Pattern Recognition for Neuroimaging Data

Edinburgh, SPM course

April 2013

C. Phillips, Cyclotron Research Centre, ULg, Belgium
http://www.cyclotron.ulg.ac.be

Overview

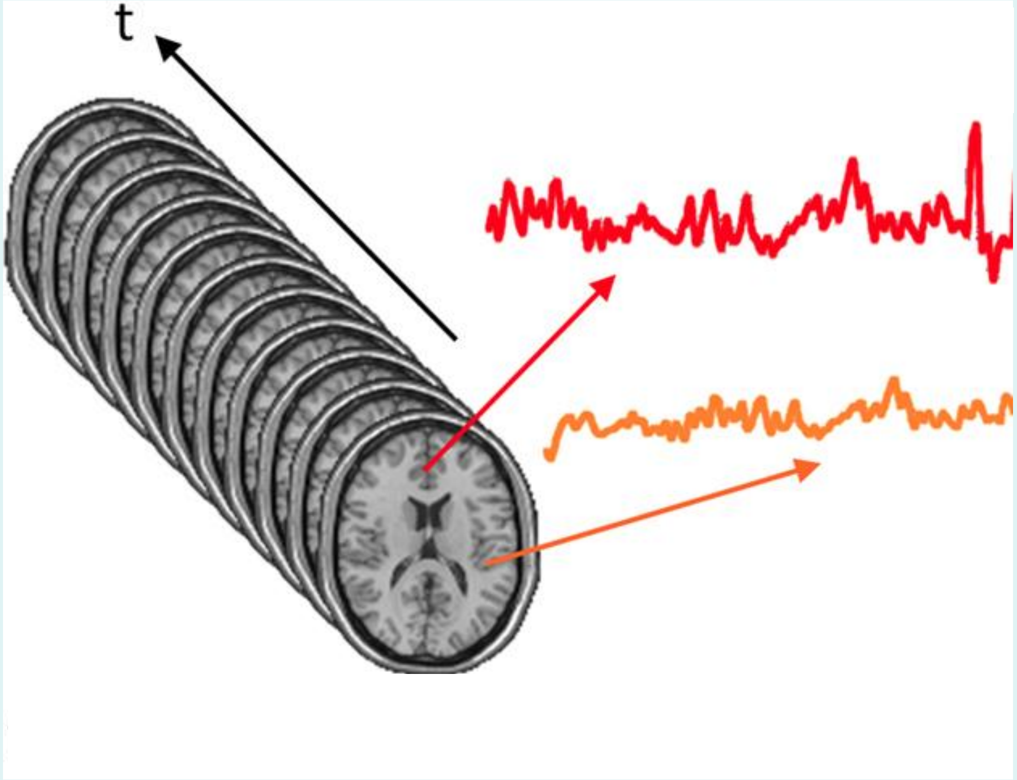
- Introduction
  - Univariate & multivariate approaches
  - Data representation
- Pattern Recognition
  - Machine learning
  - Validation & inference
  - Weight maps & feature selection
  - fMRI application
  - Multiclass problem
- Conclusion & PRoNTo

Overview

- Introduction
  - Univariate & multivariate approaches
  - Data representation
- Pattern Recognition
  - Machine learning
  - Validation & inference
  - Weight maps & feature selection
  - fMRI application
  - Multiclass problem
- Conclusion & PRoNTo

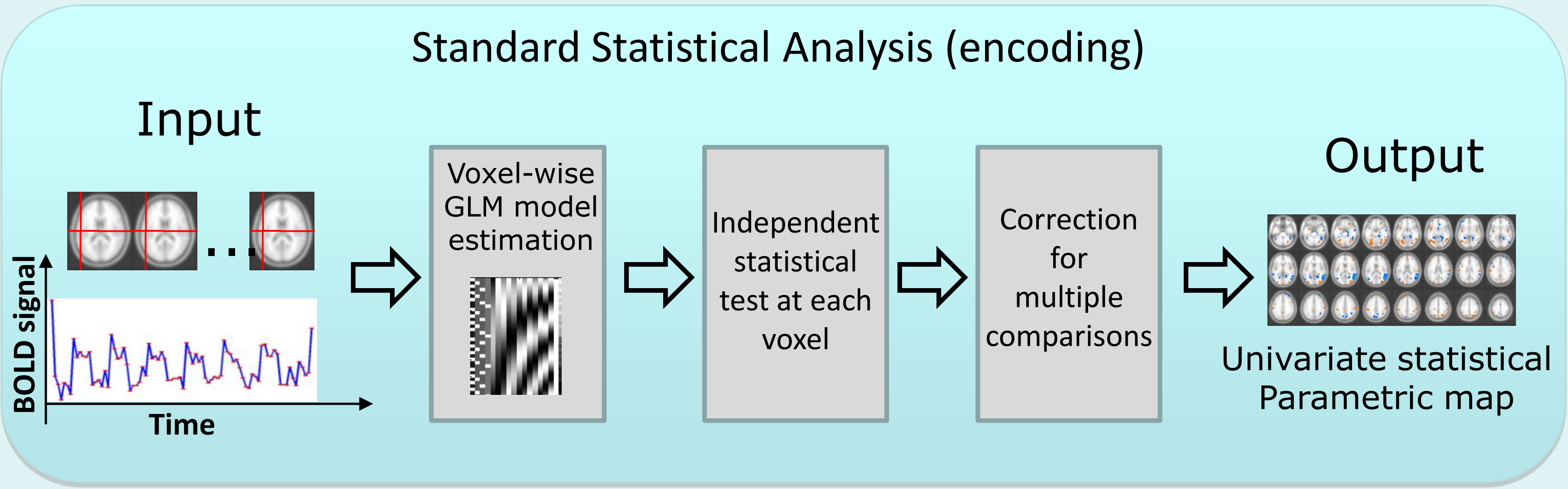
Introduction

fMRI time series = 4D image
= time series of 3D fMRI's
= 3D array of time series.



Univariate vs. multivariate

Standard univariate approach (SPM)

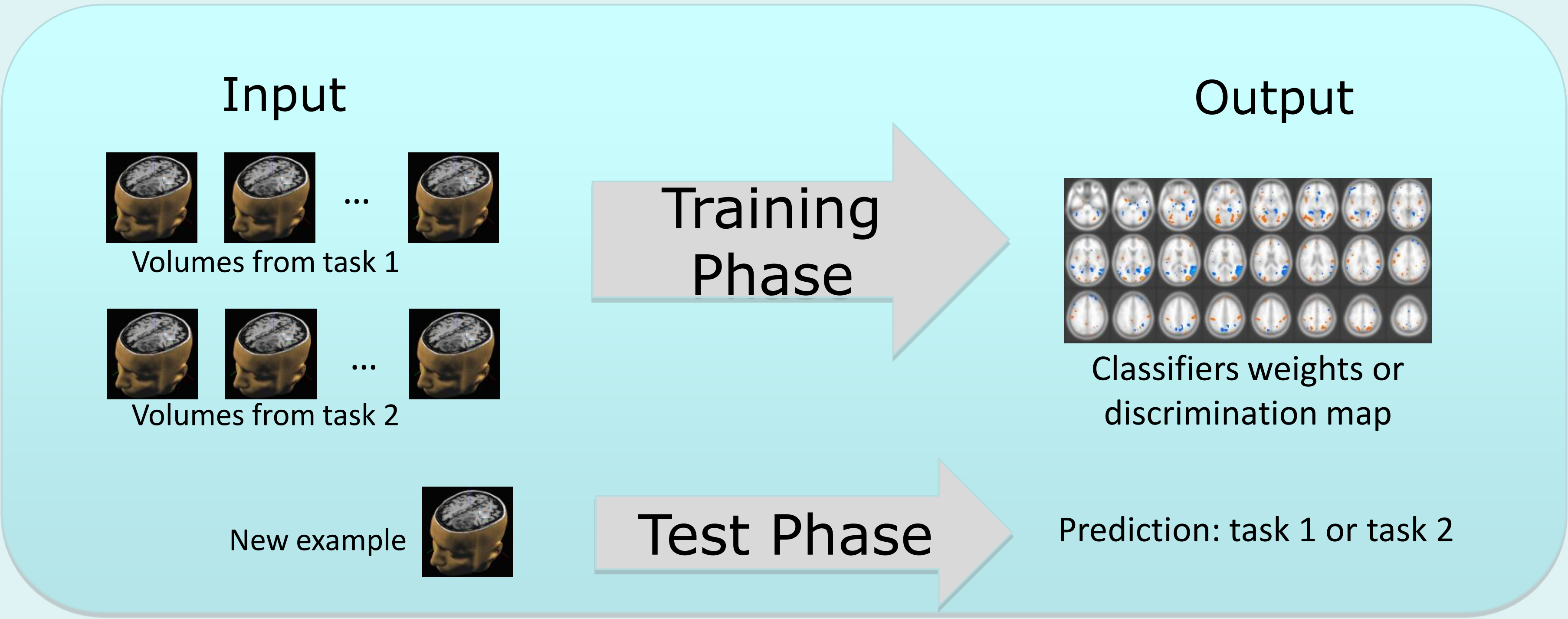


Find the mapping g from explanatory variable X to observed data Y

$g: X \rightarrow Y$

Univariate vs. multivariate

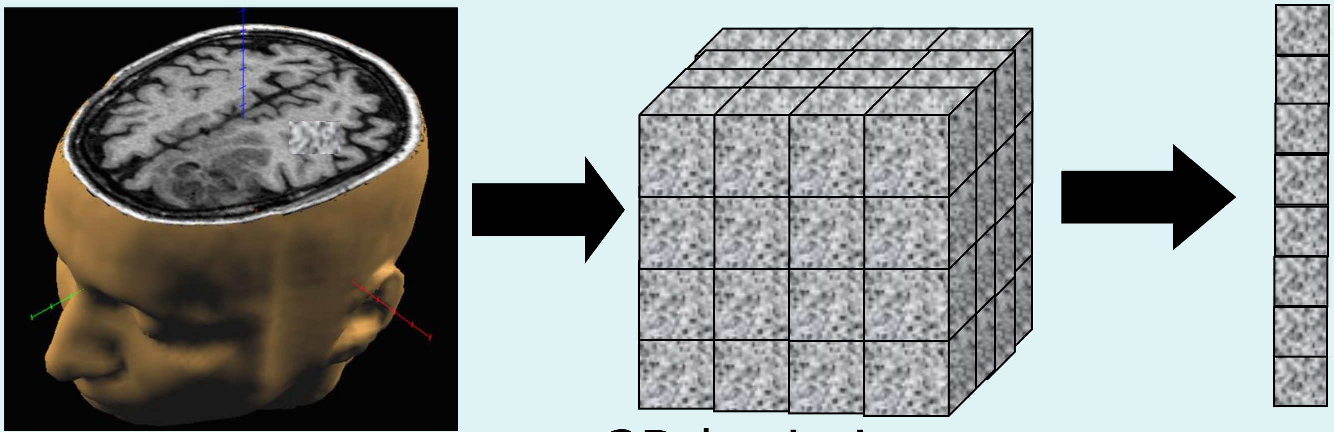
Multivariate approach, aka. "pattern recognition"



Find the mapping h from observed data Y to explanatory variable X

$h: Y \rightarrow X$

Neuroimaging data



3D brain image

“feature vector” or “data point”

Data dimensions

- dimensionality of a “data point” = #voxels considered
- number of “data point” = #scans/images considered

Note that #voxels >> #scans !

➔ “ill posed problem”

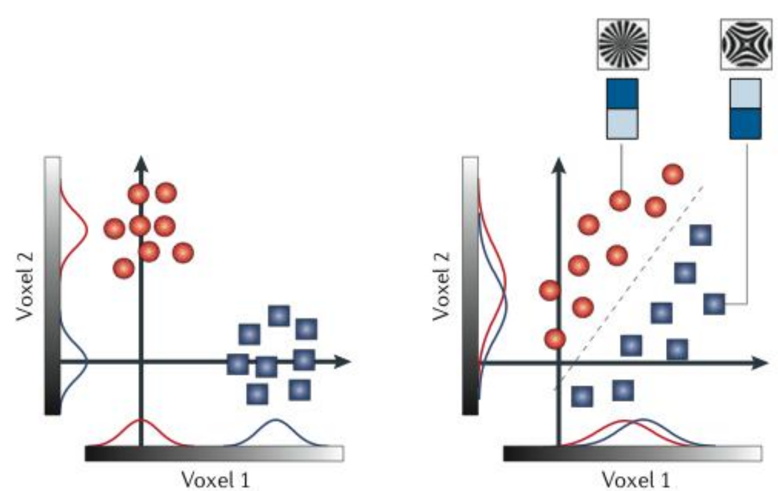
Advantages of pattern recognition

Accounts for the spatial correlation of the data (multivariate aspect)

- images are multivariate by nature.
- can yield greater sensitivity than conventional (univariate) analysis.

Enable classification/prediction of individual subjects

- ‘Mind-reading’ or decod
- Clinical application



Haynes & Rees, 2006

Pattern recognition framework

Input (brain scans) → No mathematical model available → Output (control/patient)

$\mathbf{X}_1$ → y_1

$\mathbf{X}_2$ → y_2

$\mathbf{X}_3$ → y_3

Machine Learning Methodology

Computer-based procedures that learn a function from a *series* of examples

Training Examples: $(\mathbf{X}_1, y_1), \ldots, (\mathbf{X}_s, y_s)$ → **Learning/Training Phase**

Generate a function or classifier f such that

$f(\mathbf{x}_i) \rightarrow y_i$

→ f

Test Example $\mathbf{X}_i$ → **Testing Phase**

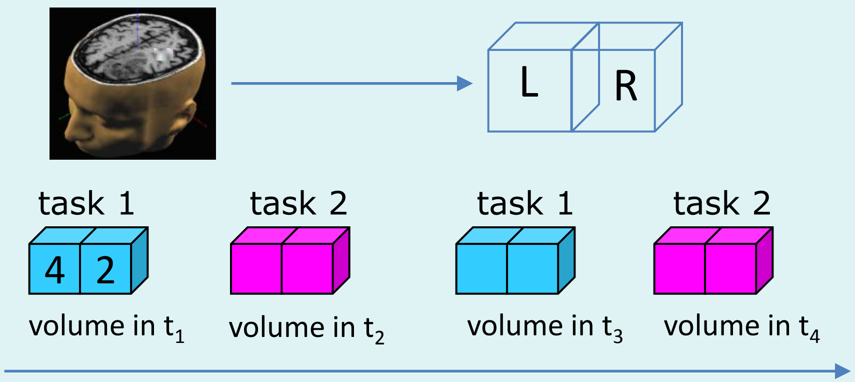
Prediction

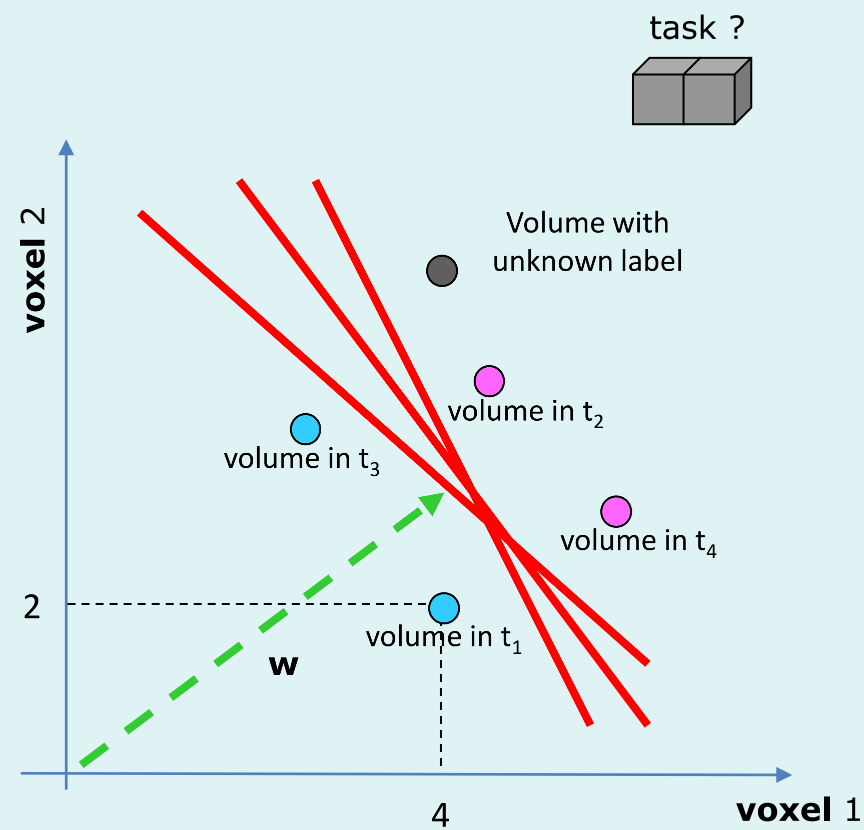
→ $f(\mathbf{X}_i) = y_i$

Overview

- Introduction
  - Univariate & multivariate approaches
  - Data representation
- Pattern Recognition
  - Machine learning
  - Validation & inference
  - Weight maps & feature selection
  - fMRI application
  - Multiclass problem
- Conclusion & PRoNTo

Classification example

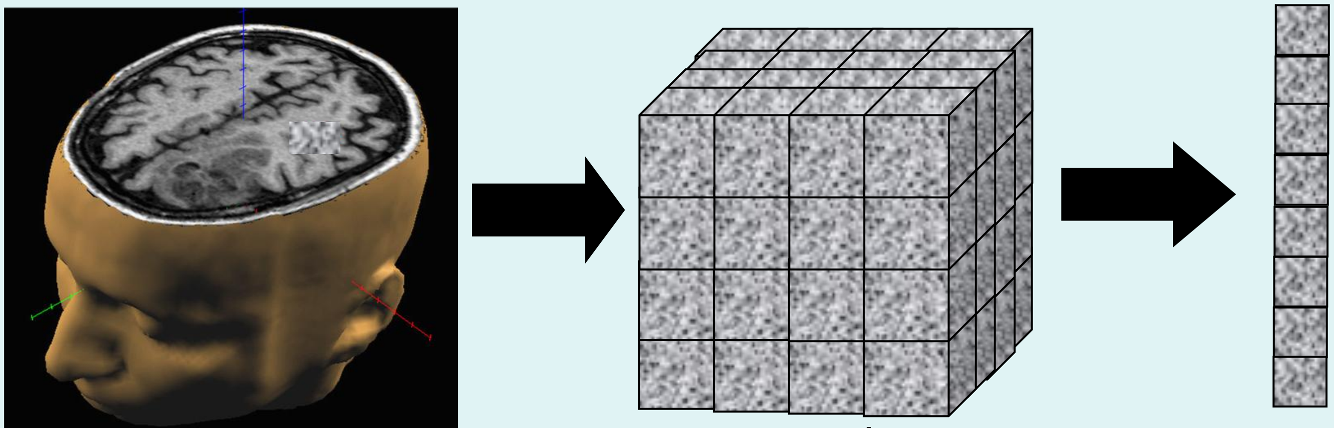




Different classifiers will compute different hyperplanes!

Note: task1/2 ~ disease/controle

Neuroimaging data



Problem: 1000’s of features vs. 10’s of data points

Possible solutions to dimensionality problem:

- Feature selection strategies (e.g. ROIS, select only activated voxels)
- (Searchlight)
- **Kernel Methods**

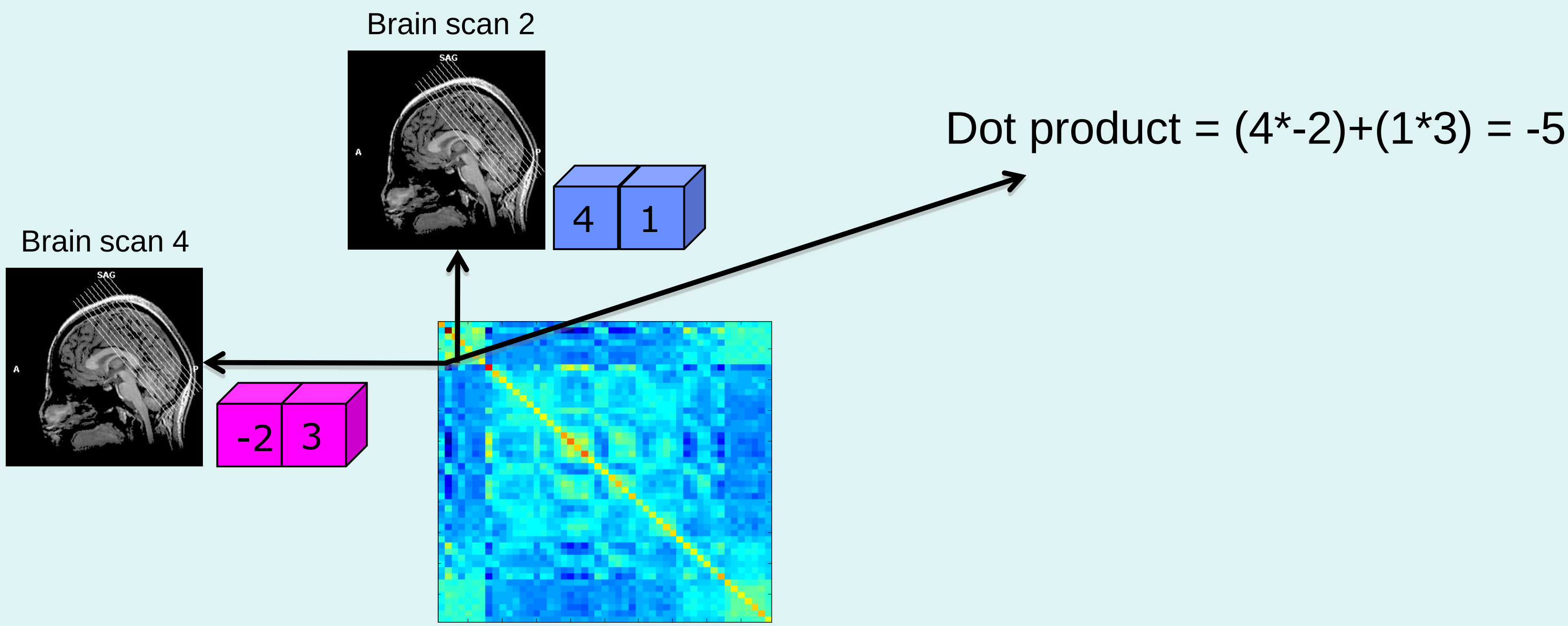
Kernel approaches

- Mathematical trick! ➔ powerful and unified framework (e.g. classification & regression)
- Consist of two parts:
  - build the kernel matrix (mapping into the feature space)
  - train using the kernel matrix (designed to discover linear patterns in the feature space)
- Advantages:
  - computational shortcut ➔ represent linear patterns efficiently in high dimensional space.
  - Using the dual representation with proper regularization ➔ efficient solution of ill-conditioned problems.
- Examples ➔ Support Vector Machine (SVM), Gaussian

  Processes (GP), Kernel Ridge Regression (KRR),…

Kernel matrix

Kernel matrix = similarity measure



The “kernel function”

- 2 patterns **x** and **x*** ➔ a real number characterizing their similarity (~distance measure).
- simple similarity measure = a dot product ➔linear kernel.

Linear classifier

- hyperplanes through the feature space
- parameterized by
  - a weight vector **w** and
  - a bias term *b*.
- weight vector **w** = linear combination of training examples $\mathbf{x}_i$ (where *i* = 1,…,N and N is the number of training examples)

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i$$

➔ Find the α_i !!!

Linear classifier prediction

General equation: making predictions for a test example $\mathbf{x}_*$ with kernel methods

$$\mathbf{w} = \sum_{i=1}^{N} a_i \mathbf{x}_i$$

$$f(\mathbf{x}_*) = \mathbf{w} \cdot \mathbf{x}_* + b$$ Primal representation

$$f(\mathbf{x}_*) = \sum_{i=1}^{N} a_i \mathbf{x}_i \cdot \mathbf{x}_* + b$$

kernel definition

$$f(\mathbf{x}_*) = \sum_{i=1}^{N} a_i K(\mathbf{x}_i, \mathbf{x}_*) + b$$ Dual representation

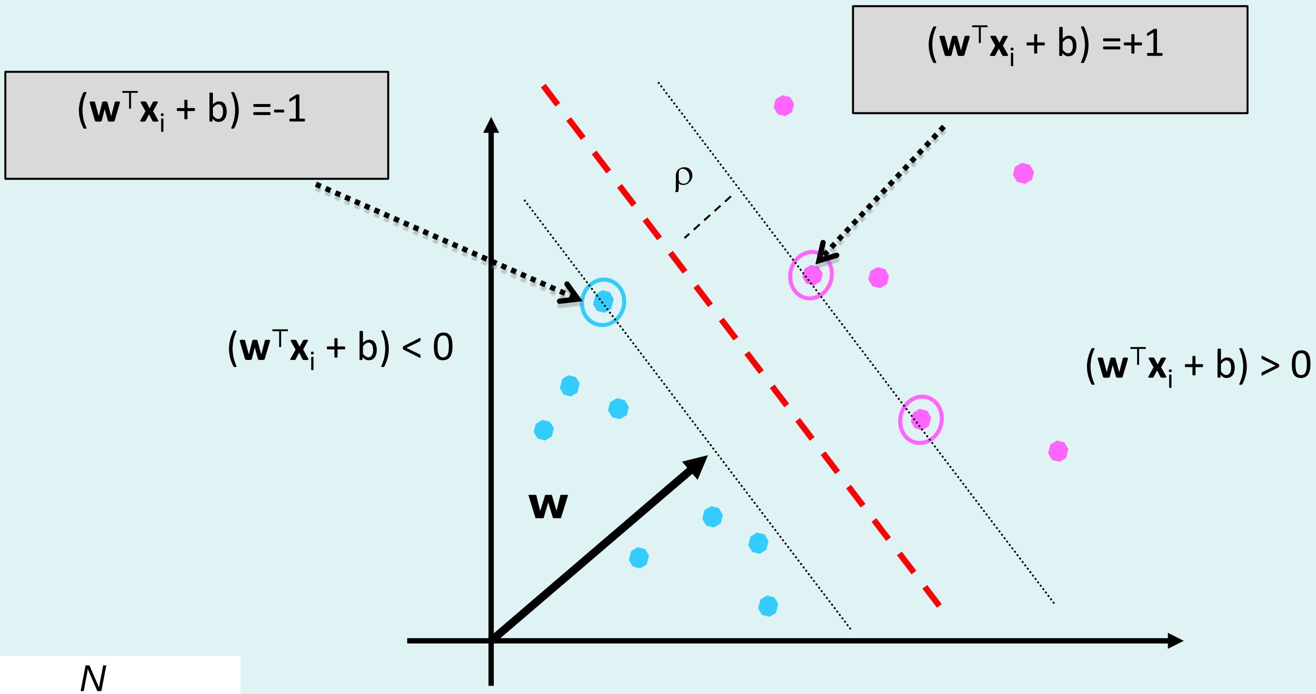
$f(\mathbf{x}_*)$ =

signed distance to boundary (classification)

predicted score (regression)

Support Vector Machine

SVM = "maximum margin" classifier

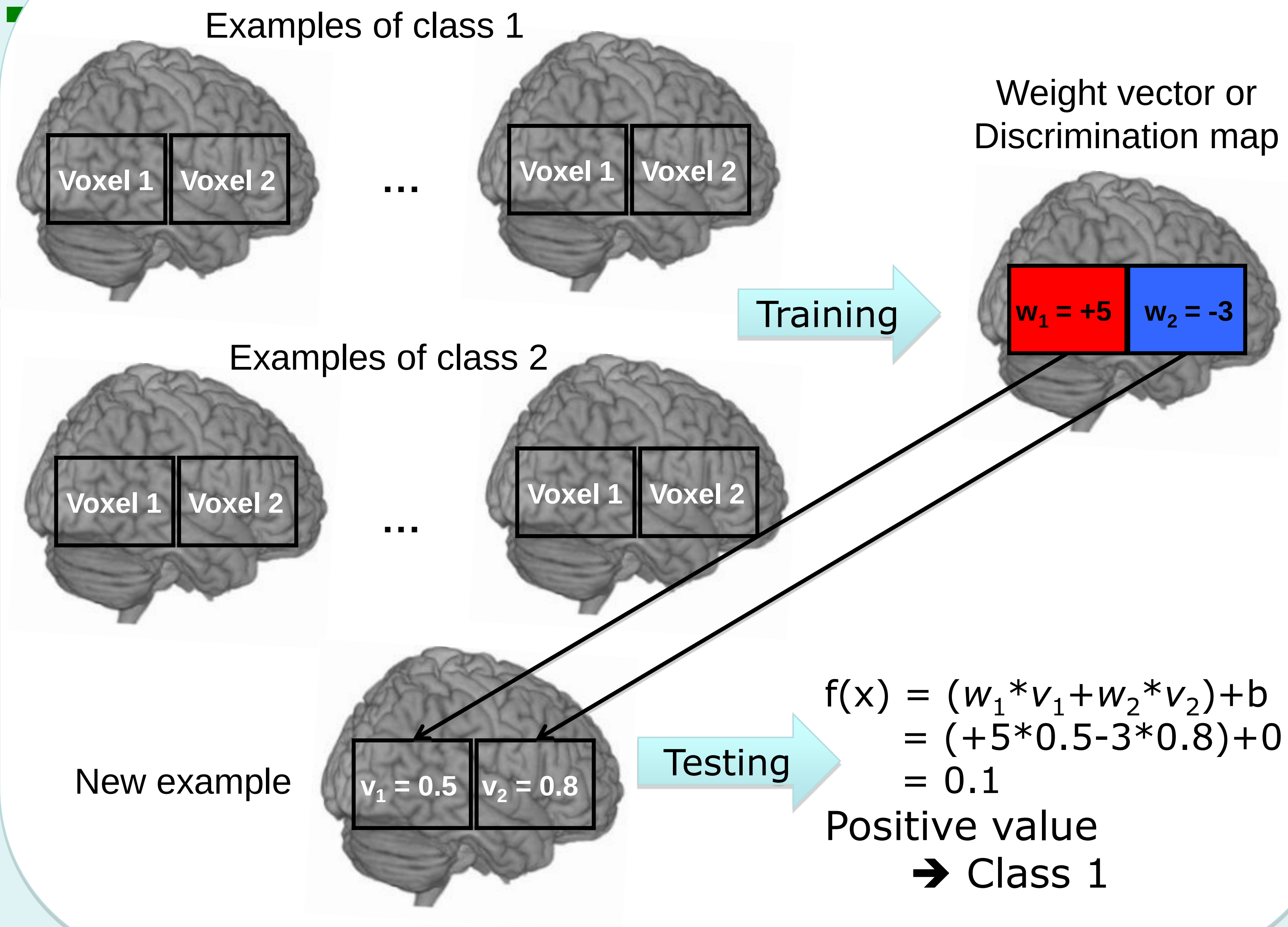


$$\mathbf{w} = \sum_{i=1}^{N} a_i \mathbf{x}_i$$

Support vectors have $\alpha_i \neq 0$

Data: $<\mathbf{x}_i, y_i>$, $i=1,..,N$
Observations: $\mathbf{x}_i \in R^d$
Labels: $y_i \in \{-1,+1\}$

Illustrative example: Classifiers as decision functions



SVM vs. GP

SVM

➔ Hard binary classification

- simple & efficient, quick calculation but
- NO ‘grading’ in output {-1, 1}

Gaussian Processes

➔ probabilistic model

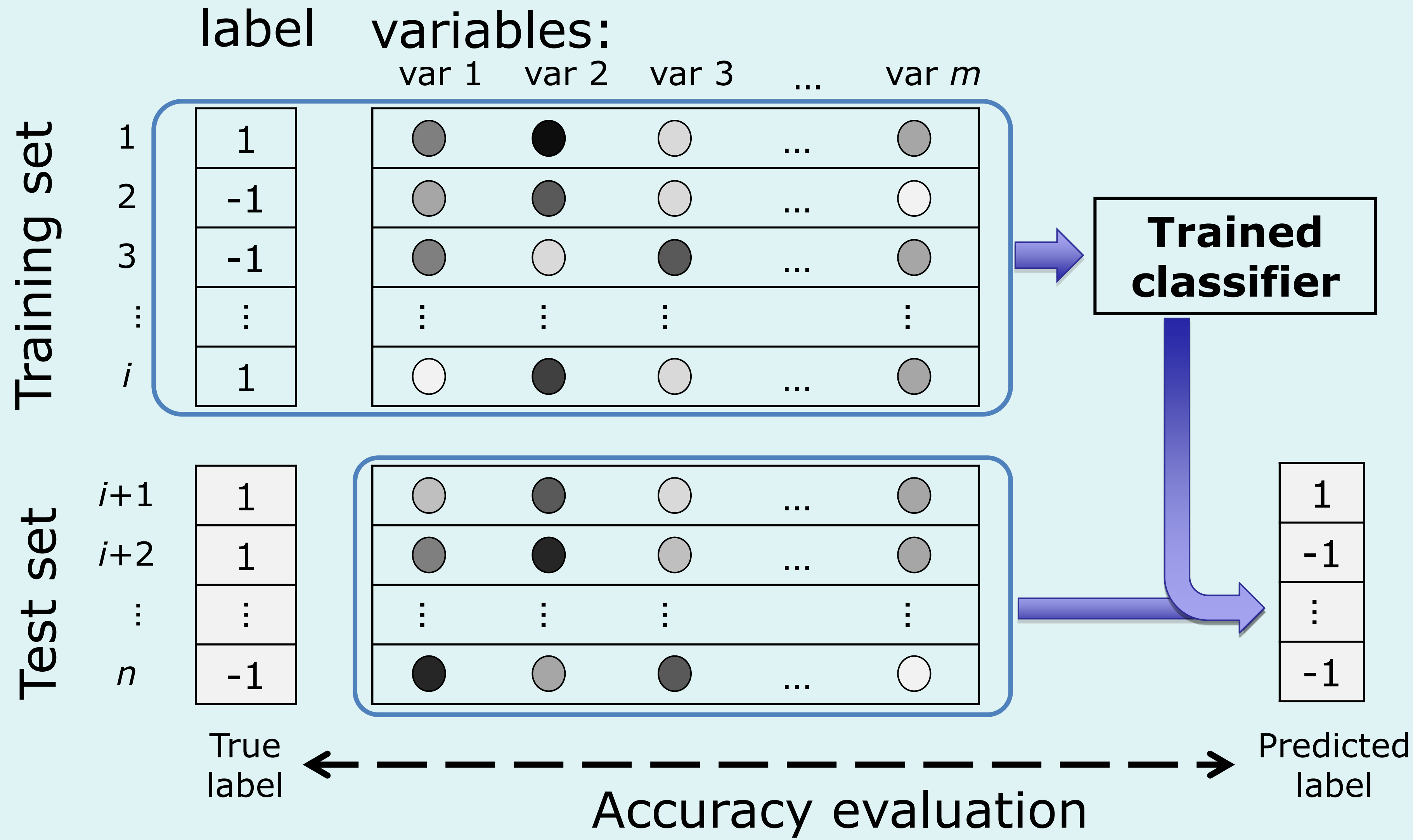
- more complicated, slower calculation but
- returns a probability [0 1]
- can be multiclass

Overview

- Introduction
  - Univariate & multivariate approaches
  - Data representation
- Pattern Recognition
  - Machine learning
  - Validation & inference
  - Weight maps & feature selection
  - fMRI application
  - Multiclass problem
- Conclusion & PRoNTo

Validation principle

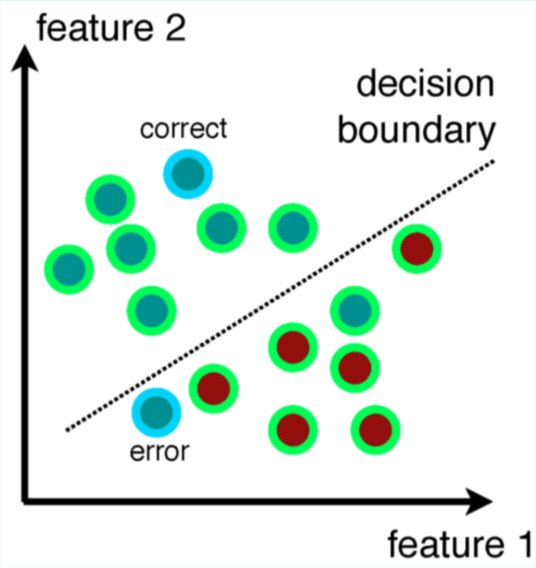
Samples



M-fold cross-validation

- Split data in 2 sets: “train” & “test”
  ➔ evaluation on 1 “fold”

- Rotate partition and repeat
  ➔ evaluations on M “folds”



- Applies to scans/events/blocks/subjects/…
  ➔ Leave-one-out (LOO) approach

Confusion matrix & accuracy

		$\hat{\omega}$	
		ω_0	ω_1
truth	ω_0	A	B
	ω_1	C	D

Confusion matrix

= summary table

Accuracy estimation

- Class 0 accuracy, p_0 = A/(A+B)
- Class 1 accuracy, p_1 = D/(C+D)
- Accuracy, p = (A+D)/(A+B+C+D)

Other criteria

- Positive Predictive Value, PPV = D/(B+D)
- Negative Predictive Value, NPV = A/(A+C)

Accuracy & Dataset balance

Watch out if #samples/class are different!

<u>Example:</u>

Good overall accuracy (72%) but

- Majority class (N_1 = 80), excellent accuracy (90%)
- Minority class (N_2 = 20), poor accuracy (0%)

<u>Good practice:</u>

Report

- class accuracies $[p_0, p_1, \ldots, p_C]$
- balanced accuracy $p_{bal} = (p_0 + p_1 + \ldots + p_C)/C$

Regression MSE

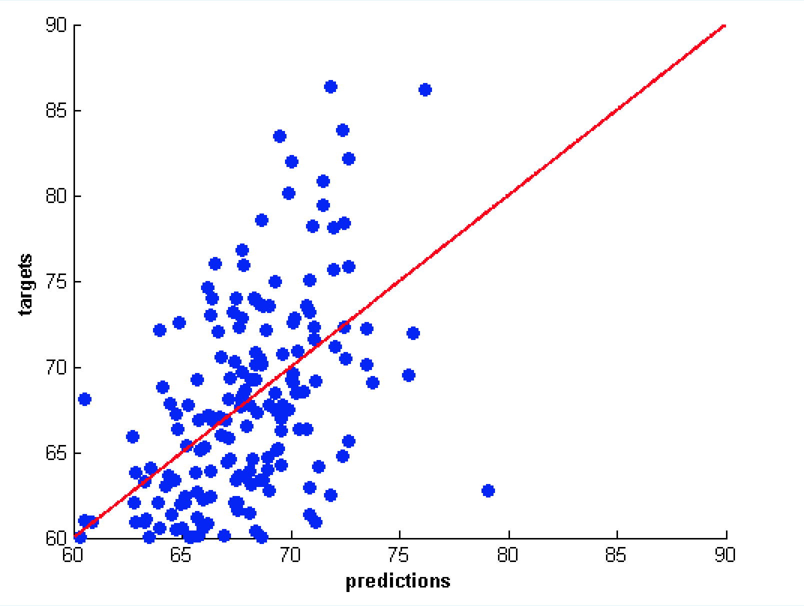
- LOO error in one fold $SE_n = (y_n - f(\mathbf{x}_n))^2$
- Across all LOO folds

$$R(f, \mathbf{X}) = MSE = \frac{1}{N} \sum_{n=1}^{N} (y_n - f(\mathbf{x}_n))^2$$

➔ Out-of-sample "mean squared error" (MSE)

Other measure:

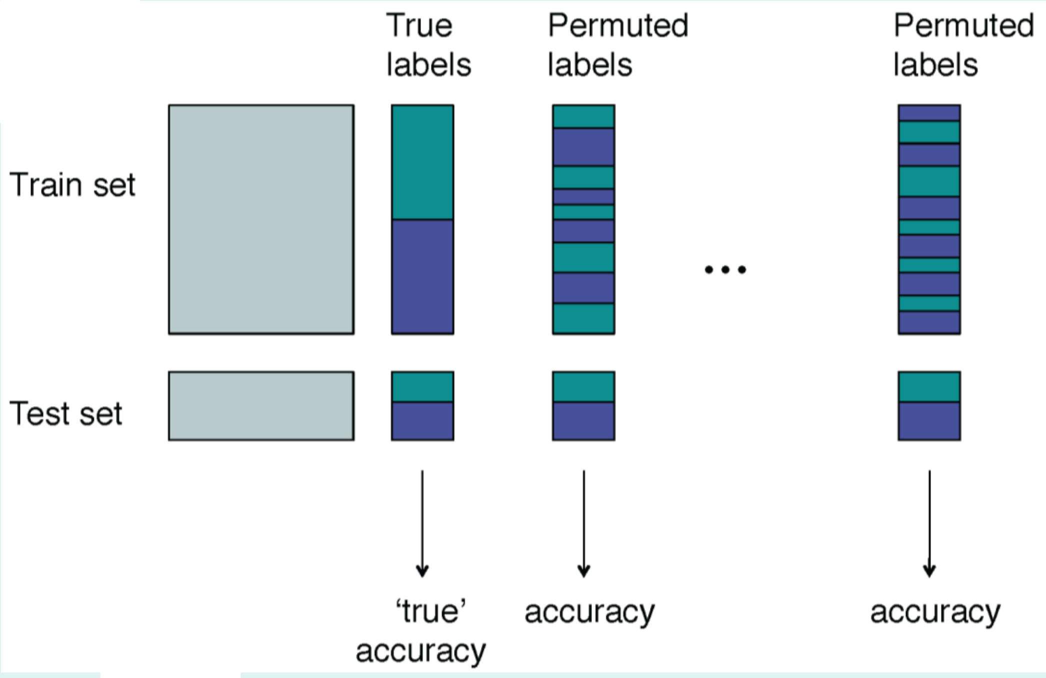
Correlation between predictions (across folds!) and 'true' targets



Inference by permutation testing

- H_0: “class labels are non-informative”
- Test statistic = CV accuracy
- Estimate distribution of test statistic under H0
  - ➔ Random permutation of labels
  - ➔ Estimate CV accuracy
  - ➔ Repeat M times
- Calculate p-value as

$$\frac{1}{M}\sum_{m}^{M}(p_m^{perm} \geq p^{real})$$



Overview

- Introduction
  - Univariate & multivariate approaches
  - Data representation
- Pattern Recognition
  - Machine learning
  - Validation & inference
  - Weight maps & feature selection
  - fMRI application
  - Multiclass problem
- Conclusion & PRoNTo

Weight vector interpretation

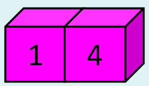
Weight vector

➔ weight (or discrimination) image !

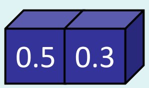
➔ how important each voxel is

➔ for which class “it votes” (mean centred data & b=0)

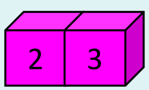
task1



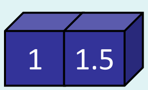
task2



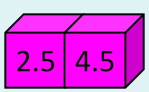
task1



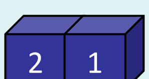
task2

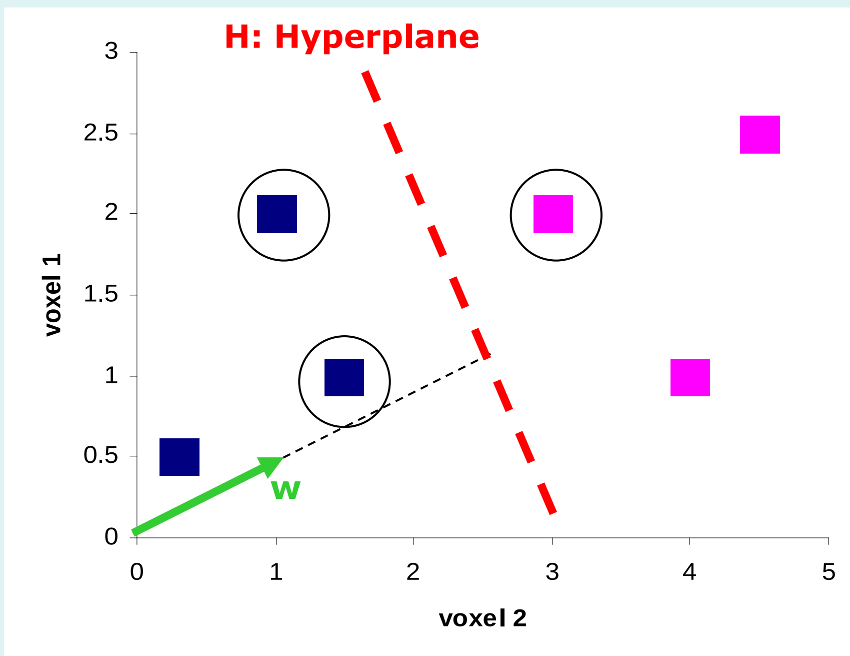


task1

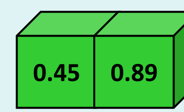


task2



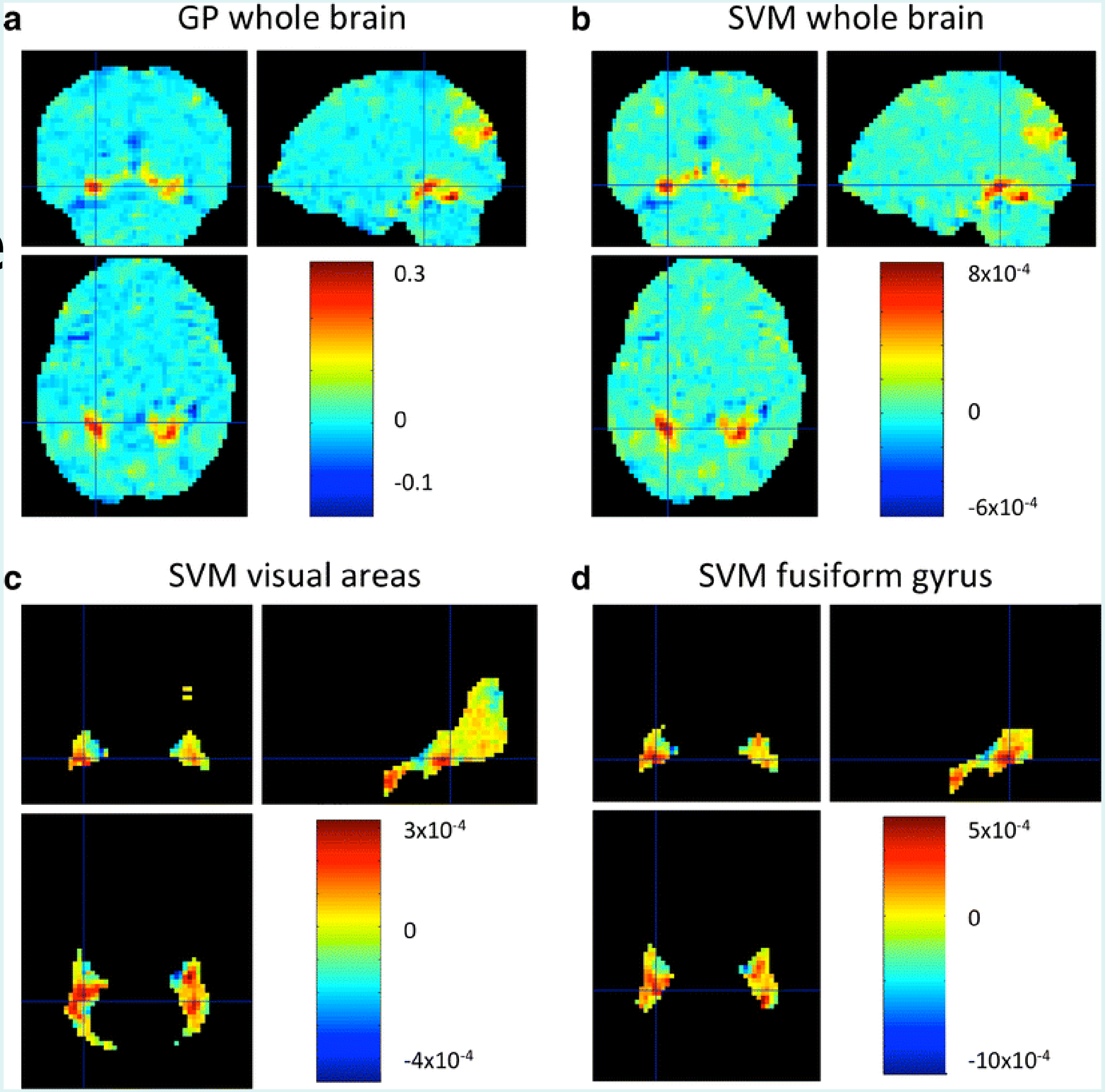


Weight vector
W = [0.45 0.89]
b = -2.8

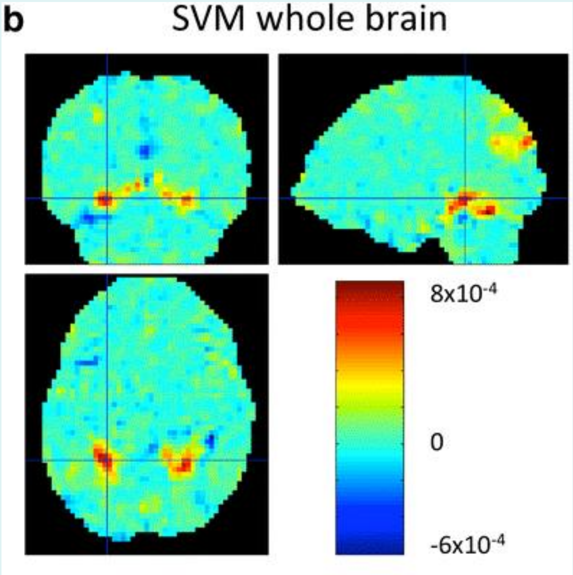


Example of masks

Linear machine
→ Weight map



Feature selection



- 1 sample image
  ➔ 1 predicted value
- use ALL the voxels
  ➔ NO thresholding of weight allowed!

Feature selection:

- a priori mask
- a priori 'filtering'
- recursive feature elimination/addition
  ➔ nested cross-validation
  (MUST be independent from test data!)

Overview

- Introduction
  - Univariate & multivariate approaches
  - Data representation
- Pattern Recognition
  - Machine learning
  - Validation & inference
  - Weight maps & feature selection
  - fMRI application
  - Multiclass problem
- Conclusion & PRoNTo

fMRI designs

Level of inference

- within subject ≈ FFX with SPM

➔ ‘decode’ subject’s brain state

- between subjects ≈ RFX with SPM

➔ ‘classify’ groups, or

regress subjects’ parameter

Between subjects

Design

- 2 groups: group A vs. group B
- 1 group: 2 conditions per subject

➔ Extract 1 (or 2) summary image(s) per subject, and classify

Leave-one-out (LOO) cross-validation:

- Leave one subject out (LOSO)
- Leave one subject per group out (LOSGO)

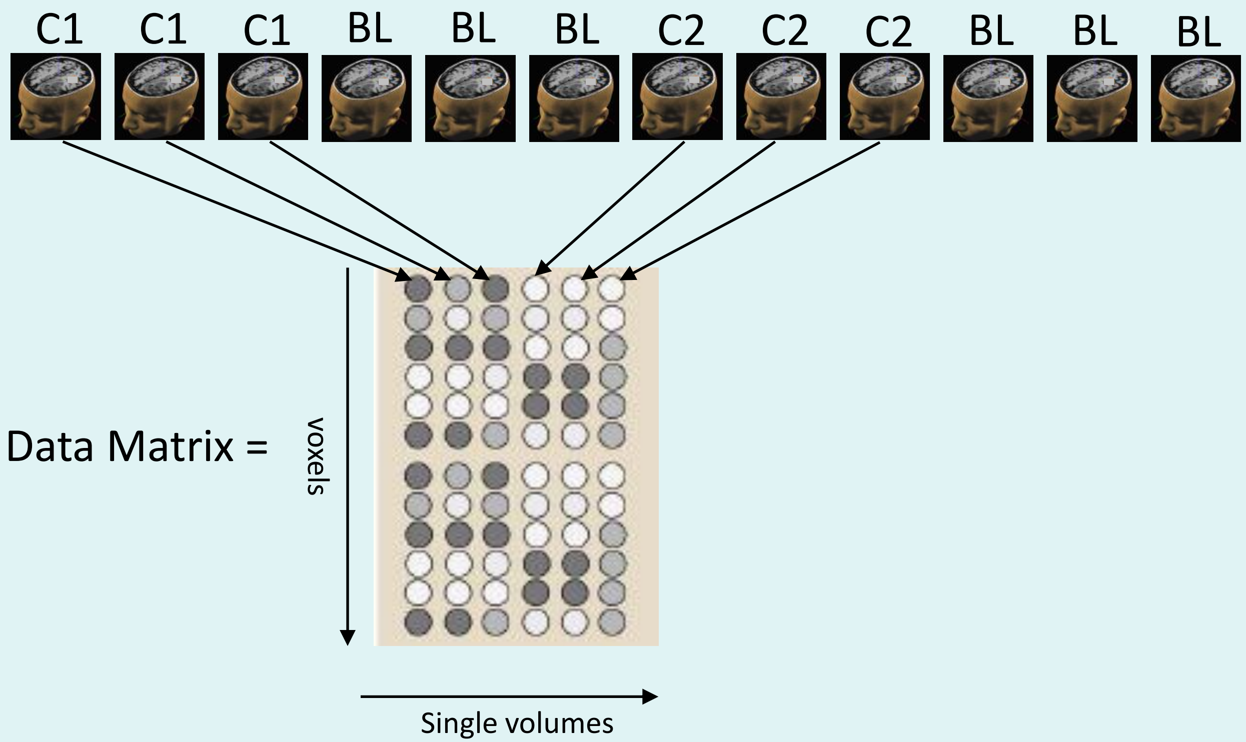
Note: this works for any type of image…

Within subject

Design:

- Block or event-related design
- Accounting for haemodynamic function

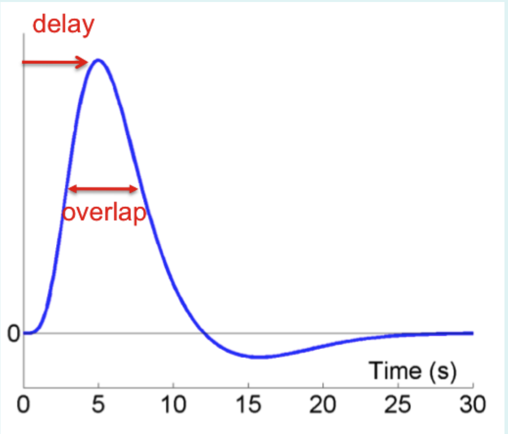
Use single scans

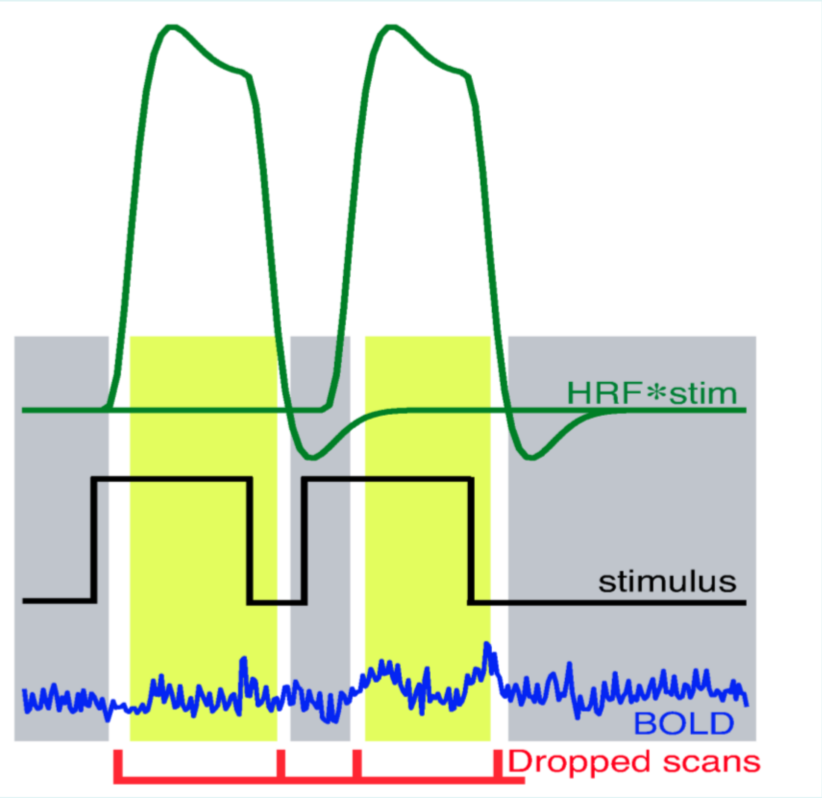


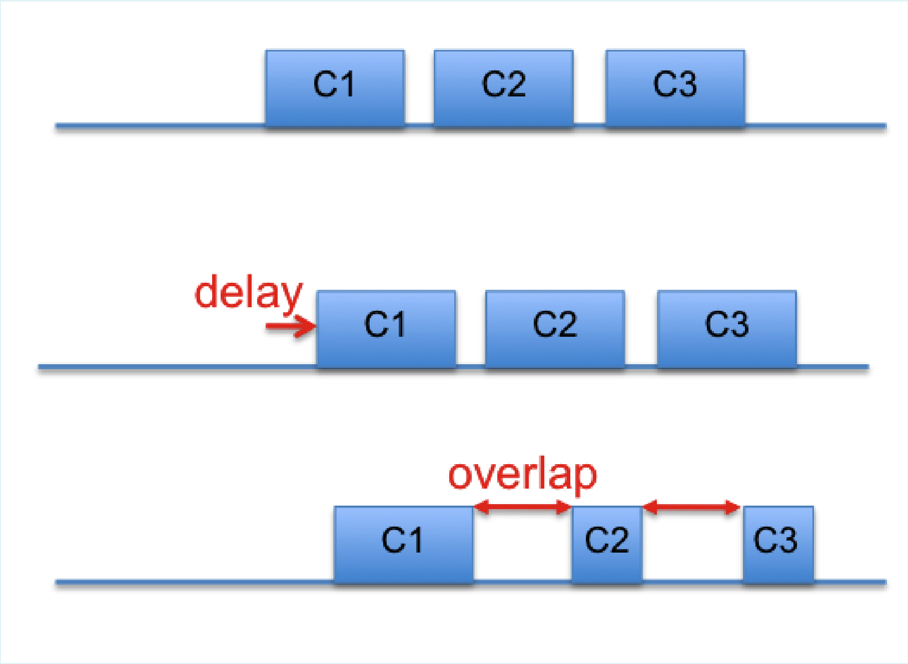
Within subject

Design:

- Block or event-related design
- Accounting for haemodynamic function





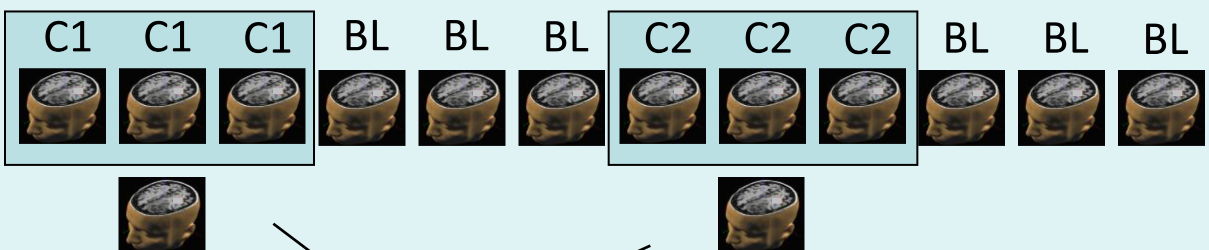


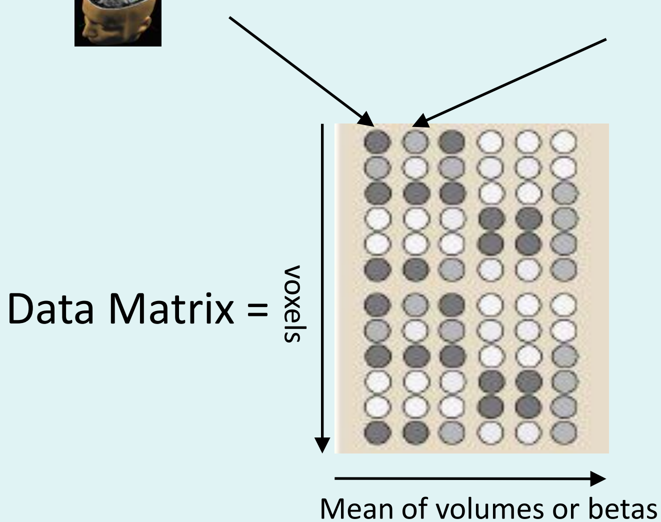
Within subject

Design:

- Block or event-related design
- Accounting for haemodynamic function

Averaging/deconvolution





How to?

- Average scans over blocks/events
- Parameter estimate from the GLM with 1 regressor per block/event

Within subject

Design:

- Block or event-related design
- Accounting for haemodynamic function

Leave-one-out (LOO) cross-validation:

- Leave one session/run out
- Leave one block/event out

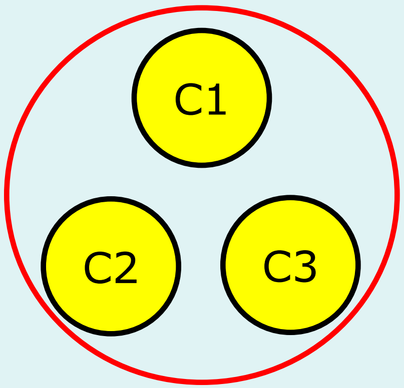
  (danger of dependent data!!!)

Overview

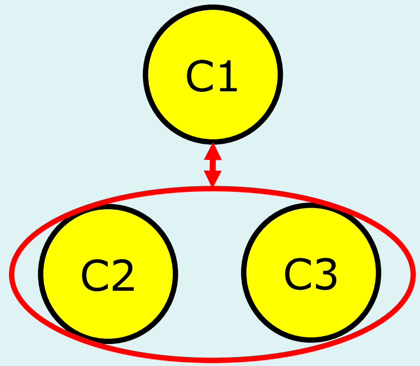
- Introduction
  - Univariate & multivariate approaches
  - Data representation
- Pattern Recognition
  - Machine learning
  - Validation & inference
  - Weight maps & feature selection
  - fMRI application
  - Multiclass problem
- Conclusion & PRoNTo

Multiclass problem

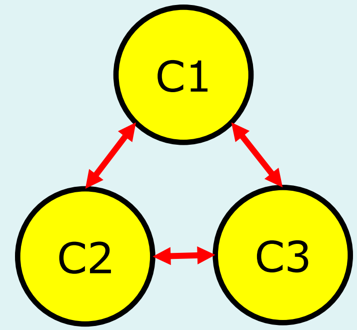
Multiclass machine



Binary machine & one-vs.-others



Binary machine & one-vs.-one



ECOC	**SVM codewords**			
	C1-C2	C1-C3	C2-C3	L
C1	1	1	0	3
C2	-1	0	1	2
C3	0	-1	-1	1
Example	-1	-1	-1	C3

"Error-Correcting Output Coding" (ECOC) approach

Overview

- Introduction
  - Univariate & multivariate approaches
  - Data representation
- Pattern Recognition
  - Machine learning
  - Validation & inference
  - Weight maps & feature selection
  - fMRI application
  - Multiclass problem
- Conclusion & PRoNTo

Conclusions

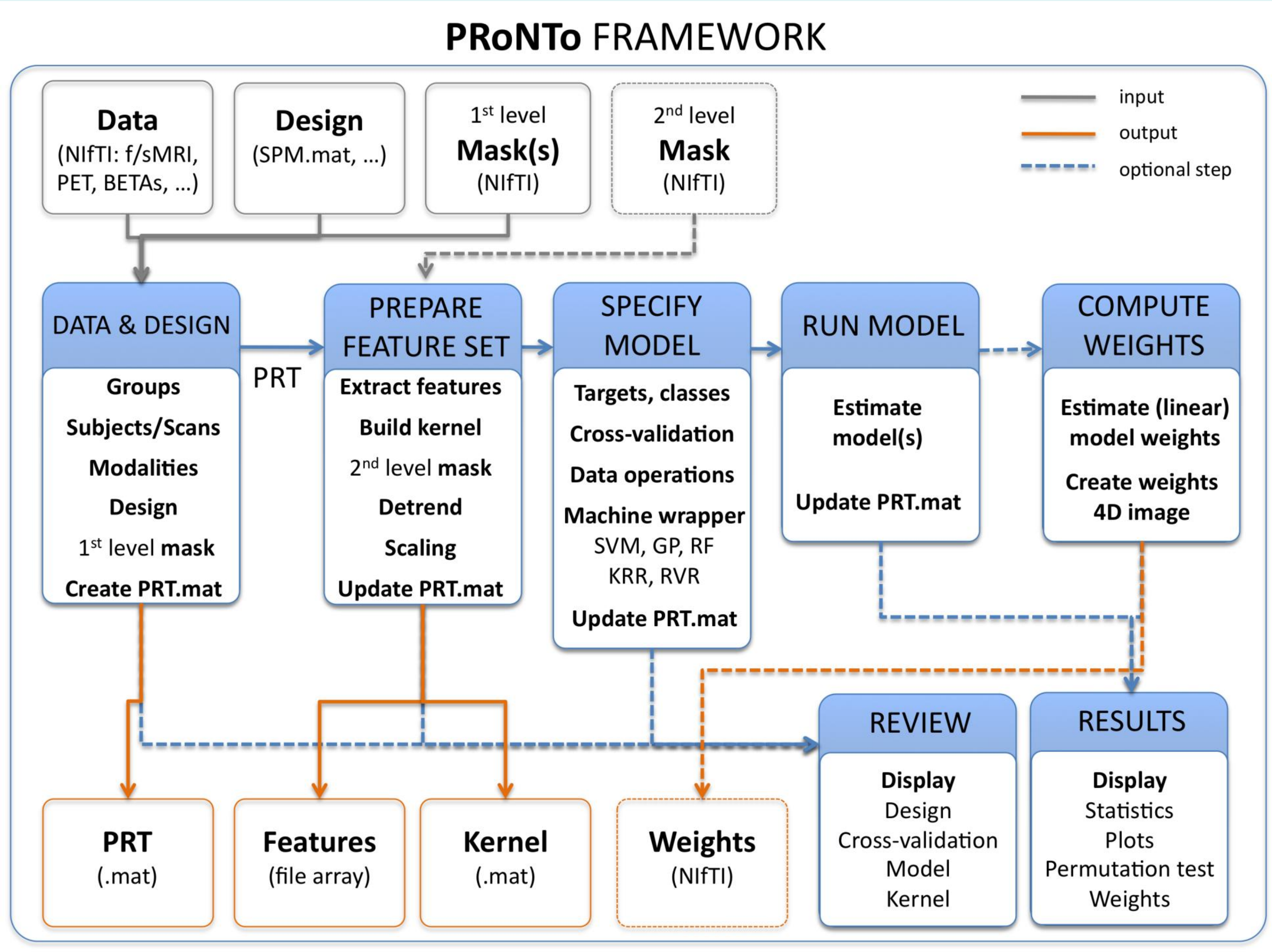
Key points:

- More sensitivity (~like omnibus test with SPM)
- NO local (voxel/blob) inference

  ➔ CANNOT report coordinates nor thresholded weight map
- Require cross-validation (split in train/test sets)

  ➔ report accuracy/PPV (or MSE)
- MUST assess significance of accuracy

  ➔ permutation approach

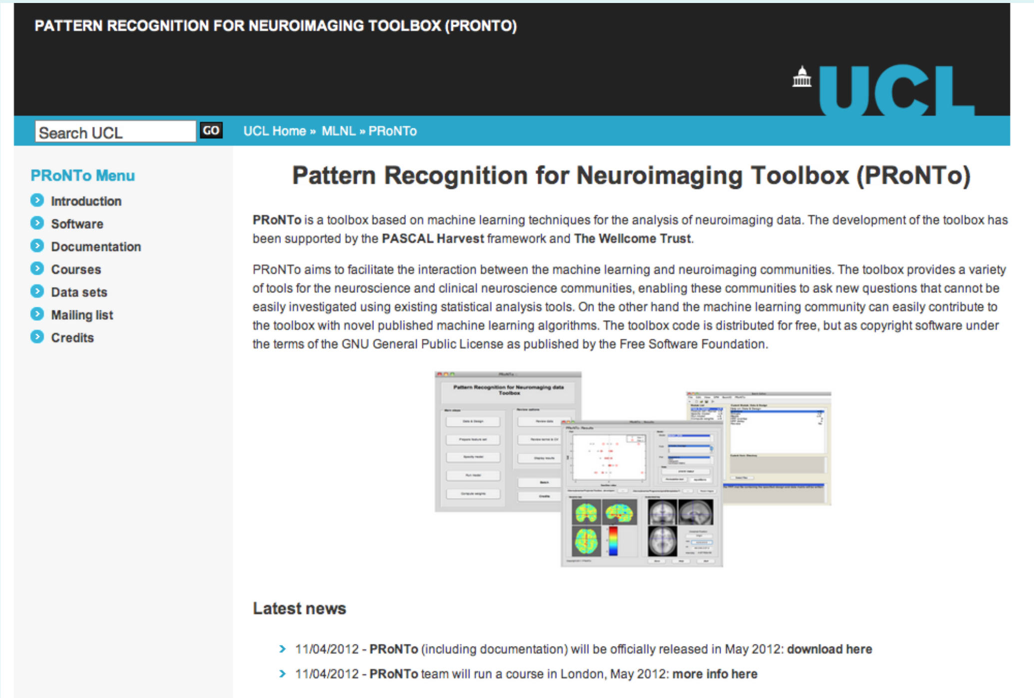

PRoNTo FRAMEWORK
Data
(NIfTI: f/sMRI, PET, BETAs, ...)
Design
(SPM.mat, ...)
1st level
Mask(s)
(NIfTI)
2nd level
Mask
(NIfTI)
input
output
optional step
DATA & DESIGN
Groups
Subjects/Scans
Modalities
Design
1st level mask
Create PRT.mat
PRT
PREPARE FEATURE SET
Extract features
Build kernel
2nd level mask
Detrend
Scaling
Update PRT.mat
SPECIFY MODEL
Targets, classes
Cross-validation
Data operations
Machine wrapper
SVM, GP, RF
KRR, RVR
Update PRT.mat
RUN MODEL
Estimate model(s)
Update PRT.mat
COMPUTE WEIGHTS
Estimate (linear) model weights
Create weights 4D image
REVIEW
Display
Design
Cross-validation
Model
Kernel
RESULTS
Display
Statistics
Plots
Permutation test
Weights
PRT
(.mat)
Features
(file array)
Kernel
(.mat)
Weights
(NIfTI)

PRoNTo

"Pattern Recognition for Neuroimaging Toolbox", aka. PRoNTo :

http://www.mlnl.cs.ucl.ac.uk/pronto/

with references, manual, demo data, course, etc.



Paper: http://dx.doi.org/10.1007/s12021-013-9178-1

Thank you for your attention!

Any question?

Thanks to the PRoNTo Team for the borrowed slides. ☺