Machine Learning in NeuroImaging

Javier González-Castillo

Section on Functional Imaging Methods, NIMH, NIH

August 2017, National Institutes of Health, Bethesda, MD

Materials used in this presentation/Where to go next

Andrew Ng, Stanford University

Machine Learning
Stanford University

Ended Mar 06

Go to Course

Purchase Course

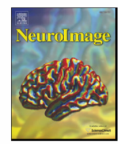
coursera

NeuroImage 45 (2009) S199–S209

Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg



Machine learning classifiers and fMRI: A tutorial overview

Francisco Pereira [a,*], Tom Mitchell [b], Matthew Botvinick [a]

[a] Princeton Neuroscience Institute/Psychology Department, Princeton University, Princeton, NJ 08540, USA
[b] Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Neuroscience & Biobehavioral Reviews

Volume 74, Part A, March 2017, Pages 58-75

Review article

Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications

Sandra Vieira [a], Walter H.L. Pinaya [b], Andrea Mechelli [a]

frontiers in NEUROINFORMATICS

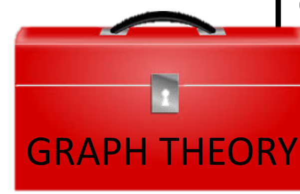
METHODS ARTICLE
published: 21 February 2014
doi: 10.3389/fninf.2014.00014

Machine learning for neuroimaging with scikit-learn

Alexandre Abraham[1,2], Fabian Pedregosa[1,2], Michael Eickenberg[1,2], Philippe Gervais[1,2], Andreas Mueller[3], Jean Kossaifi[4], Alexandre Gramfort[1,2,5], Bertrand Thirion[1,2] and Gaël Varoquaux[1,2]*

Introduction / Agenda

EXPERIMENTAL DESIGN → DATA ACQUISITION → DATA PRE-PROCESSING → GLM – ACTIVATION / FUNCTIONAL CONNECTIVITY



Centrality, Degree, Clustering Coefficient, Community, etc.

Logistic Regression, Support Vector Machines, ICA, K-Means, Convolutional Networks, etc.

Cost Function, Learning Rate, Gradient Descend, Decision Boundary, Regularization, etc.

- A few applications to fMRI data.
- A few words on software.
- Additional Resources to learn more.

What is Machine Learning?

Field of study that gives computers the ability to learn without being explicitly programmed. *[Samuel, 1959]*

A computer is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improved with experience E. *[Mitchell, 1998]*

SUPERVISED LEARNING
Algorithms that require ground truth during training

UNSUPERVISED LEARNING
Algorithms whose input has no labels/true values, and whose objective is to find hidden structure in the data

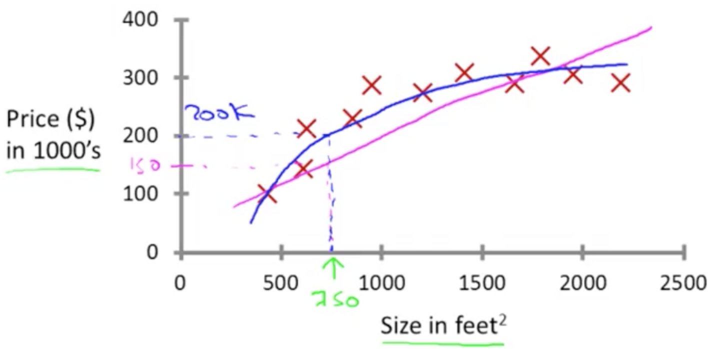
<u>Problems that ML can solve:</u>

- Predict an integer rating
- Predict a label (out of a limited set)
- Discover structure in the data, e.g., groups
- Reduce the dimensionality of the data
- Anomaly detection

Supervised Learning

Algorithms used to draw inferences from labeled datasets

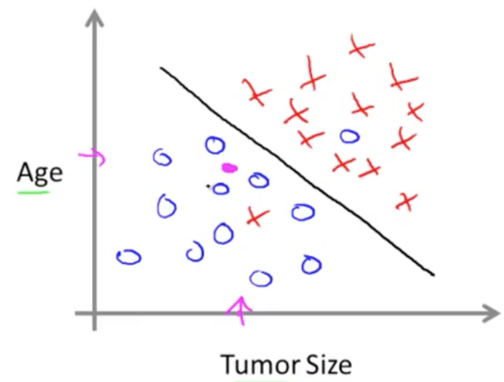
REGRESSION	CLASSIFICATION
Predict a Continuous Variable	Predict a Discrete Variable



$f_w(x_1,..x_n)$ = Real Number

Independent Variables

Dependent Variable



$f_w(x_1,..x_n)$ = Class ID

Features

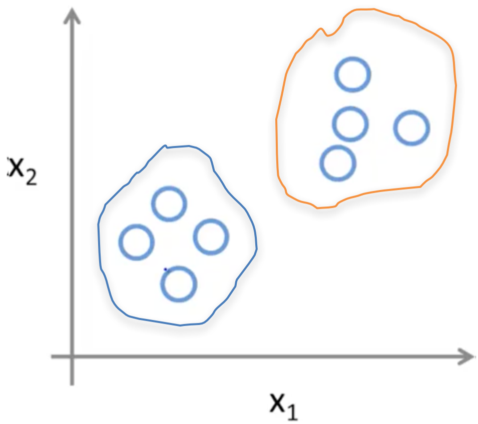
Label

- The Independent variables/features can be voxel intensity, connectivity values, etc.
- Regression:
  - Dependent Variable can be a behavioral or psychiatric score, etc.
- Classification:
  - Dependent Variable can be a task type, stimulus type, a patient group, etc.

Unsupervised Learning

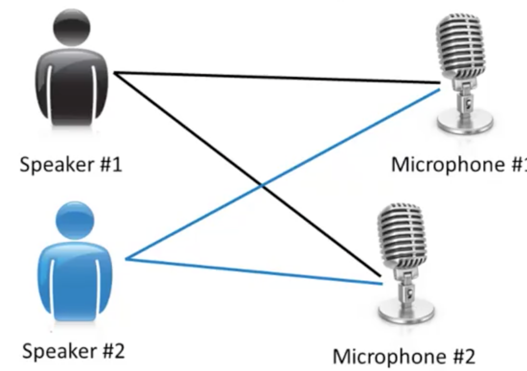
Algorithms used to draw inferences from unlabeled datasets

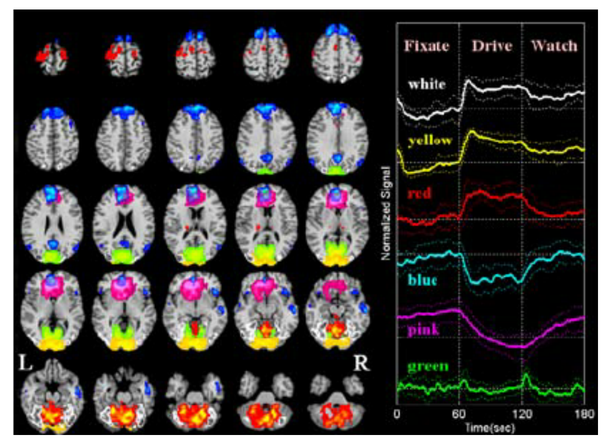
CLUSTERING ALGORITHMS



K-Means, Fuzzy K-means,
Hierarchical Clustering, DBSCAN, …

SOURCE SEPARATION

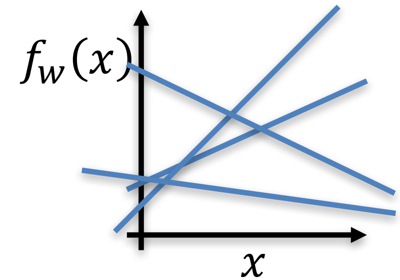




PCA, ICA, SVD, …

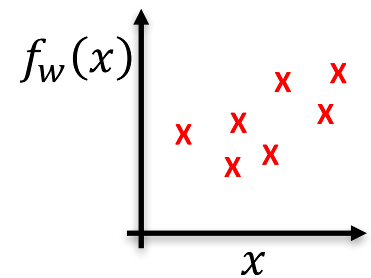
Univariate Linear Regression in Machine Learning Terms

x	$f_w(x)$	y
Features	→	Class Label/Real Value



MODEL: Univariate Linear Regression

$$f_w(x) = w_0 + w_1 x$$

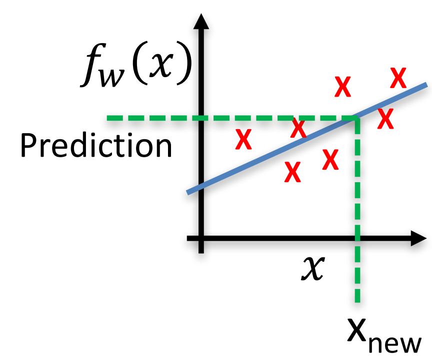


TRAINING SET:

Voxel Amplitude (x)	Test Score (y)
2104	460
1416	232
1534	315
852	178
…	…

TRAINING:
Obtaining w_0 and w_1 so that the line "fits the data well"
We need a way to measure "how well" → Cost Function

PREDICTING:
Apply $f_w(x)$ to new data



Cost Function

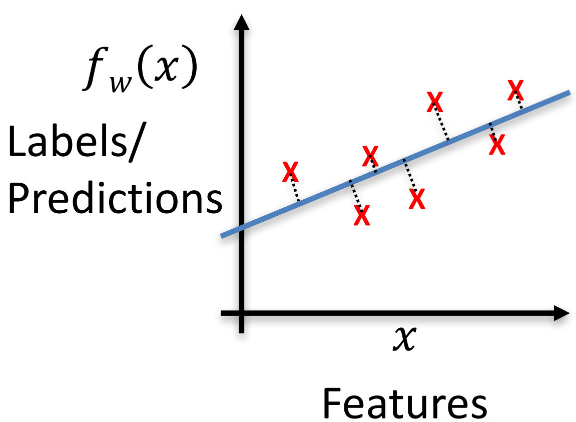
x	$f_w(x)$, $J(w)$	y
Features	→	Class Label/Real Value

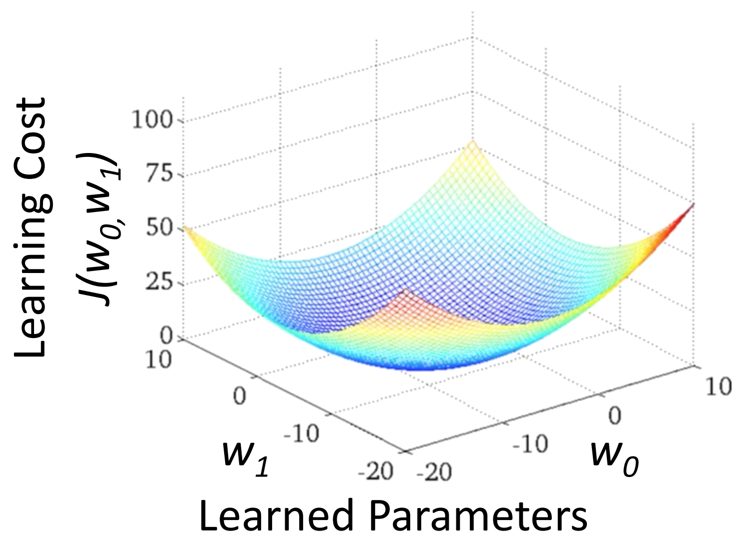
Choose w_0, w_1 so that $f_w(x)$ is close to y for all training examples

Objective Function: $$\min_{w_0, w_1} \frac{1}{2m} \sum_{i=1}^{i=m} \left(f\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Cost Function: $$J(w_o, w_1) = \frac{1}{2m} \sum_{i=1}^{i=m} \left(f\left(x^{(i)}\right) - y^{(i)}\right)^2$$

Squared Error Function





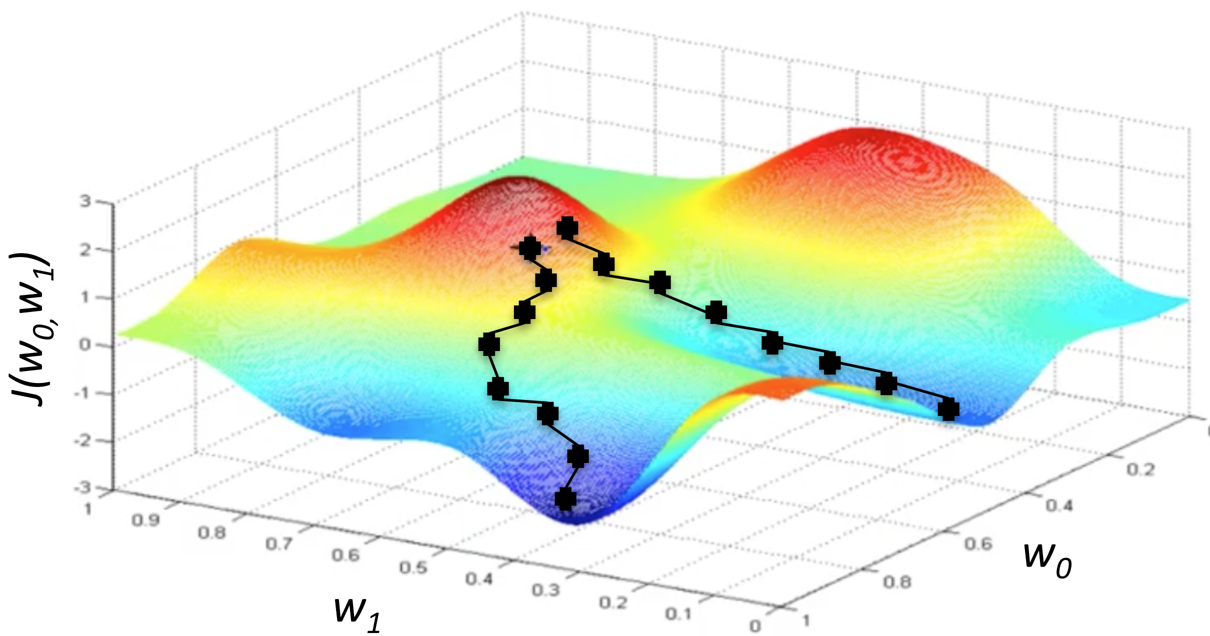
Gradient Descend (how to learn)

In this context, learning means: to find *[w_o, …, w_n]* that minimizes our cost function *J(w)*

$$\min_{w_0, w_1} J(w_0, w_1)$$

One algorithm to do such learning is GRADIENT DESCEND:

- Start with some random values of w_0 and w_1
- Keep changing them, until we find the minimum of $J(w_0, w_1)$



Gradient Descend: Learning Rate (I)

COST FUNCTION

$$J(w_o, w_1) = \frac{1}{2m} \sum_{i=1}^{i=m} \left(f\left(x^{(i)}\right) - y^{(i)}\right)^2$$

OBJECTIVE FUNCTION

$$\min_{w_0, w_1} J(w_0, w_1)$$

WHAT GRADIENT DESCEND DOES

Repeat until convergence {

$$w_j = w_j - \alpha \frac{d}{dw_j} J(w_0, w_1)$$

}

Update all parameters simultaneously

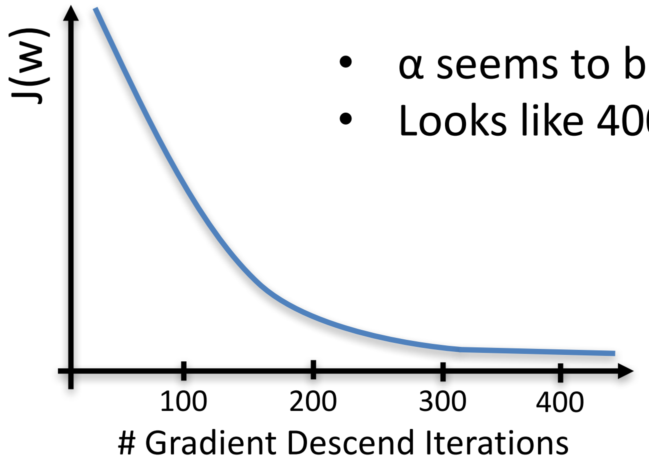
Learning Rate

Controls how big a step we take on each iteration of gradient descend

If α is too small → GD may take too long to converge.

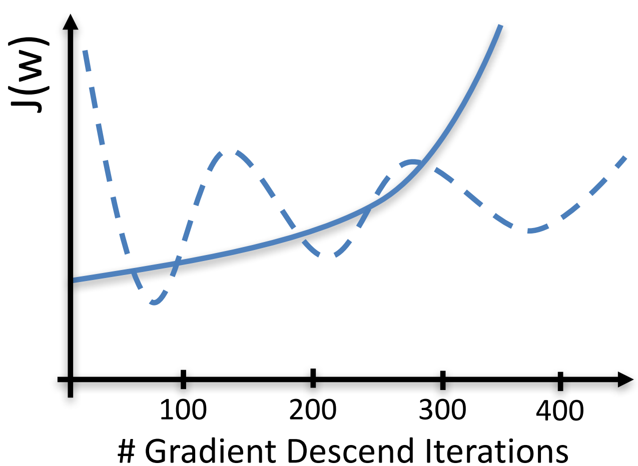
If α is too large → GD may fail to converge.

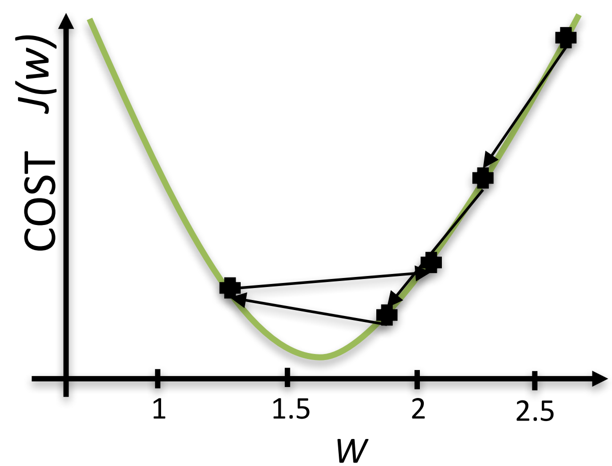
Gradient Descend – Learning Rate (II)



- α seems to be correct as cost goes down with every iteration.
- Looks like 400 iterations is sufficient for convergence.

When α is too big…





$$f_w(x) = wx$$

Logistic Regression

x	$f_w(x), J(w)$	y
Features	$\longrightarrow$	Real Value

Univariate Linear Regression: $f_w(x) = w_o + w_1 x$

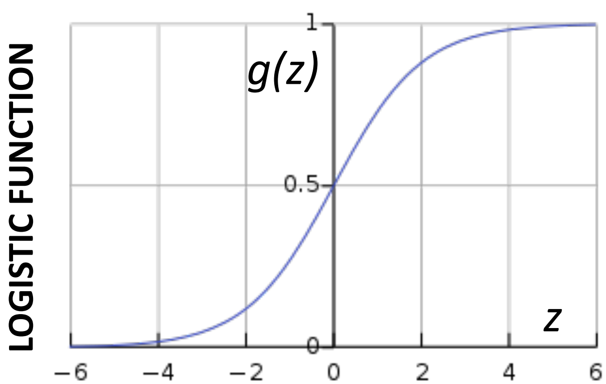
Multivariate Linear Regression: $f_w(x_1, \dots, x_n) = w_o + w_1 x_1 + \cdots + w_n x_n = w^T x$

$$x = \begin{bmatrix} 1 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \qquad w = \begin{bmatrix} w_0 \\ w_1 \\ \cdots \\ w_n \end{bmatrix}$$

x	$f_w(x), J(w)$	y
Features	$\longrightarrow$	**Class Label [0 or 1]**

Logistic Regression → we would like $f_w(x)$ to be so that: $0 \leq f_w(x) \leq 1$

$$f_w(x) = w^T x \xrightarrow[g(z)=\frac{1}{1+e^{-z}}]{} f_w(x) = \mathrm{g}(w^T x)$$



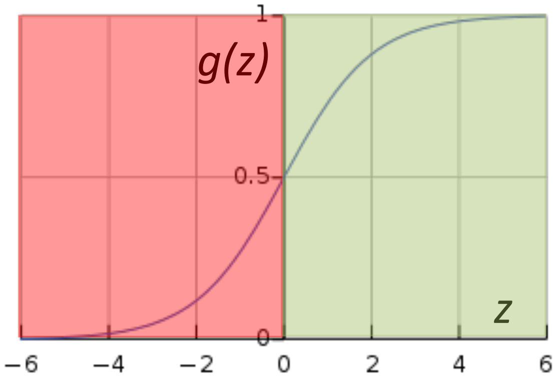
$f_w(x)$ = estimated probability that y=1 for input x

We will predict "*y=1*" if $f_w(x) \geq 0.5$

We will predict "*y=0*" if $f_w(x) < 0.5$

Cost Function: $J(w) = -\Big(y \cdot log\big(f_w(x)\big) + (1-y) \cdot log\big(1 - f_w(x)\big)\Big)$

Logistic Regression – Linear Decision Boundary



We will predict "*y=1*" if

$$f_w(x) \geq 0.5 \longrightarrow g(w^T x) \geq 0.5 \longrightarrow w^T x \geq 0$$

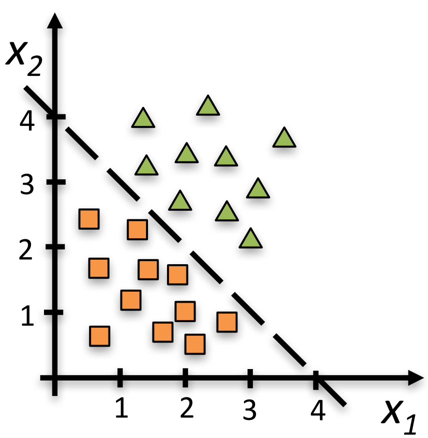
We will predict "*y=0*" if

$$f_w(x) < 0.5 \longrightarrow g(w^T x) < 0.5 \longrightarrow w^T x < 0$$

Let's imagine a case with:

- Two features: (x_1, x_2)
- A training set
- A logistic regression classifier
- Trained Solution: $w^T=[-4, 1, 1]$

$$f_w(x) = 1/1 + e^{-(w_0+w_1x_1+w_2x_2)}$$



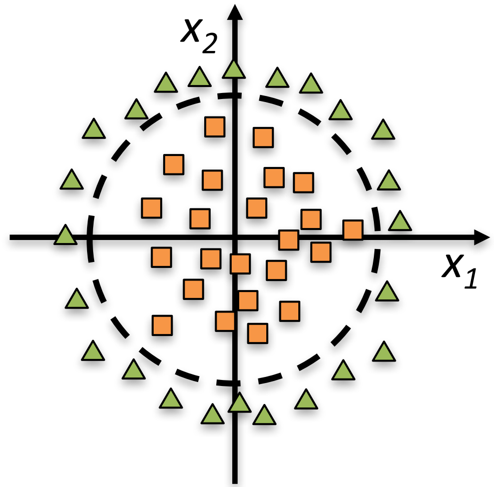
"y=1" if $\quad -4 + x_1 + x_2 \geq 0$

"y=0" if $\quad -4 + x_1 + x_2 < 0$

$$\downarrow$$

$$x_1 + x_2 = 4$$

Decision Boundary



$$f_w(x) = g(w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2)$$

$$w^T = [-1,0,0,1,1]$$

Predict "y=1" if $-1 + x_1^2 + x_2^2 \geq 0$

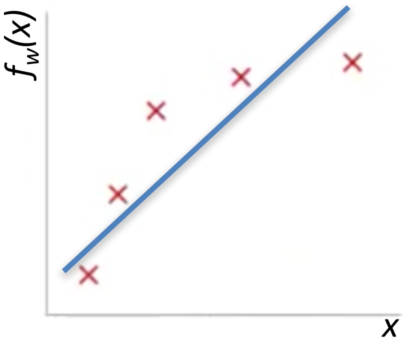
$$x_1^2 + x_2^2 = 1$$

Non Linear Decision Boundary

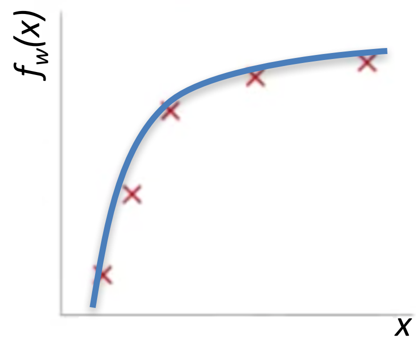
Overfitting

<u>OVERFITTING</u>: When too many features and an excessively complex model leads to an extremely good fit for the training data, but poor generalization for any additional data.

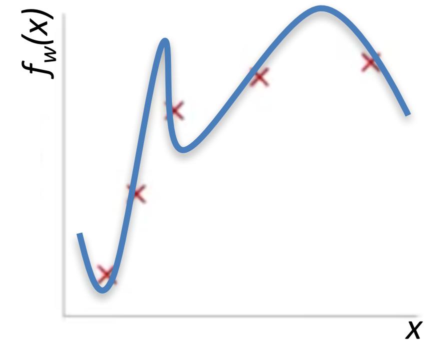
REGRESSION



$f_w(x) = w_o + w_1 x$

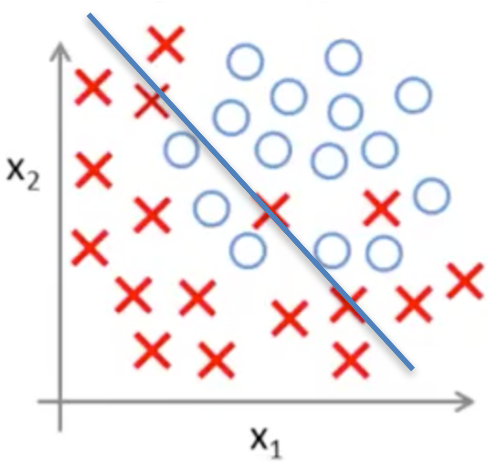


$f_w(x) = w_o + w_1 x + w_2 x^2$

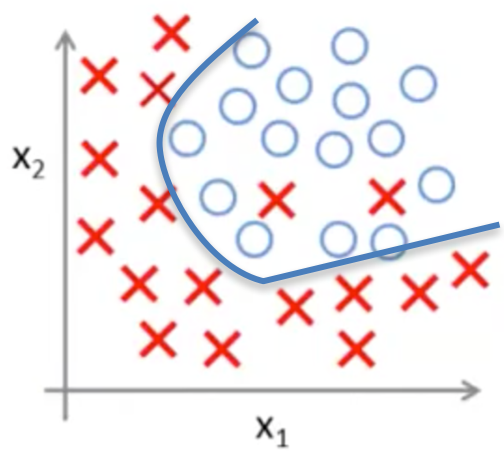


$f_w(x) = w_o + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$

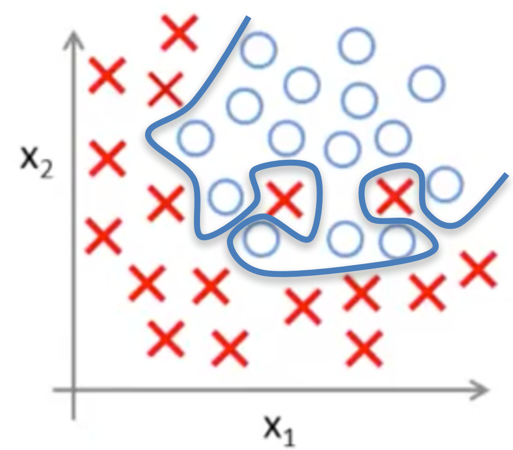
LOGISTIC REGRESSION



$f_w(x) = \mathrm{g}(w_o + w_1 x_1 + w_2 x_2)$



$$f_w(x) = \mathrm{g}\begin{pmatrix} w_o + w_1 x_1 + \\ w_2 x_2 + w_3 x_1^2 + w_4 x_2^2 \end{pmatrix}$$



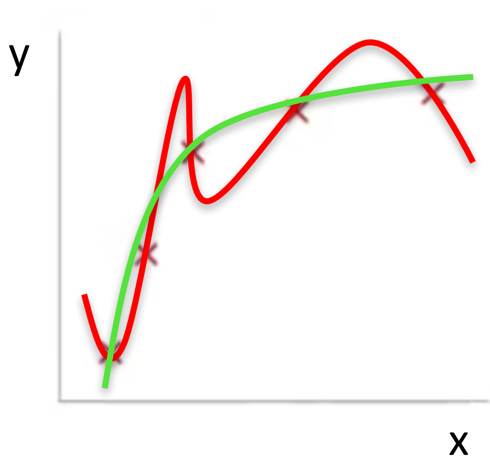
$$f_w(x) = \mathrm{g}\begin{pmatrix} w_o + w_1 x_1 + w_2 x_1^2 + \\ w_3 x_1^2 x_{2+w_4} x_1^2 x_2^2 + \cdots \end{pmatrix}$$

Overfitting – Potential Solutions

1. Reducing the number of features:
    - Model Selection Algorithms.
    - Need to be careful not to throw away useful information.

2. Regularization:
    - Keep all features, but enforce very low or zero *w* for those least informative.
    - Implemented by adding a "regularization term" to the cost function.

LINEAR REGRESSION



WITHOUT REGULARIZATION

$$f_w(x) = w_o + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

$$\min_{w_0, w_1} \frac{1}{2m} \sum_{i=1}^{i=m} \left(f\left(x^{(i)}\right) - y^{(i)}\right)^2$$

WITH REGULARIZATION

$$f_w(x) = w_o + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$

$$\min_{w_0, w_1} \frac{1}{2m} \sum_{i=1}^{i=m} \left(f\left(x^{(i)}\right) - y^{(i)}\right)^2 + 1{,}000 \cdot w_3 + 1{,}000 \cdot w_4$$

Regularization (cont.)

Promoting small values for learning parameters will:

- Enforce the adoption of "simpler" models / smoother functions
- Be more robust against overfitting

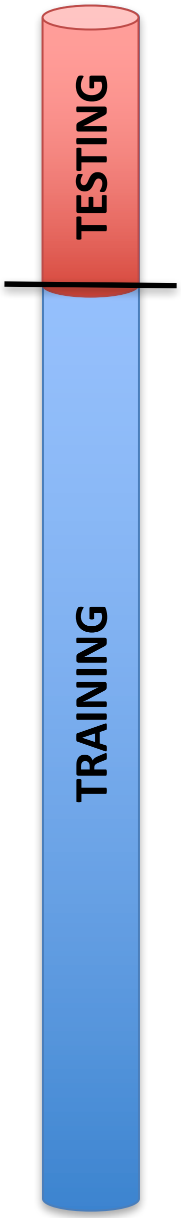
In fMRI, maybe our feature space is composed of over 100 voxels…

- Feature Space: $x^{(i)} = [x^{(i)}_1, x^{(i)}_2, x^{(i)}_3, \ldots x^{(i)}_{100}]$
- Linear Regression Model: $f_w(x) = w_o + w_1 x_1 + \ldots + w_{100} x_{100}$
- Objective Function:

$$\min_{w_0, w_1} \frac{1}{2m} \sum_{i=1}^{i=m} \left(f\left(x^{(i)}\right) - y^{(i)}\right)^2 + \underbrace{\lambda \sum_{i=1}^{n} w_i^2}_{\text{Regularization Term}}$$

- λ is the regularization parameter
  - Controls the tradeoff between fitting the data as best as possible (first term of the cost function) and keeping the model simple (regularization term).
  - λ excessively high → all w will be close to zero (even good ones) / Underfitting
  - Model selection algorithms can help us select λ automatically.

Testing the Performance of a Classifier (I)



OPTION 1:

- Use all data for training
- Report error training as the performance of the classifier
- **INCORRECT**: Prone to overfitting / Too optimistic estimates of performance

OPTION 2 (TRAINING A SINGLE MODEL):

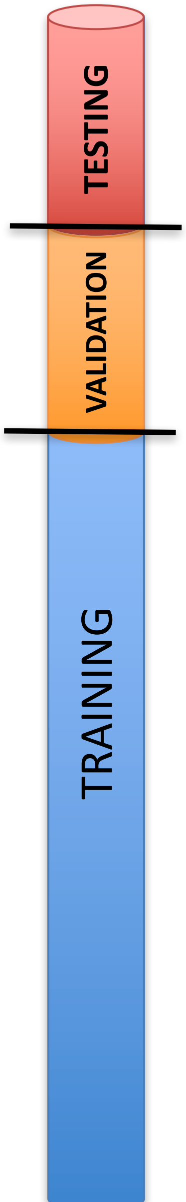
- All meta-parameters fixed.
- Divide the dataset in two subsets:
  - **TRAINING**: We use this data to learn the model parameters (w^T).
  - **TESTING**: We use this data to estimate the performance / generality.
- Controls against overfitting / overestimating performance.
- Examples in each subset should be drawn randomly
- Ensure a balanced presence of classes in both subsets

True accuracy: the probability that a classifier will correctly label a new example drawn at random from the same distribution that the training examples came from.

Accuracy on test set is an estimate of the true accuracy.

How precise this estimate is depends on the size of the test set

Testing the Performance of a Classifier (II)



OPTION 3 (MODEL SELECTION PROBLEM):

- We want to train, but also do some sort of model selection.
- Example: Not sure which one of three linear models to use

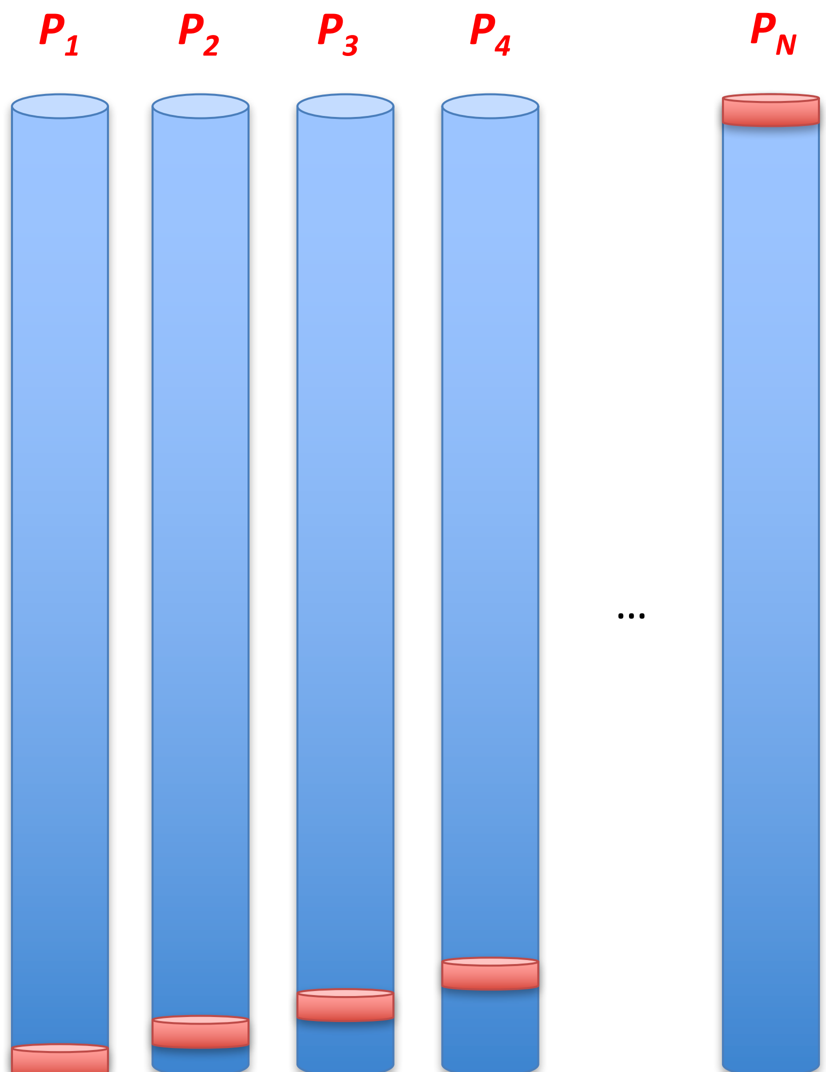
$$\text{Degree } d=1 \rightarrow f_w(x) = w_o + w_1x_1$$
$$\text{Degree } d=2 \rightarrow f_w(x) = w_o + w_1x_1 + w_2x_1^2$$
$$\text{Degree } d=3 \rightarrow f_w(x) = w_o + w_1x_1 + w_2x_1^2 + w_3x_1^3$$

In addition to training each model, we want to automatically pick d

- We need to subdivide our dataset in three subsets:
    - **TRAINING**: We use this to train all models (estimate w^T for all models)
    - **VALIDATION**: We use this to select the best model
    - **TESTING**: We use this to estimate final performance (generality)

Testing the Performance of a Classifier (III)

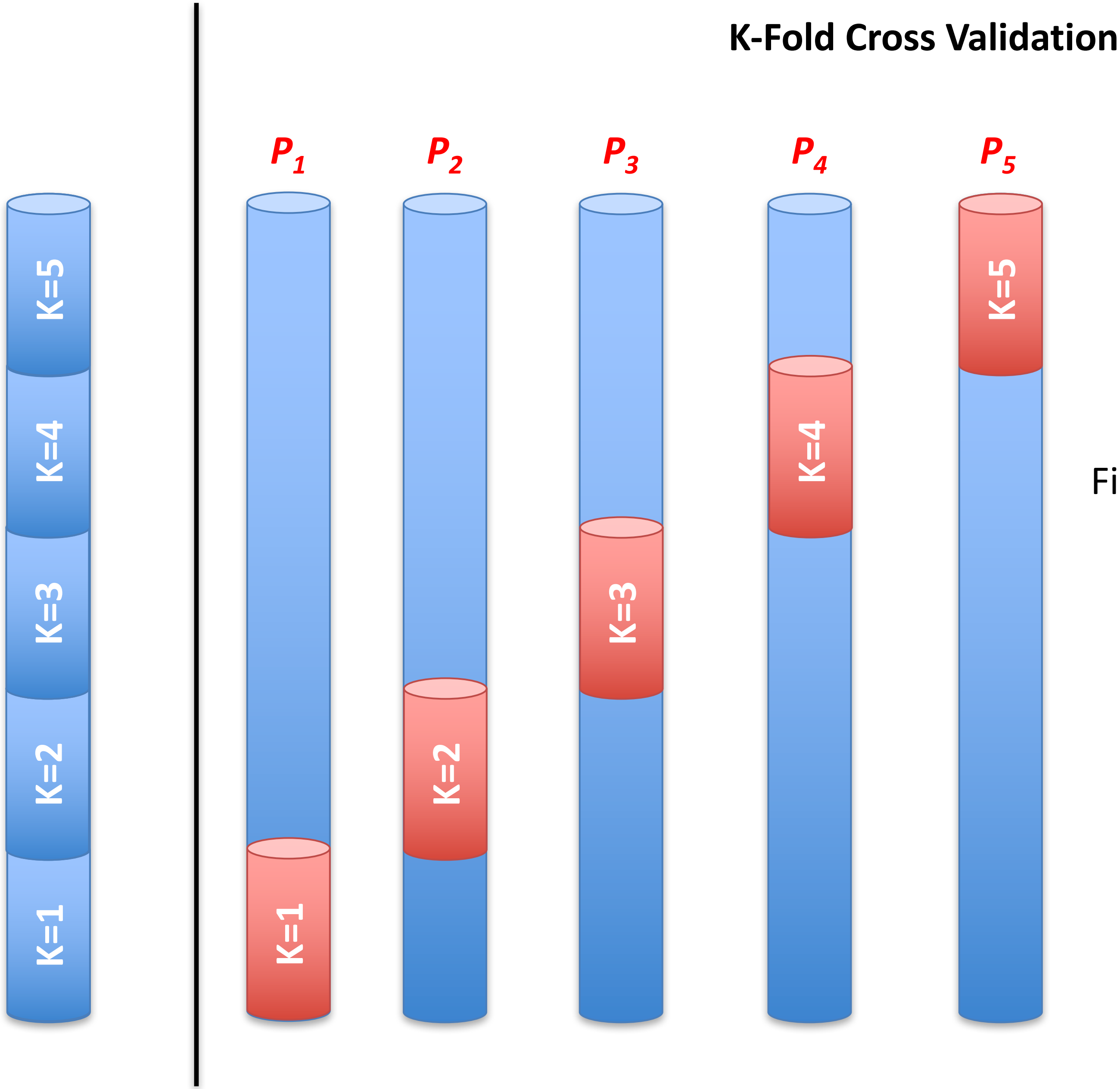
Leave One Out Cross Validation



Final Classifier Performance

$$P = \frac{1}{N}\sum_{i=1}^{N} P_i$$

Testing the Performance of a Classifier (III)



Final Classifier Performance

$$P = \frac{1}{N} \sum_{i=1}^{K_{max}} P_i$$

Concepts Reviewed so far….

- Types of Machine Learning: Supervised / Unsupervised
- Regression vs. Classification Problems
- Objective Function / Cost Function
- Learning in terms of Gradient Descend
- Learning Rate & How to monitor learning
- Logistic Regression
- Linear & Non-Linear Decision Boundaries
- Feature selection (non-linear boundaries)
- Overfitting
- Regularization / Regularization Parameter
- Training / Validation / Testing

A word on Classifier Selection

LINEAR REGRESSION

SUPPORT VECTOR MACHINE

KERNEL SUPPORT VECTOR MACHINE

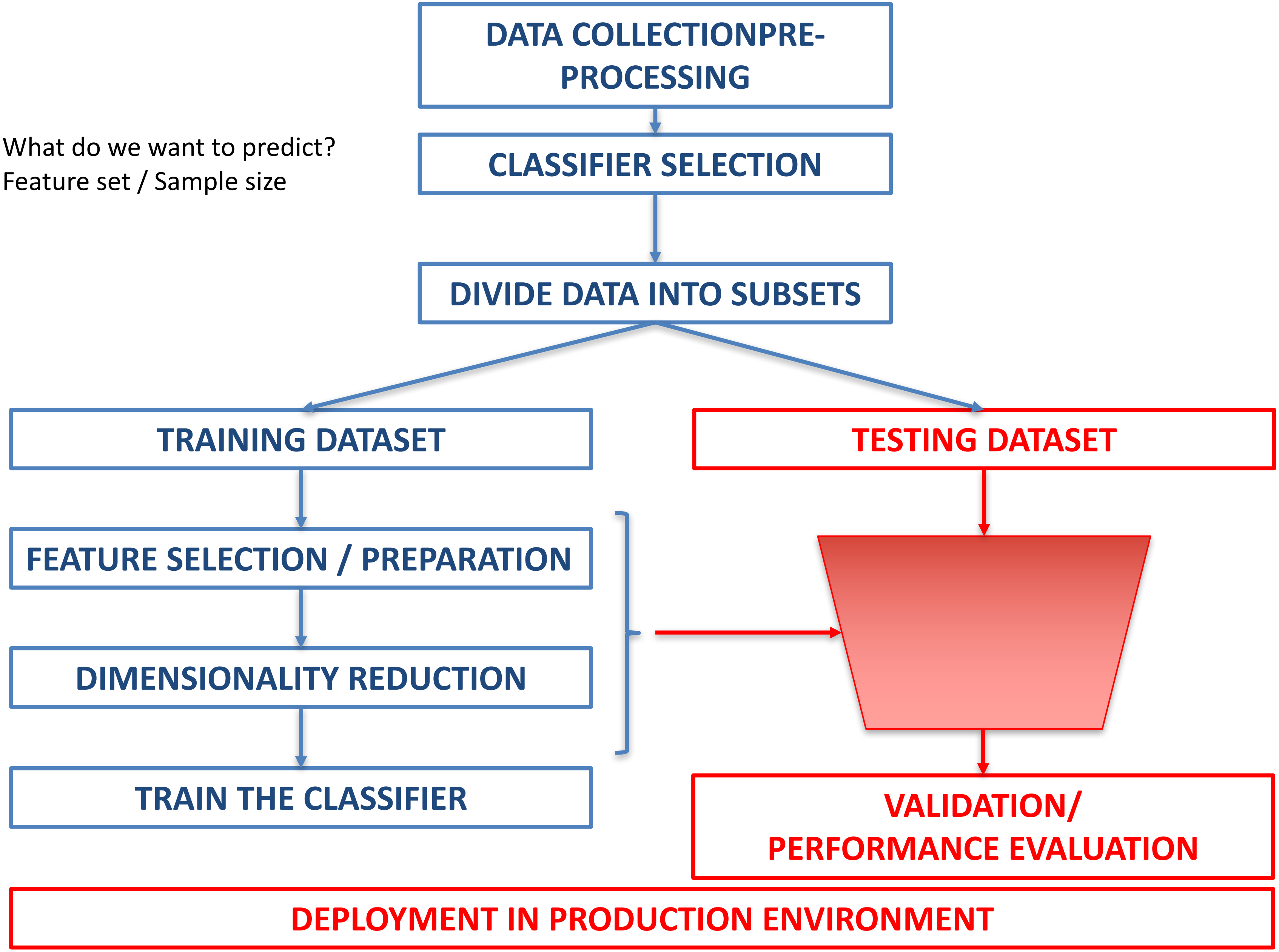
DEEP NEURONAL NETWORKS

COMPLEXITY OF MODEL INCREASES

Need of additional examples
Risk of overfitting
Difficulty of interpretation

ML in Neuro-Imaging Workflow

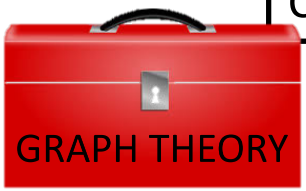
What do we want to predict?
Feature set / Sample size



Introduction / Agenda

EXPERIMENTAL DESIGN → DATA ACQUISITION → DATA PRE-PROCESSING → GLM – ACTIVATION

DATA PRE-PROCESSING → FUNCTIONAL CONNECTIVITY



Centrality, Degree, Clustering Coefficient, Community, etc.

Logistic Regression, Support Vector Machines, ICA, K-Means, Convolutional Networks, etc.

Cost Function, Learning Rate, Gradient Descend, Decision Boundary, Regularization, etc.

- A few applications to fMRI data.
- A few words on software.
- Additional Resources to learn more.

Logistic Regression & fMRI Neurofeedback (I)

Perceptual Learning Incepted by Decoded fMRI Neurofeedback Without Stimulus Presentation

Kazuhisa Shibata*, Takeo Watanabe*,†, Yuka Sasaki‡, Mitsuo Kawato

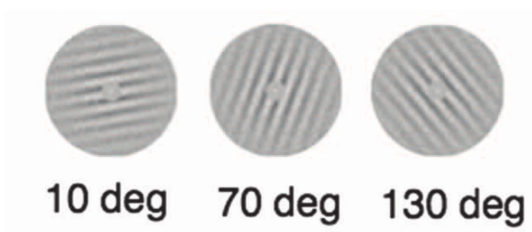
Science AAAS

GOAL: Is early visual cortex sufficiently plastic to undergo visual perception learning (VPL)?

METHODS: fMRI + Neurofeedback + Logistic Regression

EXPERIMENT:

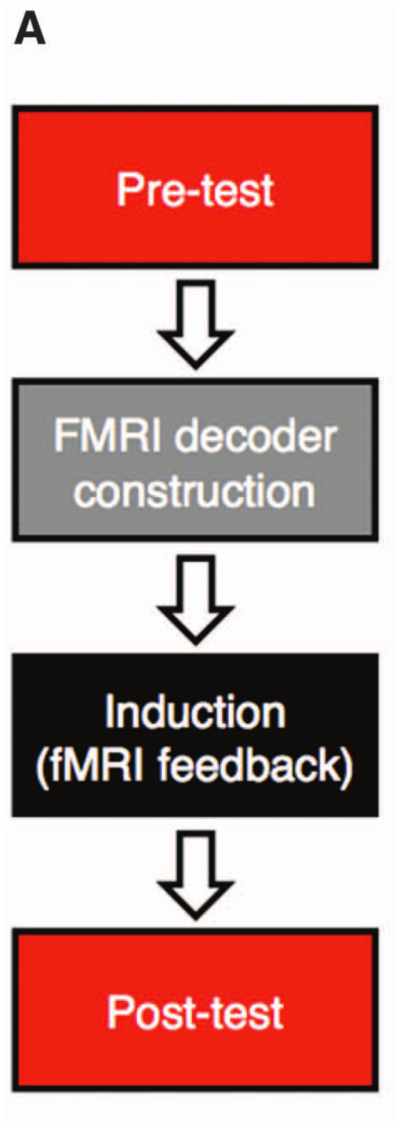
- Induce activity patterns in V1/V2 that correspond to given stimulus orientation without stimuli/subject awareness (fMRI+NF+LR)
- Evaluate whether such induced activation caused VPL specific to that orientation.



RESULTS/CONCLUSIONS:

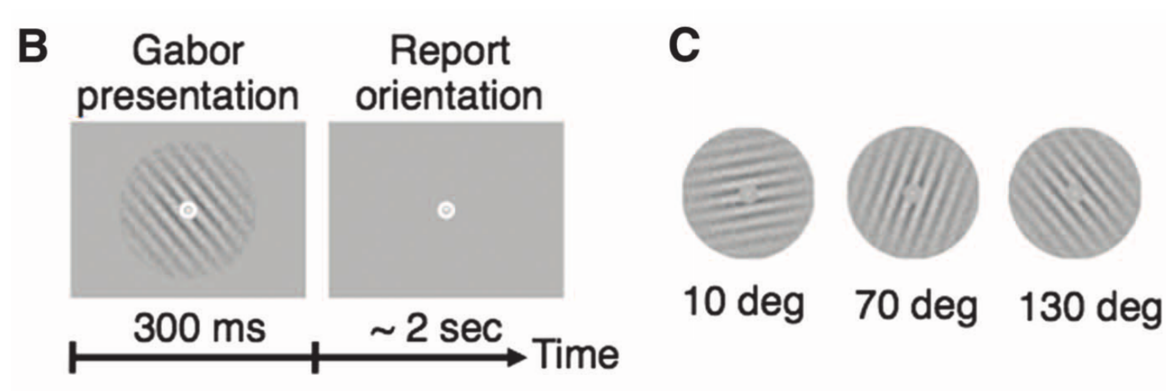
- The induced activation caused VPL specific to the orientation.
- V1/V2 is so plastic that mere induction of activity patterns can lead to VPL
- This fMRI/NF/LR technique can induce plasticity in a highly selective manner

Logistic Regression & fMRI Neurofeedback (II)

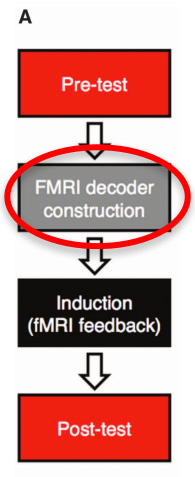


Behavioral Performance on orientation discrimination task

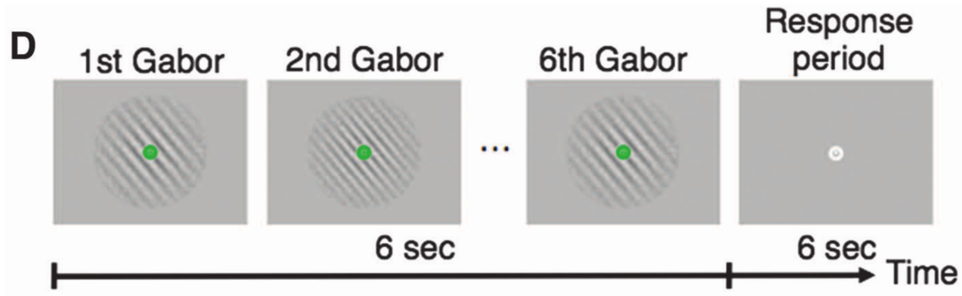
Behavioral Performance on orientation discrimination task



Shibata et al., Science 2011

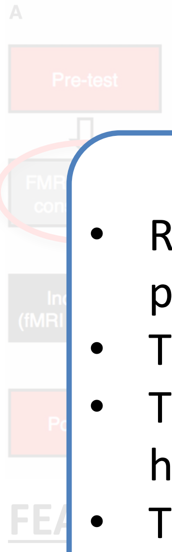


DATA COLLECTION: Perform task designed to maintain attention to the Gabor patches while fMRI signals were recorded.



FEATURE SELECTION: Obtain activity patterns induced by each orientation from V1/V2.

Shibata et al., Science 2011



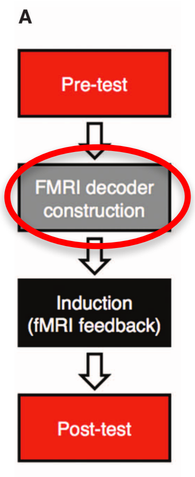
DATA COLLECTION: Perform task designed to maintain attention to the Gabor patches while fMRI signals were recorded.

- Retinotopic mapping + V1/V2 localizer (areas to be activated by the Gabor patches).
- Training data pre-processing: motion correction, no spatial or temporal smoothing.
- Time-courses from ref. regions were extracted and shifted by 6s (to account for hemodynamic response delay).
- Time-courses were linearly detrended and converted to Z-scores (feature normalization – avoid baseline differences across runs).
- Decoder input = voxel-wise average BOLD signal across the 3 volumes that correspond to the 6s of stim. Presentation per trial.
- Automatic feature selection (only relevant V1/V2 voxels enter the final model)

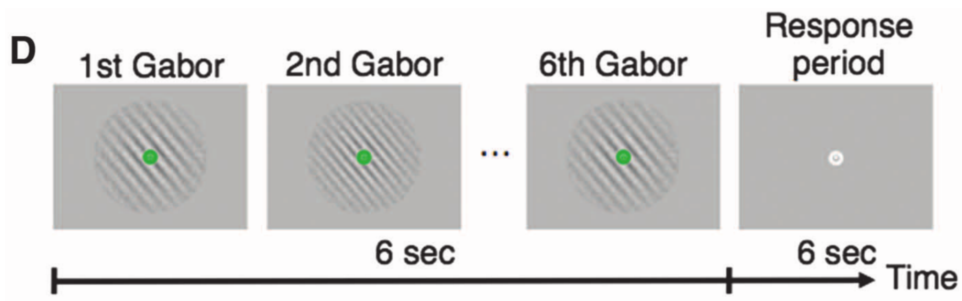
- SAMPLE SIZE: 240 samples per subject
- MEAN # FEATURES: 239 +/- 29 voxels.

Shibata et al., Science 2011

Logistic Regression & fMRI Neurofeedback (II)



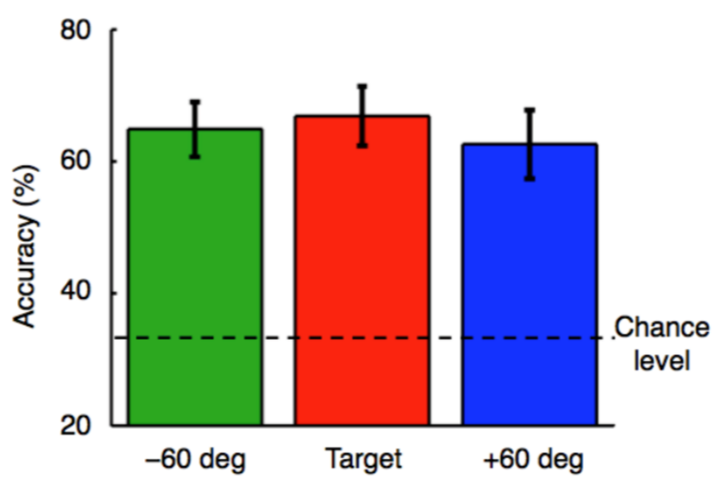
DATA COLLECTION: Perform task designed to maintain attention to the Gabor patches while fMRI signals were recorded.



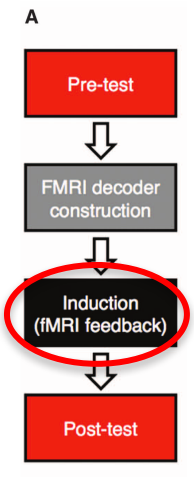
FEATURE SELECTION: Obtain activity patterns induced by each orientation from V1/V2.

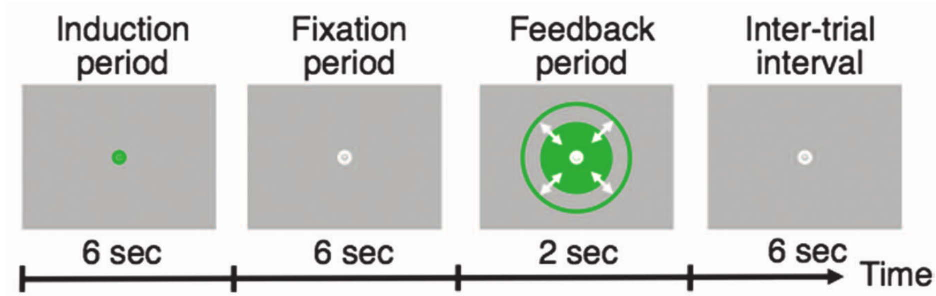
CLASSIFIER TRAINING: Construct a multinomial sparse logistic regression decoder that would classify upcoming patterns of fMRI signals into one of three orientations.

TESTING: Perform LOOV + T-test against chance level (33%)



Shibata et al., Science 2011





5-10 days

PURPOSE OF NF EXPERIMENT: Subject had to learn to induce activity patterns that corresponded to a target orientation.

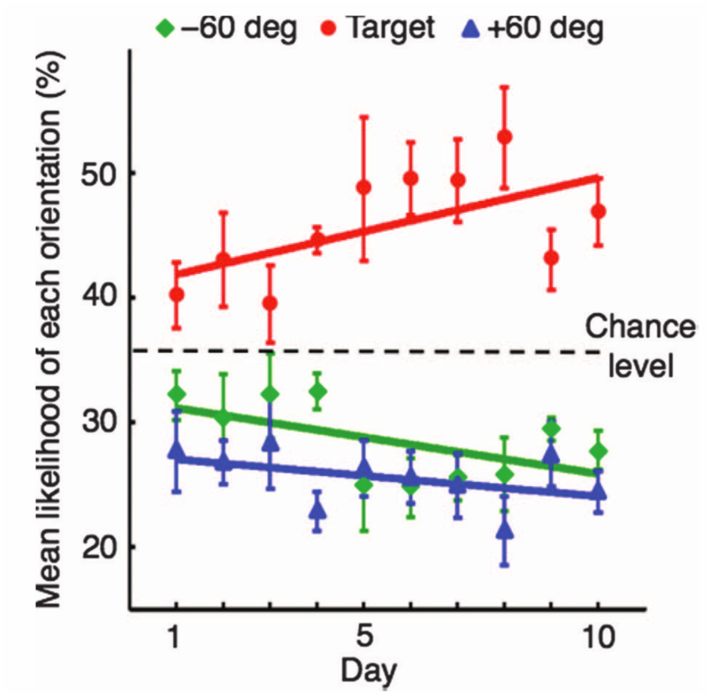
SUBJECT INSTRUCTIONS: “somehow regulate activity in the posterior part of the brain to make the solid green disc that was presented 6 s later as large as possible (the maximum possible size corresponds to the outer green circle)” + Payment proportional to avg. disk size.

SUBJECTS DIDN’T KNOW: The size of the disc in the NF period corresponded to the decoder output for the target orientation, which roughly represented how similar activity in V1/V2 during induction period agreed with activity in V1/V2 during presentation of the target stimuli during the decoder construction stage.

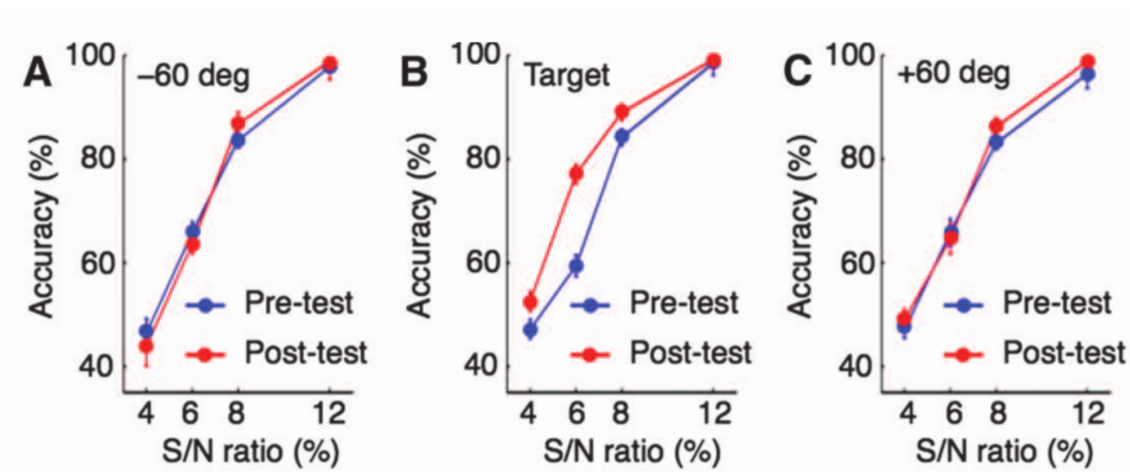
Shibata et al., Science 2011

Logistic Regression & fMRI Neurofeedback (IV)

NF DID INDUCE THE EXPECTED PATTERNS OF ACTIVATION



NF-INDUCED LEARNING TRANSLATED INTO BEHAVIORAL CHANGES IN PERFORMANCE ONLY FOR THE TARGET ORIENTATION



Shibata et al., Science 2011

Support Vector Machine for fMRI-based Sleep Staging (I)

ELSEVIER
Contents lists available at SciVerse ScienceDirect
NeuroImage
journal homepage: www.elsevier.com/locate/ynimg

Automatic sleep staging using fMRI functional connectivity data

Enzo Tagliazucchi *, Frederic von Wegner, Astrid Morzelewski, Sergey Borisov, Kolja Jahnke, Helmut Laufs

GOAL: Develop a method for automatic sleep staging based only on fMRI FC data

METHODS: fMRI + EEG + SVM

EXPERIMENT:

- Concurrent rest fMRI/EEG data was acquired continuously for approx. 50min
- Runs were segmented in periods of 60s
- For each segment, sleep staging was performed with EEG (generation of labeled data)
- A Multi-class SVM was trained on the fMRI data + sleep labels derived from EEG
- 5-Fold Cross-validation

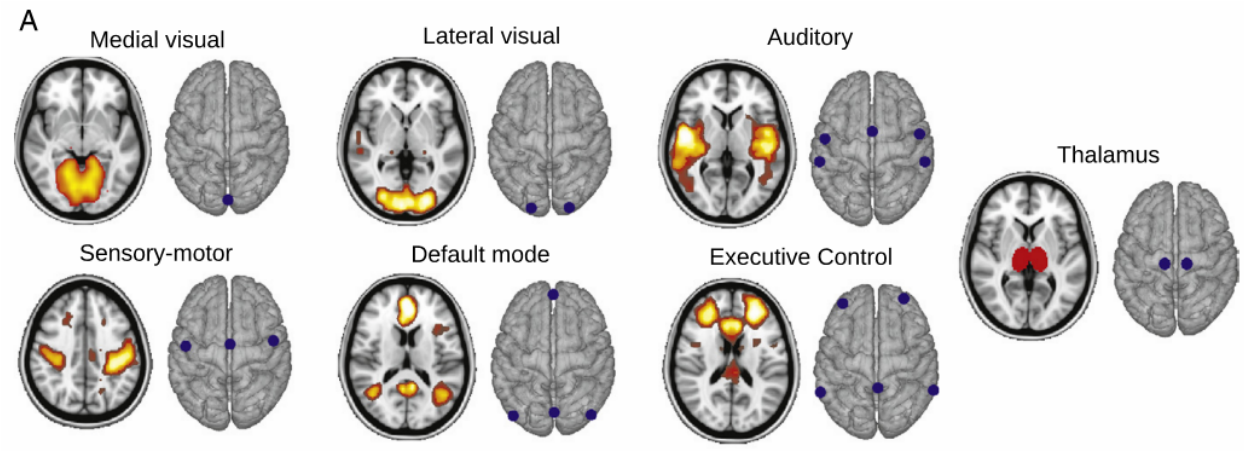
RESULTS/CONCLUSIONS:

- 80% Accuracy achieved.
- Good generalization to two additional datasets (over 80% accuracy)
- Method may help avoid/model confounds in resting state due to fluctuations in vigilance levels

Support Vector Machine for fMRI-based Sleep Staging (II)

- **PREPROCESSING**: head motion, spatial normalization to MNI, physio correction, ***spatial smoothing***, bandpass filtering.

- **FEATURES**: 20 functionally defined ROIs + bilateral thalamus. Decision based on previous literature

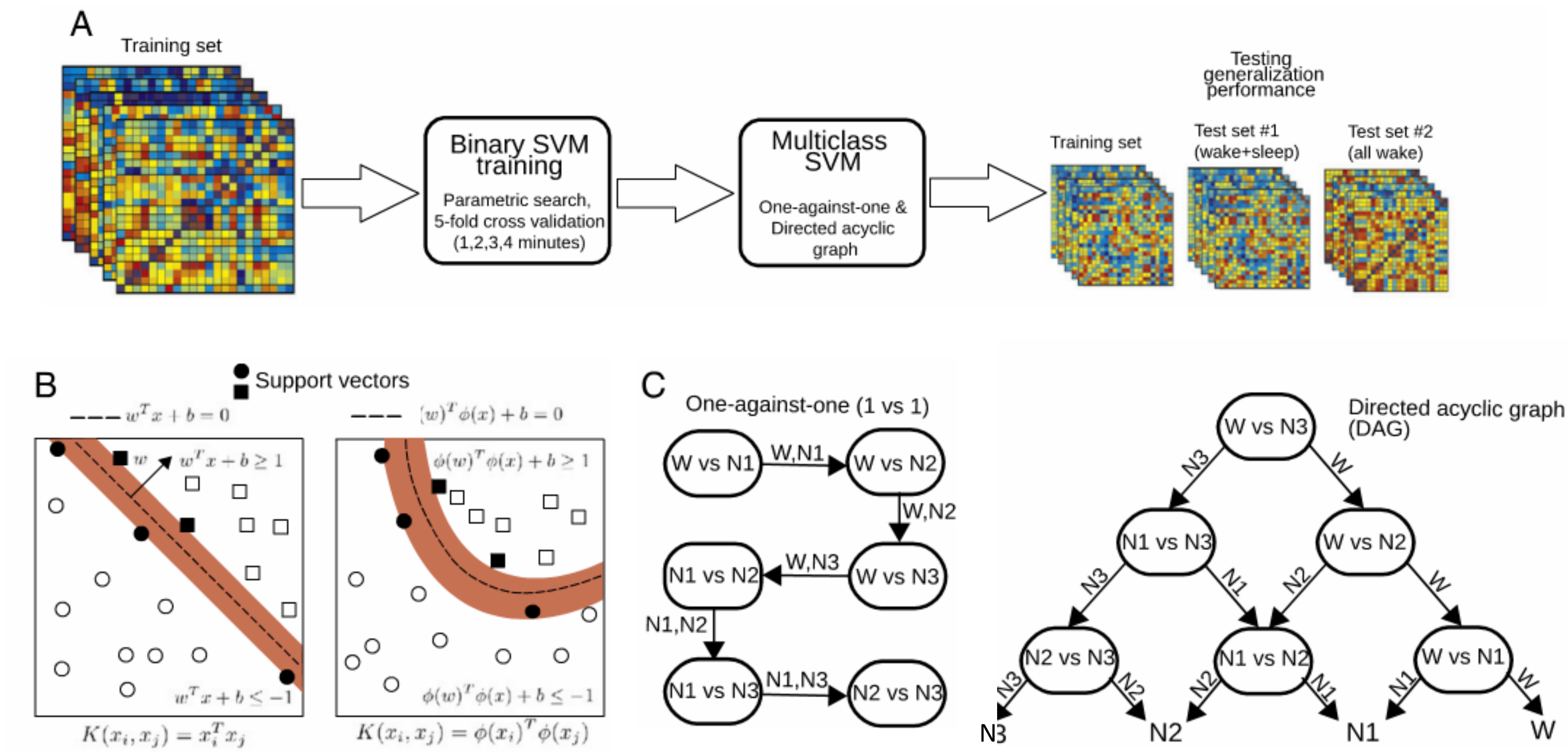


Tagliazucchi et al. NeuroImage 2012

Support Vector Machine for fMRI-based Sleep Staging (III)

TRAINING PROCEDURE:

- Four classes (REM, N1, N2, N3) ➔ 6 Binary classification problems

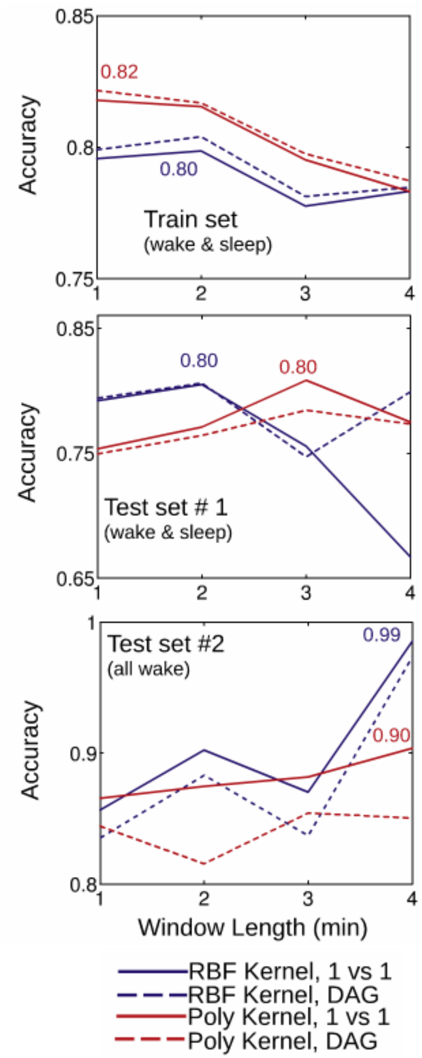


The one with most votes wins

Tagliazucchi et al. NeuroImage 2012

Support Vector Machine for fMRI-based Sleep Staging (IV)

CLASSIFICATION VALIDATION



	W	N1	N2	N3
W	0.90	0.08	0.02	0
N1	0.28	0.62	0.10	0
N2	0.10	0.07	0.77	0.05
N3	0.05	0	0.14	0.81

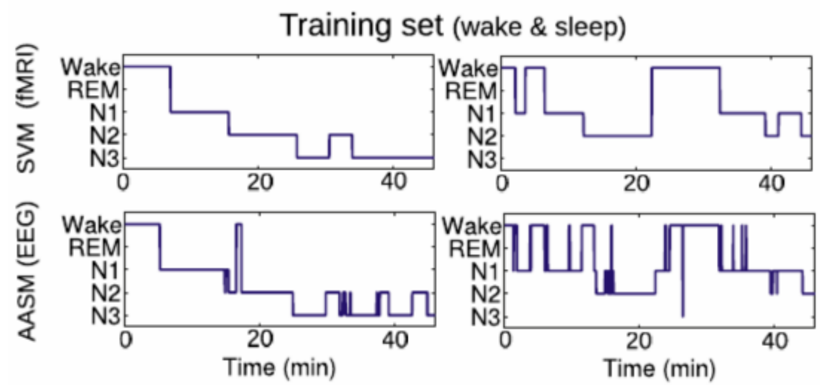
Train set

	W	N1	N2	N3
W	0.81	0.14	0.05	0
N1	0.11	0.78	0.09	0.01
N2	0.10	0.09	0.76	0.06
N3	0.06	0	0.10	0.85

Test set #1

	W	N1	N2	N3
W	0.87	0.10	0.01	0.01
N1				
N2				
N3				

Test set #2



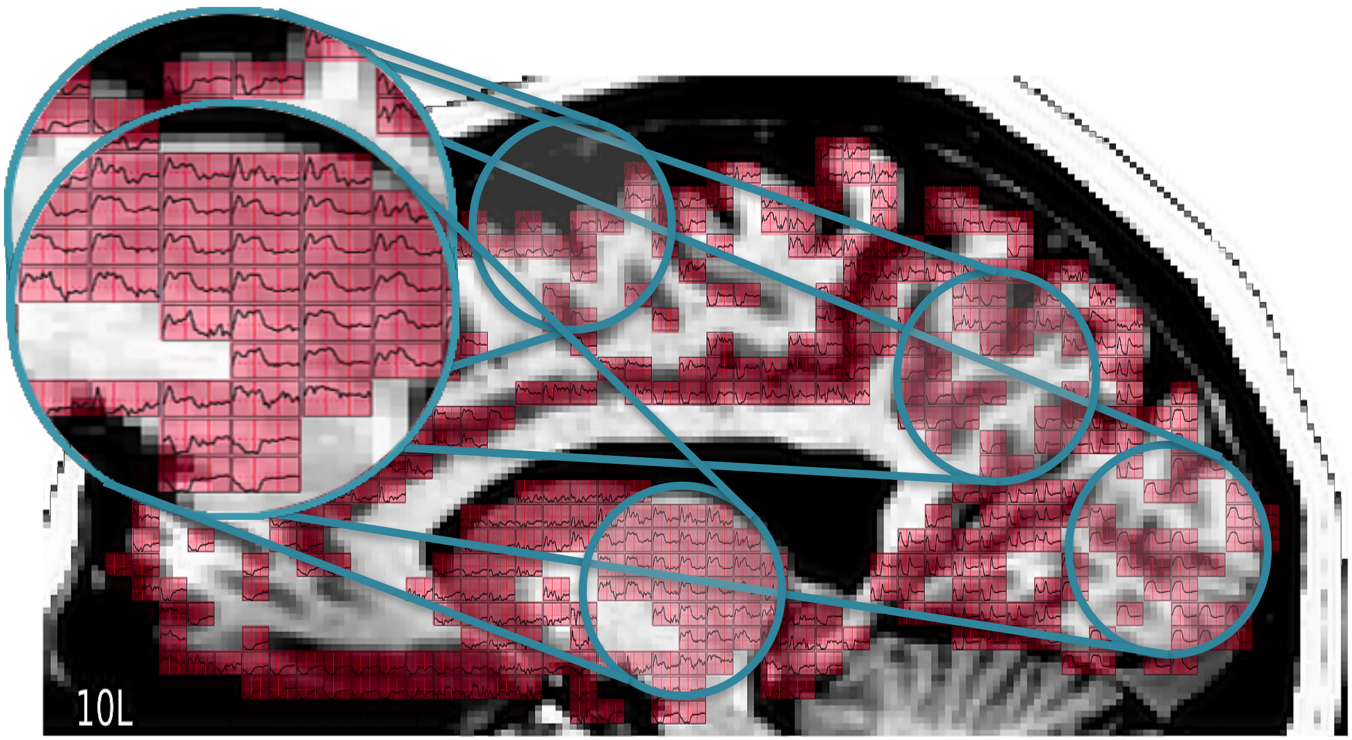
Tagliazucchi et al. NeuroImage 2012

Unsupervised Learning (K-means) to cluster voxel-wise HRFs (I)

Whole-brain, time-locked activation with simple tasks revealed using massive averaging and model-free analysis

Javier Gonzalez-Castillo[a,1], Ziad S. Saad[b], Daniel A. Handwerker[a], Souheil J. Inati[c], Noah Brenowitz[a], and Peter A. Bandettini[a,c]

[a]Section on Functional Imaging Methods, Laboratory of Brain and Cognition, [b]Scientific and Statistical Computing Core, and [c]Functional MRI Facility, National Institute of Mental Health, National Institutes of Health, Bethesda, MD 20892



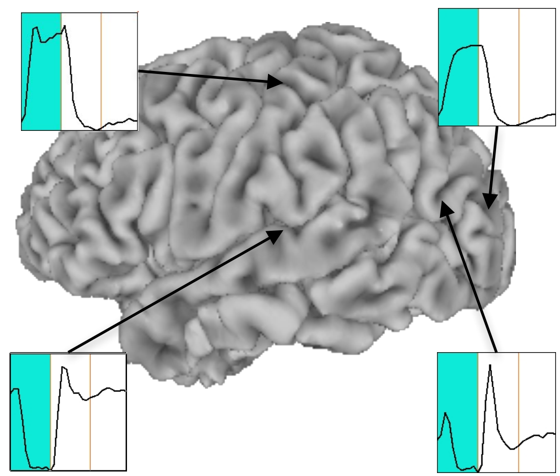
ARE RESPONSE SHAPES RANDOMLY DISTRIBUTED ACROSS THE BRAIN

OR

DO THEY CLUSTER IN A FUNCTIONALLY/ANTOMICALLY MEANINGFUL MANNER?

Unsupervised Learning (K-means) to cluster voxel-wise HRFs (II)

FEATURES:
- Each voxel is characterized by its response to a block of visual stimulation.
- Blocks are 20s (ON) + 40s (OFF) & TR=2s ➔ Each voxel has 30 features (time-points)



UNSUPERVISED LEARNING:
- Trying to uncover if there is some structure in the data
- Labels are missing (Don't have names for the different response profiles)
- K-Means Clustering Algorithm

Gonzalez-Castillo et al., PNAS 2012

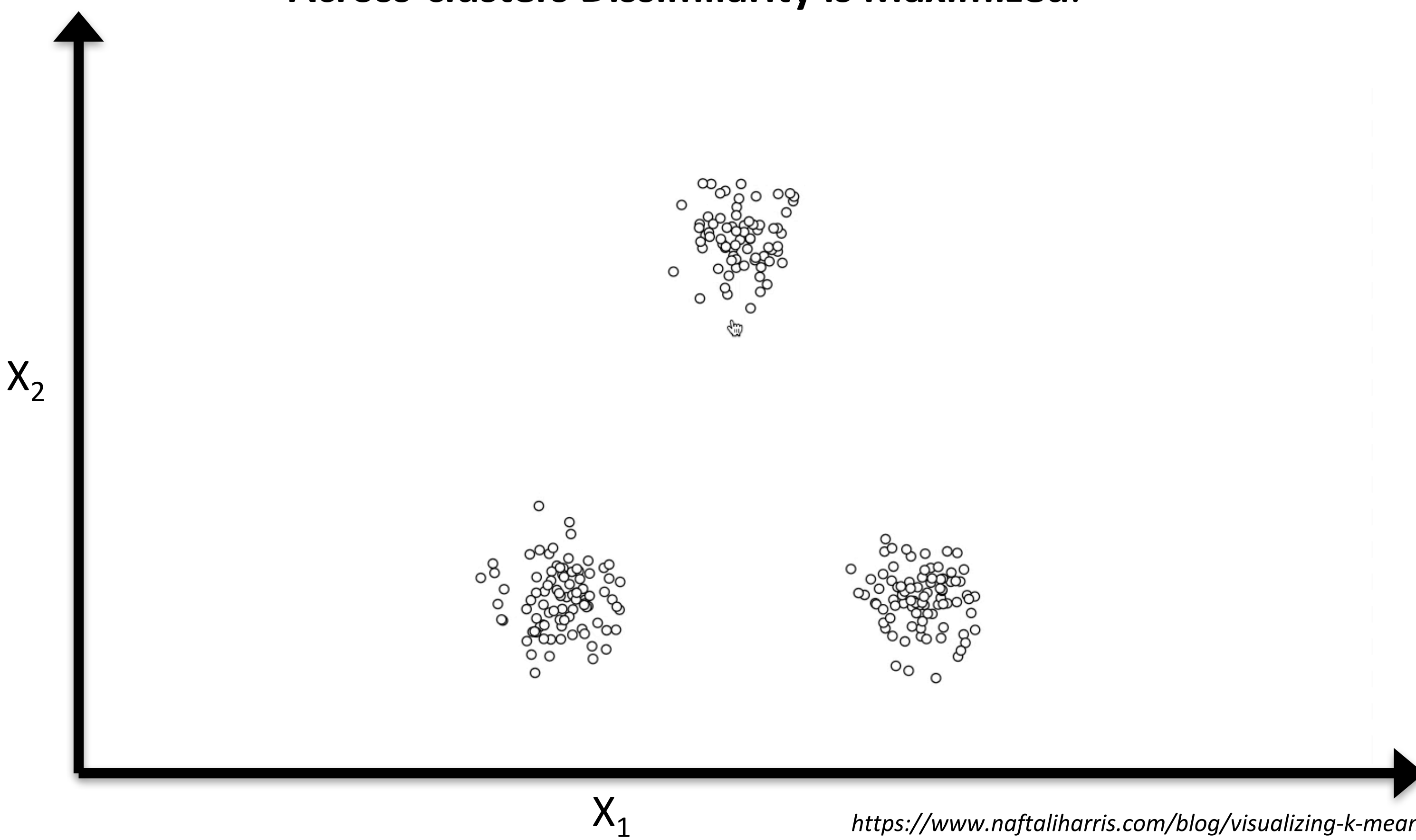
Unsupervised Learning (K-means) to cluster voxel-wise HRFs (III)

K-MEAN ALGORITHM

- Set of N examples $\{z_j\}$ from R^P
- Dissimilarity metric (D)
- K= Number of expected clusters

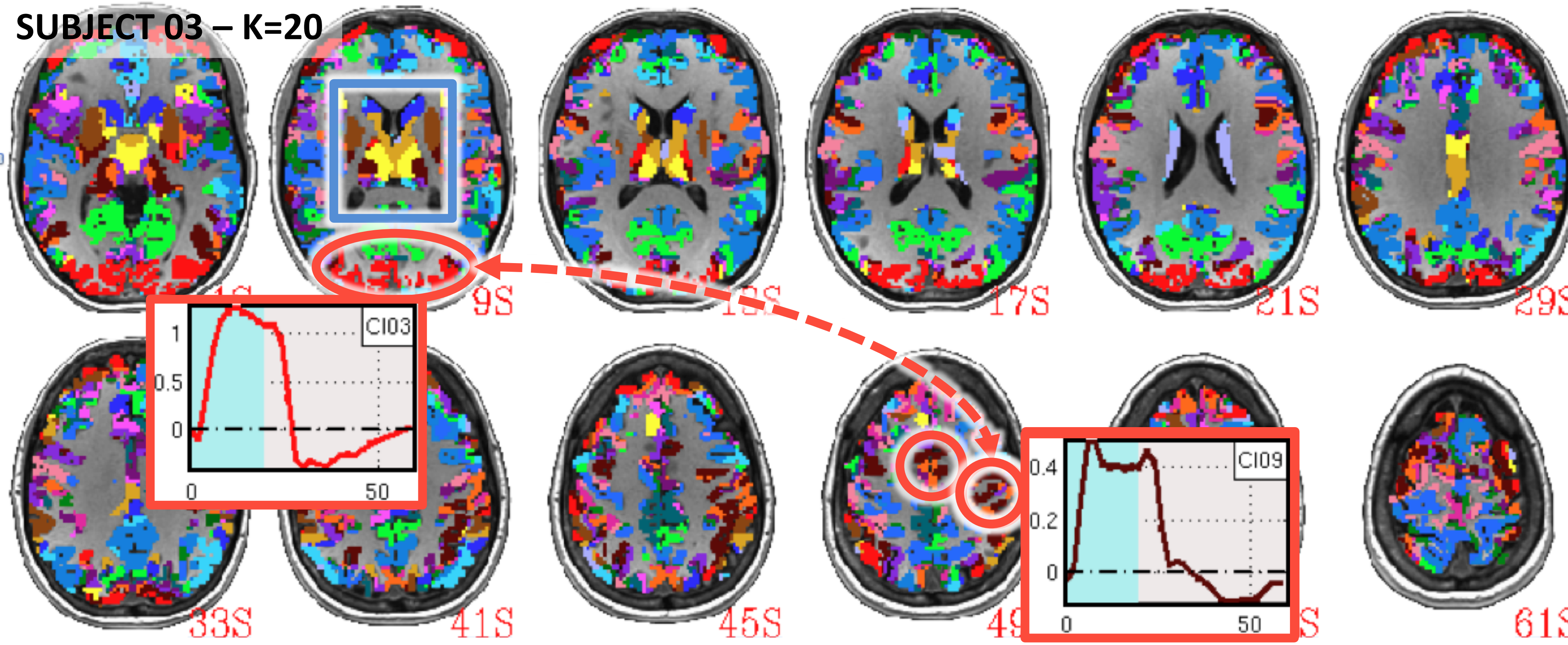
- 60 Points in a 2-D space $z_1=\{x_1,y_1\}\ldots z_{60}=\{x_{60},y_{60}\}$
- D = Euclidean Distance
- 3 Clusters

K-Means algorithm generate clusters so that **Within-cluster Dissimilarity is Minimized** and **Across-clusters Dissimilarity is Maximized**.



https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Unsupervised Learning (K-means) to cluster voxel-wise HRFs (IV)



NOT RANDOMLY DISTRIBUTED IN SPACE

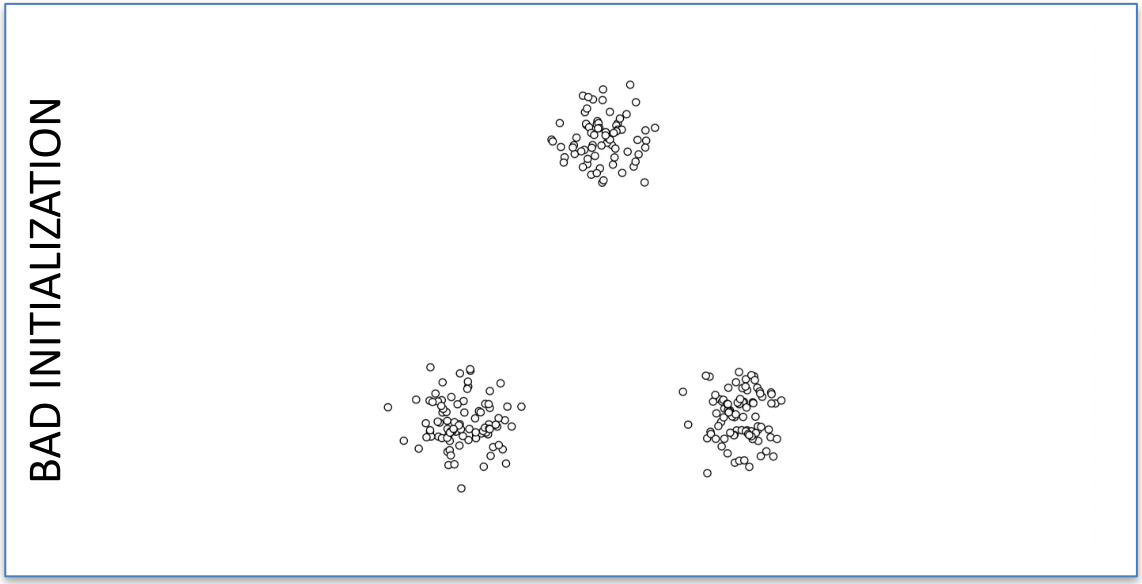
SYMETRICAL ACROSS HEMISPHERES

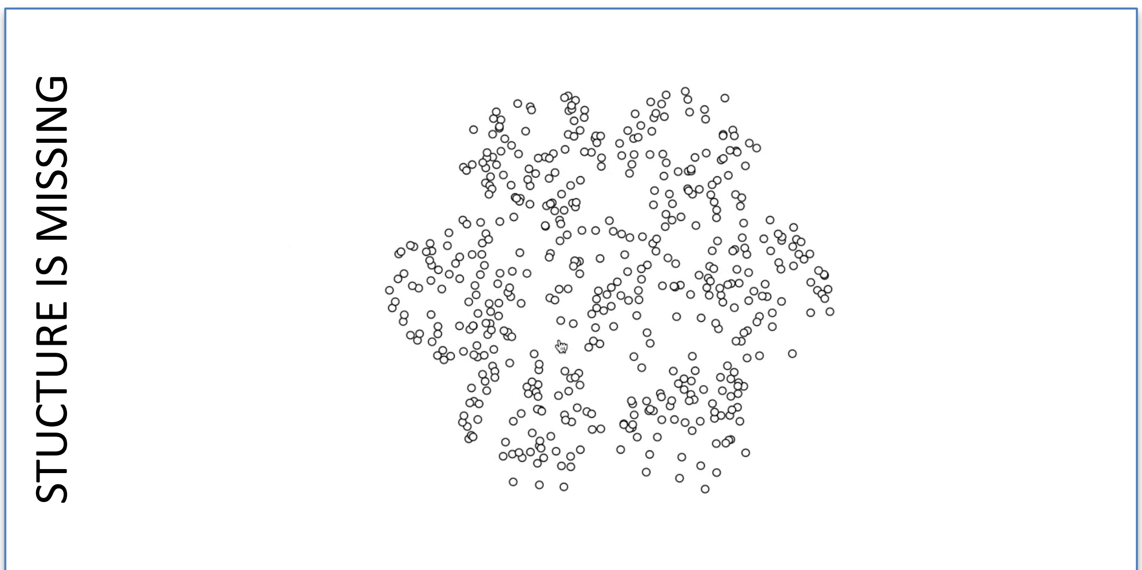
FUNCTIONALLY & ANATOMICALLY MEANINGFUL

REPRODUCIBLE PARCELLATION ACROSS SUBJECTS

Gonzalez-Castillo et al., PNAS 2012

K-Means Interpretation Issues

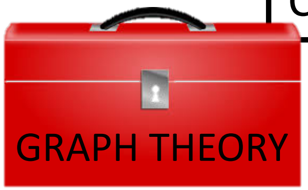




https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Introduction / Agenda

EXPERIMENTAL DESIGN → DATA ACQUISITION → DATA PRE-PROCESSING → GLM – ACTIVATION / FUNCTIONAL CONNECTIVITY



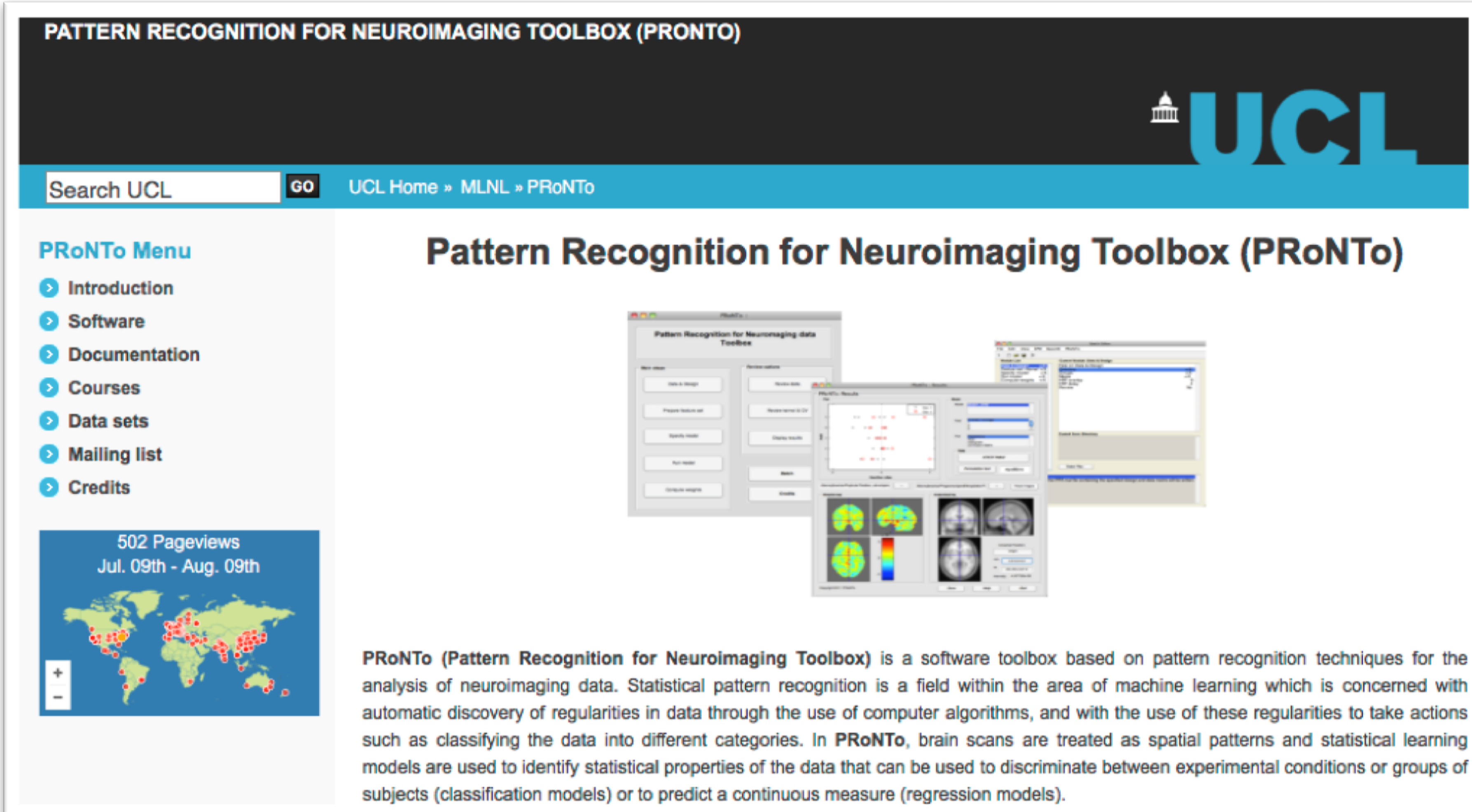
Centrality, Degree, Clustering Coefficient, Community, etc.

Logistic Regression, Support Vector Machines, ICA, K-Means, Convolutional Networks, etc.

Cost Function, Learning Rate, Gradient Descend, Decision Boundary, Regularization, etc.

- A few applications to fMRI data.
- A few words on software.
- Additional Resources to learn more.

Toolboxes: PRONTO


PATTERN RECOGNITION FOR NEUROIMAGING TOOLBOX (PRONTO)

UCL

Search UCL GO

UCL Home » MLNL » PRoNTo

PRoNTo Menu

- Introduction
- Software
- Documentation
- Courses
- Data sets
- Mailing list
- Credits

502 Pageviews
Jul. 09th - Aug. 09th

Pattern Recognition for Neuroimaging Toolbox (PRoNTo)

PRoNTo (Pattern Recognition for Neuroimaging Toolbox) is a software toolbox based on pattern recognition techniques for the analysis of neuroimaging data. Statistical pattern recognition is a field within the area of machine learning which is concerned with automatic discovery of regularities in data through the use of computer algorithms, and with the use of these regularities to take actions such as classifying the data into different categories. In **PRoNTo**, brain scans are treated as spatial patterns and statistical learning models are used to identify statistical properties of the data that can be used to discriminate between experimental conditions or groups of subjects (classification models) or to predict a continuous measure (regression models).

Run on MATLAB

TOOLBOX WEBSITE: http://www.mlnl.cs.ucl.ac.uk/pronto/index.html

Toolboxes: The Decoding Toolbox

Welcome to TDT – The Decoding Toolbox

--- UPDATE ---
TDT version 3.98 with simpler crossnobis distance estimation (multivariate encoding).
TDT version 3.97 with prevalence analysis for valid group-level analysis now available.

TDT (download) is an easy to use, fast and versatile Matlab toolbox for the multivariate analysis of functional and structural MRI data. It contains searchlight, region-of-interest, and whole-brain analyses, as well as many feature selection and parameter selection methods including recursive feature elimination. More recent versions allow fast and efficient representational similarity analysis in a regression framework. The toolbox is optimized for the use with SPM or AFNI and can be used with minimal or no programming experience. A simple decoding analysis can be conducted in just one line of code or in SPM with a simple graphical user interface. At the same time, for people with a little programming background in Matlab the full functionality can be exploited very easily, and new features can be added without problem.

The key benefits of TDT are:

- **Accessibility:** If you classify on betas created with SPM or did run-wise deconvolutions in AFNI, you can get to run your first decoding analysis with decoding_example in minutes, and with almost no programming experience with decoding_tutorial in less than 10 minutes
- **Speed:** This is probably one of the fastest toolboxes out, with an SVM-based searchlight on runwise beta estimates, two classes and 100.000 voxels completed in 3-5 minutes. Speed is of essence if you want to quickly inspect your results.
- **Experience:** Originally the toolbox was created in 2008 and continuously improved to be released to the public only in 2014. This means that you can trust the core functionality (but as with any tool: no guarantee ;).
- **Error management:** We spend a lot of time on optimizing error management, i.e. we prevent you from making many mistakes (e.g. non-independence) and you get informative feedback and not some cryptic error message. If you do get an error message you don't understand - contact us so we can fix it.
- **Readability:** We try to make code accessible and easy to follow and it should be no problem to extend the toolbox for your own classifier or method.

For more details and description and a basic tutorial with example code, please consult our publication: *Martin N Hebart*, Kai Görgen* and John-Dylan Haynes (2015). The Decoding Toolbox (TDT): A versatile software package for multivariate analyses of functional imaging data. Front. Neuroinform. 8:88. doi: 10.3389/fninf.2014.00088. *equal contribution.*

Download: Click here to download TDT or fill out the form below for immediate access.
Getting started: We believe that no tutorial is necessary, the toolbox should be self-explanatory. Just look at the README.txt in the decoding_toolbox folder, or consider our publication as reference.
Questions: Please use the TDT mailing list (please also check the list archive).
Example dataset: We have made an example dataset for one subject available (SPM.mat and betas, ROIs, structural image and description; a lower resolution version (18MB) is available here). If you are interested in pre-processing the data yourself, we also provide DICOM files for subject 1, for subject 2, and a batch script for preprocessing in SPM8. This is not a published study, data were acquired only for illustrating the use of TDT.

Happy decoding!
Kai & Martin

- Runs over MATLAB
- Works on both functional and anatomical datasets
- Works well with SPM and AFNI datasets
- Fast implementation of common linear classifiers (e.g. SVM, LDA, Logistic Regression)
- **One developer works here @ NIH: Martin Hebart**
- **Local workshop in November (9th & 10th /Right before sfn)**
    - REGISTRATION: https://goo.gl/forms/CwWUqqTV9vTSrmcH3

TOOLBOX WEBSITE: https://sites.google.com/site/tdtdecodingtoolbox/

Toolboxes: MATLAB Environment

AFNI MATLAB TOOLS

or

SPM

MATLAB

+

Statistics and Machine Learning Toolbox

+

Neuronal Network Toolbox

Toolboxes: Python Environment

Basic Python Environment:

- ***Numpy***: Basic Matrix and Numerical Capabilities
- ***Scipy***: eco-system for mathematic, science, engineering
- ***Matplotlib***: 2D and 3D figures
- ***Seaborn, bokeh***: Interactive, advance figure capabilities

NeuroImaging Specific:

- ***Nibabel***: read/write access to common Neuroimaging file formats.
- ***Nipype***: pre-processing pipelines for Neuroimaging data.

Machine Learning:

- ***Nilearn***: Machine Learning for neuroimaging data/visualization
- ***Scikit-learn***: Machine Learning in Python

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

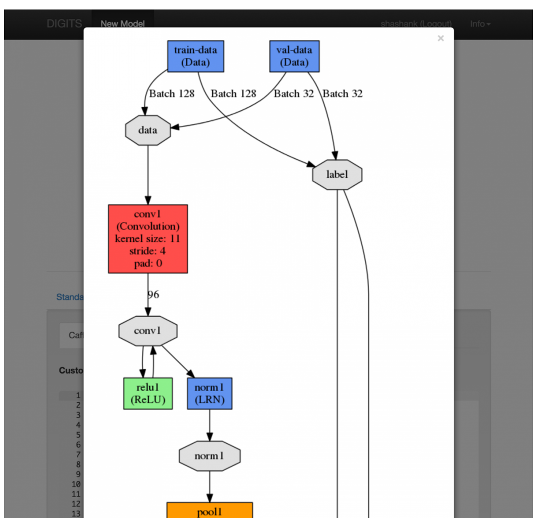
Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

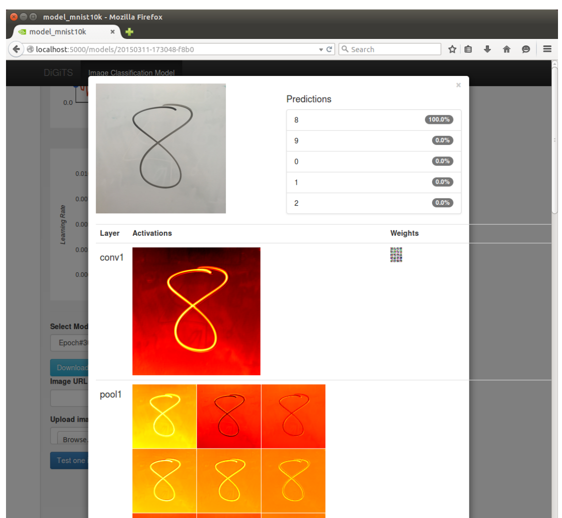
Deep Learning – Model Definition & Training:

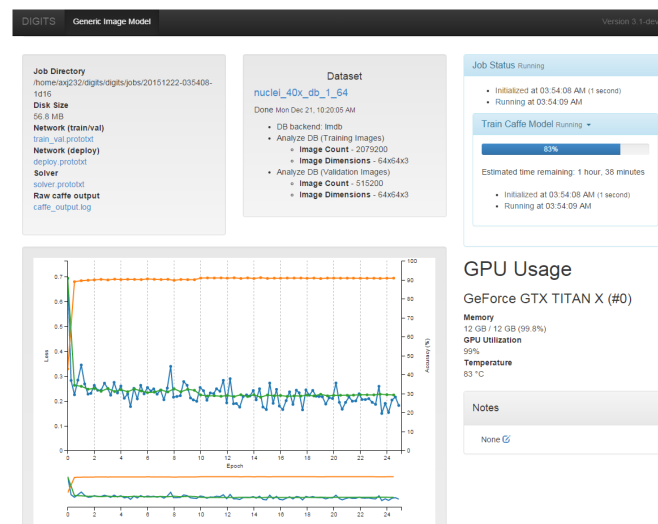
- Theano
- Tensor-flow

Toolboxes: NVIDIA Digits

- Design, train and visualize deep neural networks for image classification, segmentation and object detection.
- Easy access to pre-trained models.
- Schedule, monitor, and manage neural network training jobs, and analyze accuracy and loss in real time.
- Scale training jobs across multiple GPUs automatically.
- Available at NIH – HPC: https://hpc.nih.gov/apps/digits.html
- Simple Web-based GUI.

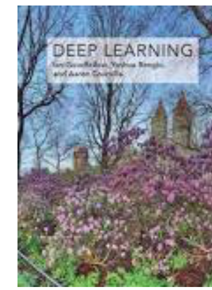






Additional Resources

- **Online Materials:**
  - Coursera Course: Machine Learning by Andrew Ng
  - Udacity Course: Deep Learning by Google
  - NVIDIA: Basic Tutorials on DNN & DIGITS 5.0



- **Online & Hardcover Book:**
  - Deep Learning by Ian Goodfellow, Yoshua Bengio & Aaron Courville

- **Here @ NIH:**

  - NIMH – Machine Learning Core: http://cmn.nimh.nih.gov/mlt
    Contacts: Adam Thomas, Charles Zheng

  - NIMH – Data Science and Sharing Team: https://cmn.nimh.nih.gov/dsst
    Contact: Adam Thomas

  - Special Interest Group on Deep Learning
    Contact: Sunbin Song

  - Special Interest Group on Machine Learning & Brain Imaging
    Contact: Javier Gonzalez-Castillo

Upcoming Talks in the Machine Learning & Brain Imaging Series

Date	Speaker/Topic
September, 2017	**Dr. Jessica Schrouff, *UCL, London, UK*** Multiple Kernel Learning for ML modeling of neuroimaging and electrophysiological data
October, 2017	**Dr. Gael Varoquaux, *NeuroSpin, France*** Machine Learning for Cognitive Neuro-Imaging
November, 2017	**Dr. Jonas Richiardi, *Lausane University Hospital, Switzerland*** Graph-based inference and prediction for NeuroImaging
December, 2017	**Dr. Chris Baker, *Laboratory of Brain and Cognition, NIMH*** TBD
January, 2018	**Dr. Yoshua Bengio, *Montreal University, CA*** Towards biologically plausible Deep Learning
February, 2018	**Dr. Adam Marblestone, *MIT, Cambridge, MA*** Towards integration of Deep Learning and Neuroscience
March, 2018	**Dr. Niko Kriegeskorte, *Columbia University, NY*** Modeling brain processing with Deep Learning + Representational Similarity Analysis
April, 2018	**Dr. Aude Oliva, *MIT, Cambridge, MA*** Comparison of DNNs to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence
May, 2018	**Dr. Josh Tenenbaum, *MIT, Cambridge, MA*** Human-concept learning through probabilistic program induction
June, 2018	**Dr. Marcel Van Gerven, *Donders Institute, Nijmegen, Netherlands*** Encoding and decoding of neural representations with artificial neural networks
July, 2018	**Dr. Vince Calhoun, *MIND Research Institute, NM*** Deep Learning for Classification of Patient Populations

NIH National Institute of Mental Health

Acknowledgements

Section on Functional Imaging Methods

Peter A. Bandettini
Daniel A. Handwerker
Peter Molfese
Prantik Kundu
Dave Jangraw
Laurentius Huber
Natasha Topolski
Andrew Hall

Scientific and Statistical Computing Core

Robert W. Cox
Paul Taylor
Daniel Glen
Richard Reynolds
Gang Chen

Advanced MRI

Catie Chang

Advanced MRI
LFMI, NINDS,
National Institutes of Health

Functional MRI Facility

Sean Marrett
Vinai Roopchansingh
Souheil Inati
Andy Derbishire

FMRIF
fmrif.nimh.nih.gov