	1	1	2	1	7
1	1	1	2	1	8
1	1	1	0	0	9
:	÷	:	:	:	:



Specify model – input and output

Main inputs



Main outputs

- New fields in PRT structure
- ✓ PRT.model.model_name✓ PRT.model.input



Specify model - model type and machine

🔛 prt_machine_rvr.m



Class labels

 ✓ Class labels 1,2,3,... refer to Class 1, Class 2, Class 3, respectively. They'll be converted to specific machine formats if needed.

• Machine

- Different machines are interfaced between PRoNTo and several third-party libraries
- ✓ Custom machine (Batch)



Specify model - model type and machine

🔺 🛃 Batch Editor		- 0	×
File Edit View SPM	BasiclO PRoNTo		з
D 🚅 🖬 🕨			
Module List	Current Module: Specify model		
Specify model ^ Run model DE	Help on: Specify model Load PRT.mat	om Machine\PRT.mat	^
	Model name	MyMachine	
	Use kernels	Yes	
	Feature sets	IXICustomMachine	
	. Regression		
	Groups		
	Group		
	Group name	Aged	
	Machine	TUZXT double	
	Custom machine		~
	Current Item: Model name		
	MyMachine		^
			~
< >	Specify		
Madalasana			
Name for model			
A String is entered			
The string must hav	e at least 1 characters.		
			¥

Machine •

✓ Custom machine (Batch): e.g. , prt_machine_esvr.m

Module List	Current Module: Specify model	
Specify model \land	Groups	^
Run model DI	Group	
	Group name	Aged
	Subjects	102x1 double
	. Machine	
	Custom machine	
	Function	\prt_machine_esvr.m
	Arguments	-q-s3-t4-c1
	Cross-validation type	
	. k-folds CV on subjects	
	k	2



읗

Specify model – machine

function output = prt machine esvr(d, args) % Run Epsilon-SVR - wrapper for libSVM % FORMAT output = prt machine esvr(d,args) Inputs: structure with data informa d 읗 .train - training data (cell arr each [Ntr x D]). each m 읓 of the data. This is us * multiple kernel learnin 읗 - testing data (cell arr 읗 .test [Nte x D]) * .tr targets - training labels (for cl 읗 regression) (column vec * .use kernel - flag, is data in form o 읗 form of features (false) 2 - libSVM arguments args Output: output of machine (struct). output * Mandatory fields: s. .predictions - predictions of classi 읓 * Optional fields: 8 .func val - value of the decision fu 읗 .type - which type of machine th 읗

- Machine inputs:
- ✓ Structure 'd' containing training and test data information
- ✓ Machine argument structure 'args': '-q -s 3 -t 4 -c 1'
- Machine outputs
- \checkmark A structure 'Output' containing predictions and model parameters



GUI or Batch

🕍 prt_machine_esvr.m

Libsvm accepted argument format: '-q -s 3 -t 4 -c 10'







Run model – inputs and outputs



• Run model performs model estimations:

✓ hyperparameter optimization by using inner CV schemes
 ✓ model training and testing using external CV schemes
 ✓ computing statistics for model performance measures



Run model - Cross Validation (CV)





Run model: tips for efficient scripting

- Change any subfields in PRT.model.input, then call prt_cv_model.m, a new model will be created – but the former model will be overwritten. If you don't want to, re-name the new model.
- For example, change the value of the hyperparameter C for the custom machine from 1 to 100:

>>PRT.model.input.machine.args = '-q -s 3 -t 4 -c 100' >>prt_cv_model(PRT,in)

A new model using C = 100 is estimated and stored using the same model name. $^{\scriptscriptstyle 21}$



Run model – statistical metrics



- How to access stats in PRT?
- ✓ Fold level: PRT.model.output.fold(i).stats
- ✓ Model level: PRT.model.output.stats

📝 Editor - prt_stats.m		🔏 Variables - PR	T.mo	del.output.fold(1).stats	(
PRT × PRT.model × PRT.mod	el.output 🛛 🕹	PRT.model.output.fold	×	PRT.model.output.fold(1).stats	×
PRT.model.output.fold(1).stats					
Field 🔺	Value				
🕂 corr	0.5263				
<u></u> r 2	0.2770				
🛨 mse	25.5470				
🖶 nmse	1.0928				
<					



Compute weights – the general procedure





Compute weights – weights and machines





. . .

Developments – other considerations

- Backwards compatibility
- Mex files if compiled machine
- MATLAB backwards compatibility



Outline

- PRoNTo architecture
- Module 1: Data & Design
- Module 2: Prepare feature set
- Module 3: Specify Model
- Module 4: Run Model/Model estimation
- Module 5: Compute weights
- Ongoing developments



Ongoing development – two slides

• Data & Design: add new modalities of MEEG and .mat

承 PRoNTo :: Specify modality		_	o x
	Modality		
Name	Enter new		~
Data format	nifti		~
	nifti MEEC		
Files	.mat		
Design	Nedecian	~	
Design	No design	•	
O	< Can	cel	
			J ,





Ongoing development

- Non-kernel methods
- ✓ Linear classification machines in Liblinear: binary and multiclass
- ✓ Good for data with large samples and features
- ✓ Good for data with small number of features: 140 subjects vs 700 features
- Combining modalities from different types (MEEG, fMRI, .mat) with different number of features

• Test model - sharing models



- Website: http://www.mlnl.cs.ucl.ac.uk/pronto/prtsoftware.html
- Manual for users
- Developer's manual will come soon
- Video tutorials will come soon



Thank you very much!

Questions?

