The 8th International Conference on Brain Inspired Cognitive Systems (BICS 2016)

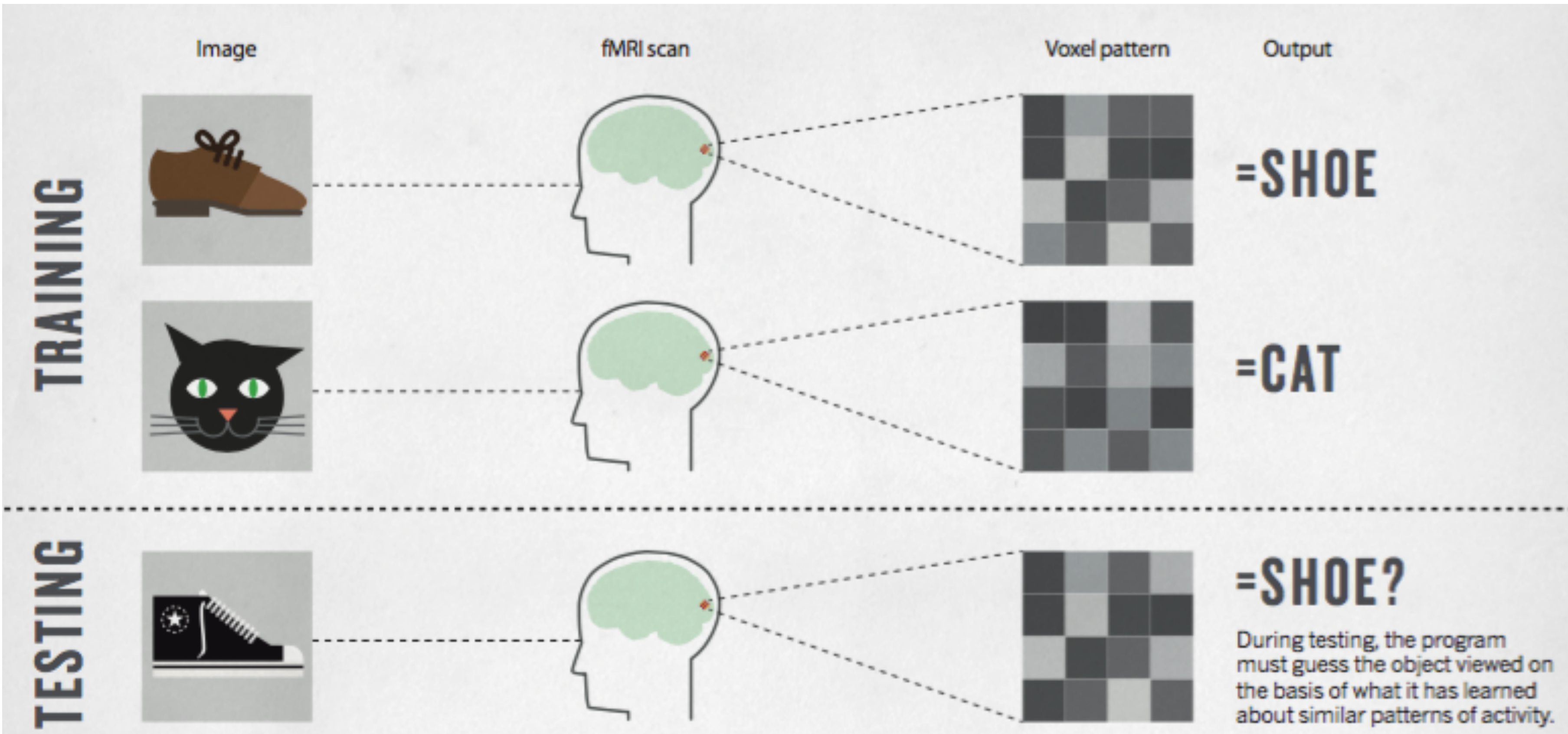
Decoding visual stimuli in human brain by using Anatomical Pattern Analysis on fMRI images

Muhammad Yousefnezhad, Daoqiang Zhang

Presented by: Muhammad Yousefnezhad

College of Computer Science & Technology
Nanjing University of Aeronautics and Astronautics

Motivation



[Smith, Nature, 2013]

Outline

1 Preliminaries

2 The Proposed Method

2.1 Feature Extraction

2.2 Classification Algorithm

3 Experiments

4 Conclusion

Modalities of Measurement

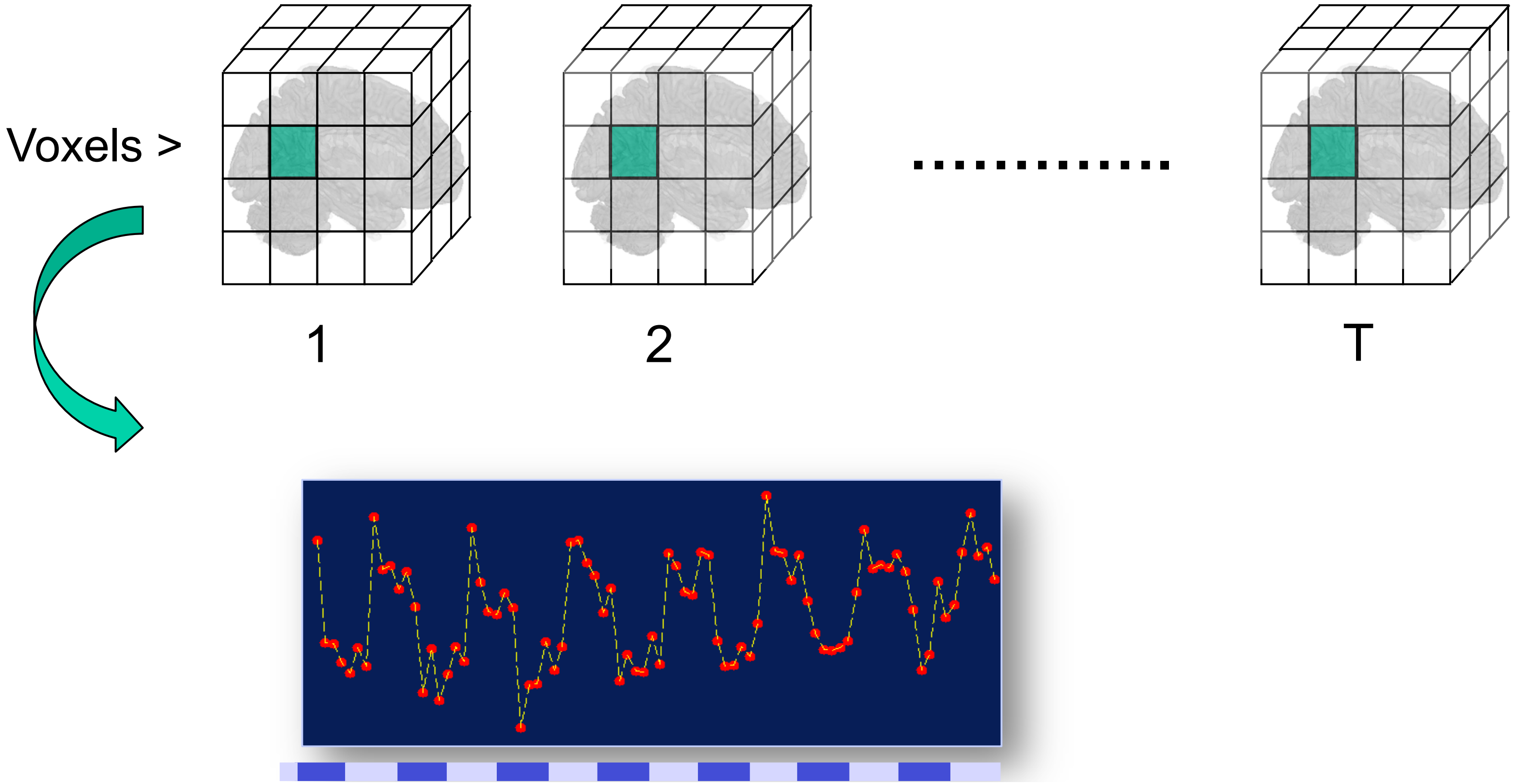
- Modalities of measure:
  - ✓ Single-unit recording
  - ✓ Electrocorticography (ECoG)
  - ✓ electroencephalography (EEG)
  - ✓ Magnetoencephalographic (MEG)
  - ✓ functional Magnetic Resonance Imaging (fMRI)

- The spatial resolution of fMRI allowed investigators to ask what information is represented in a region instead of asking what a region's function is?

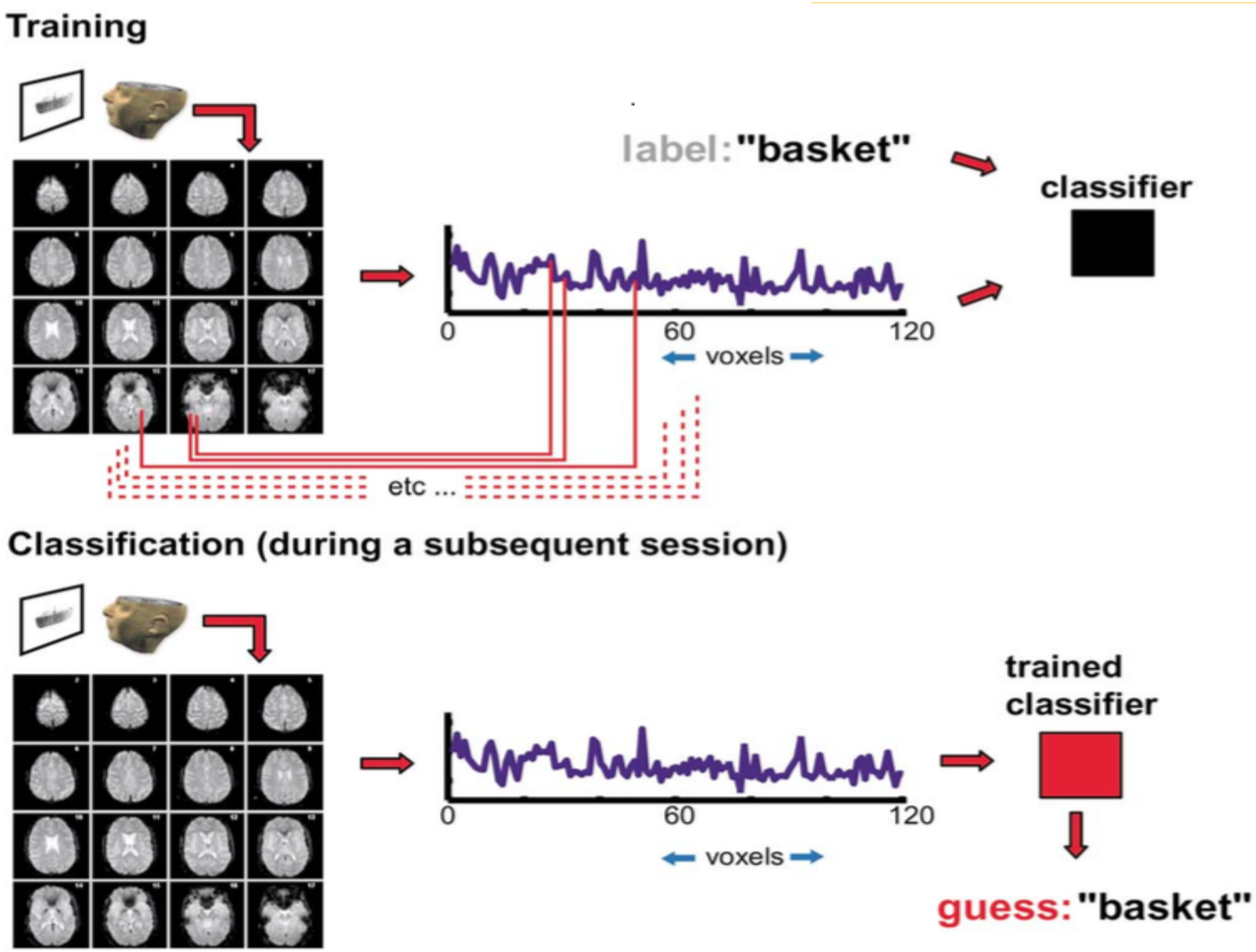
- Our method contributions are:
  - ✓ Automatically detecting the Region of Interests (ROIs)
  - ✓ A new feature representation for removing noise and sparsity
  - ✓ A customized classification algorithm for brain decoding problems

fMRI Data

- Tracking the intensity over time gives us a time series.



Multivariate Pattern Classification (MVP)

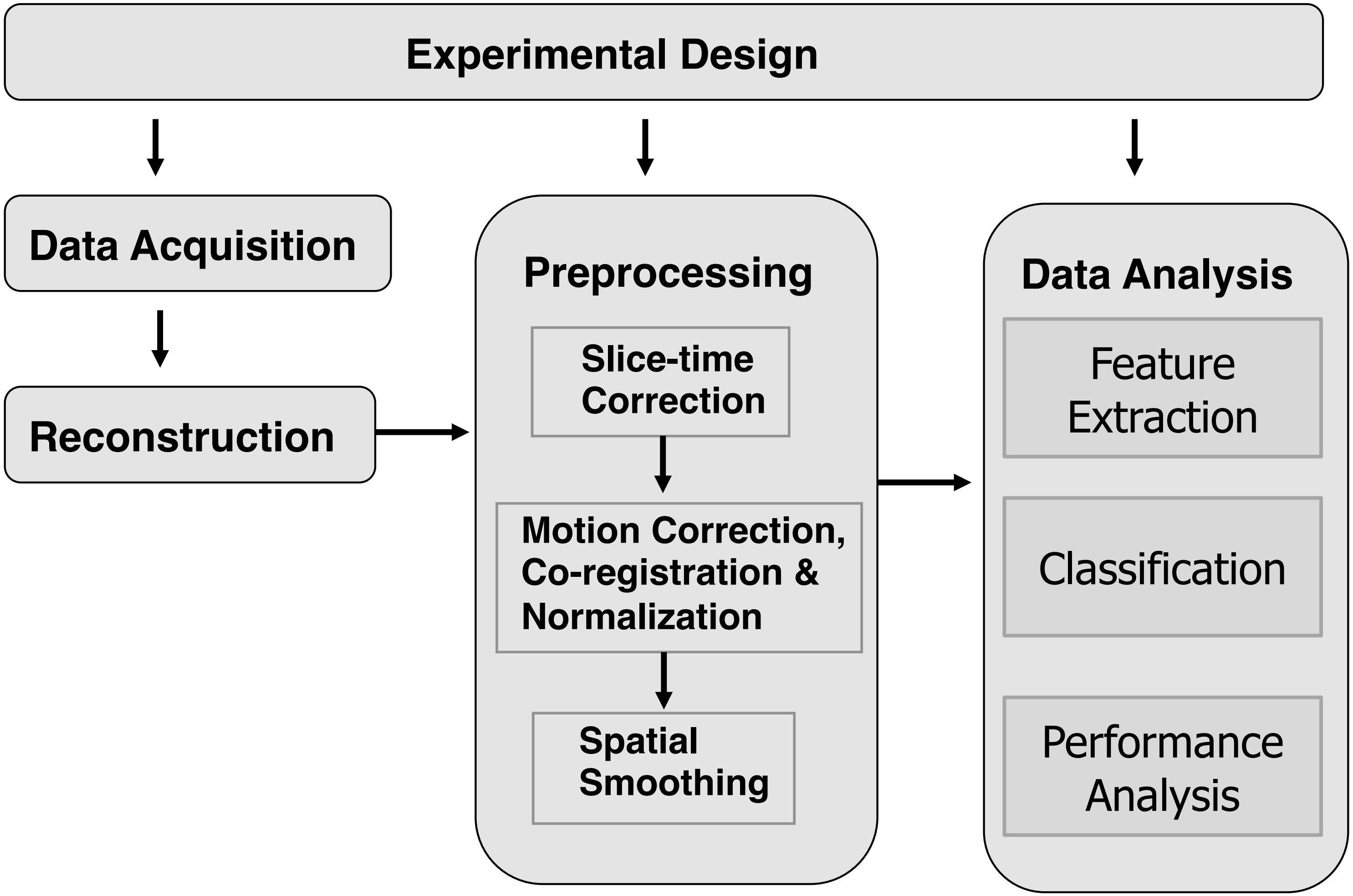


[Cox et al. 2003]

Decoding visual stimuli in human brain by using Anatomical Pattern Analysis on fMRI images

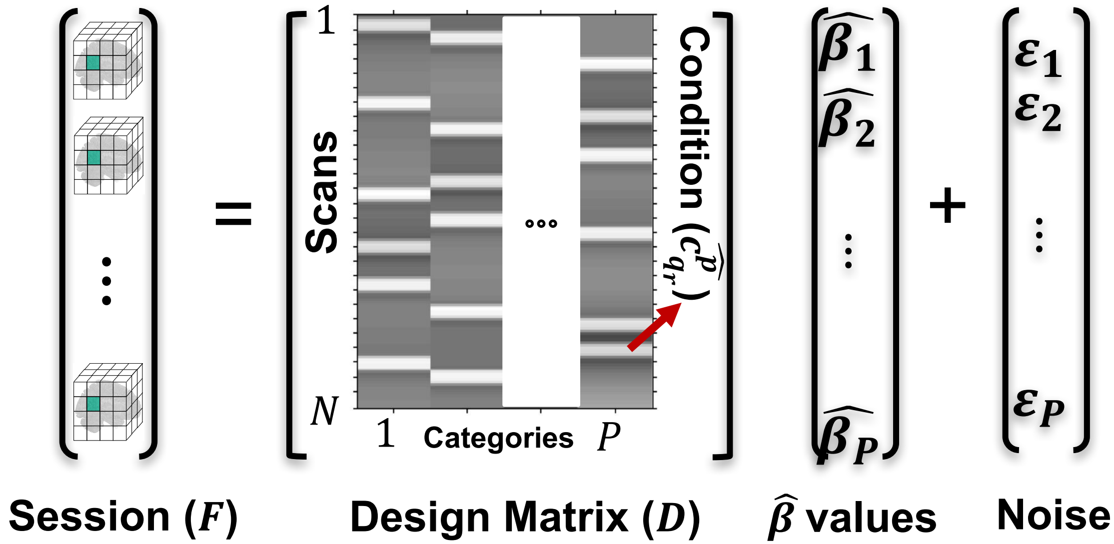
Outline

General framework of study



Outline

General Linear Model for Each Session



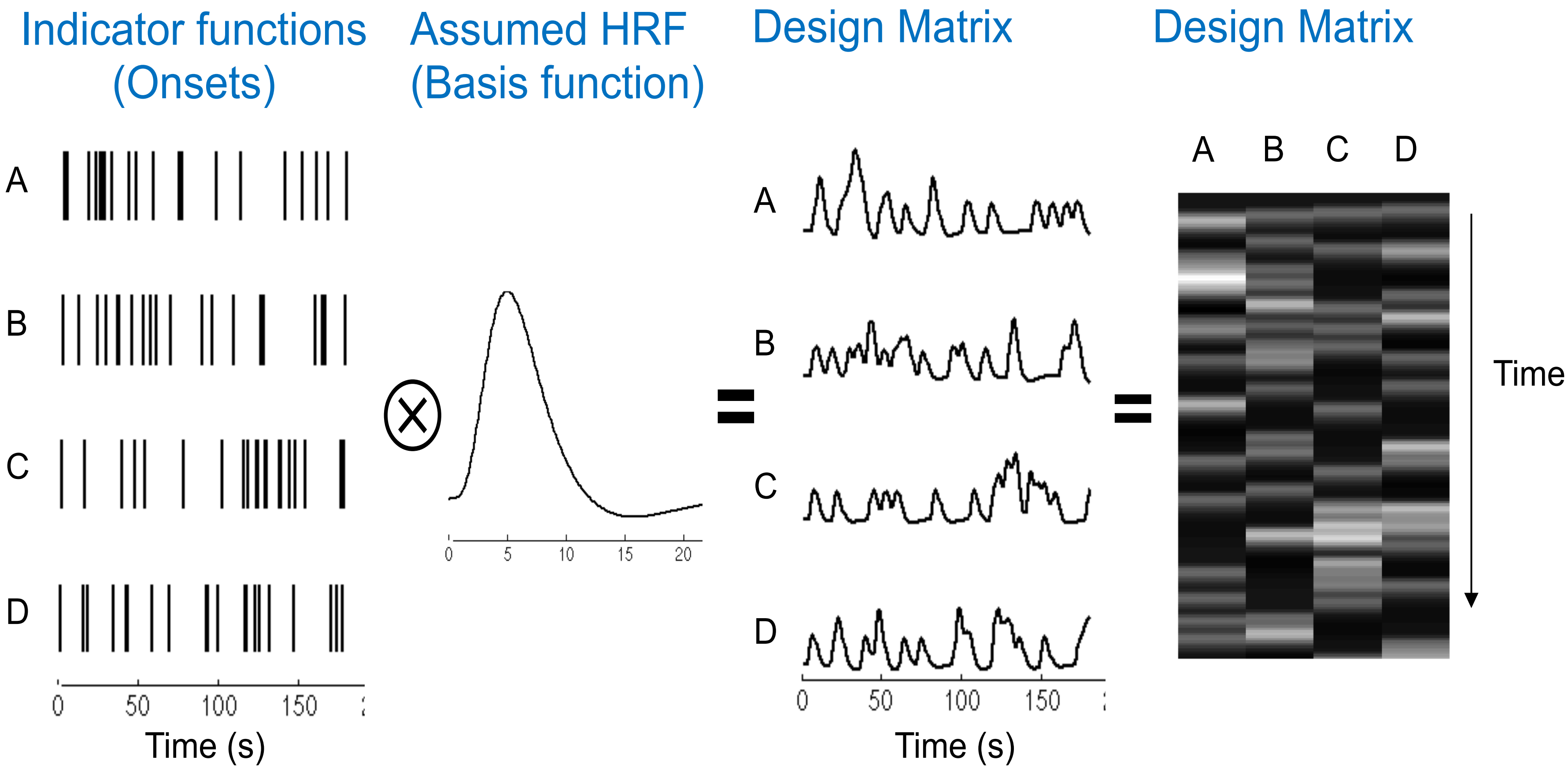
$$F = D\hat{\beta} + \varepsilon$$

- This paper uses Generalized Least Squares (GLS) approach for estimating optimized solution:

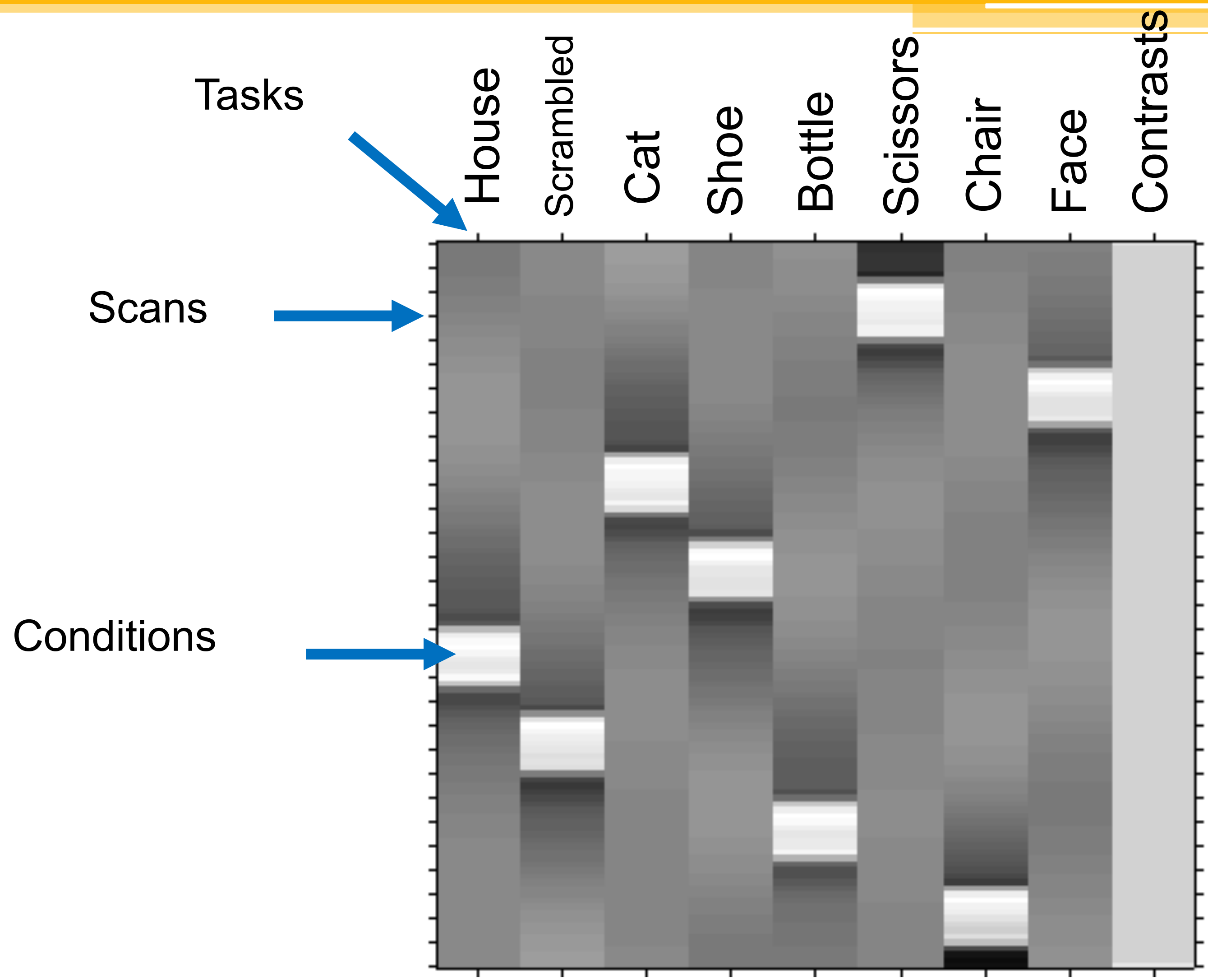
$$\hat{\beta} = (D^\intercal V^{-1} D)^{-1} D^\intercal V^{-1} F$$

$$Var(\varepsilon) = V\sigma^2 \neq \mathbb{I}\sigma^2$$

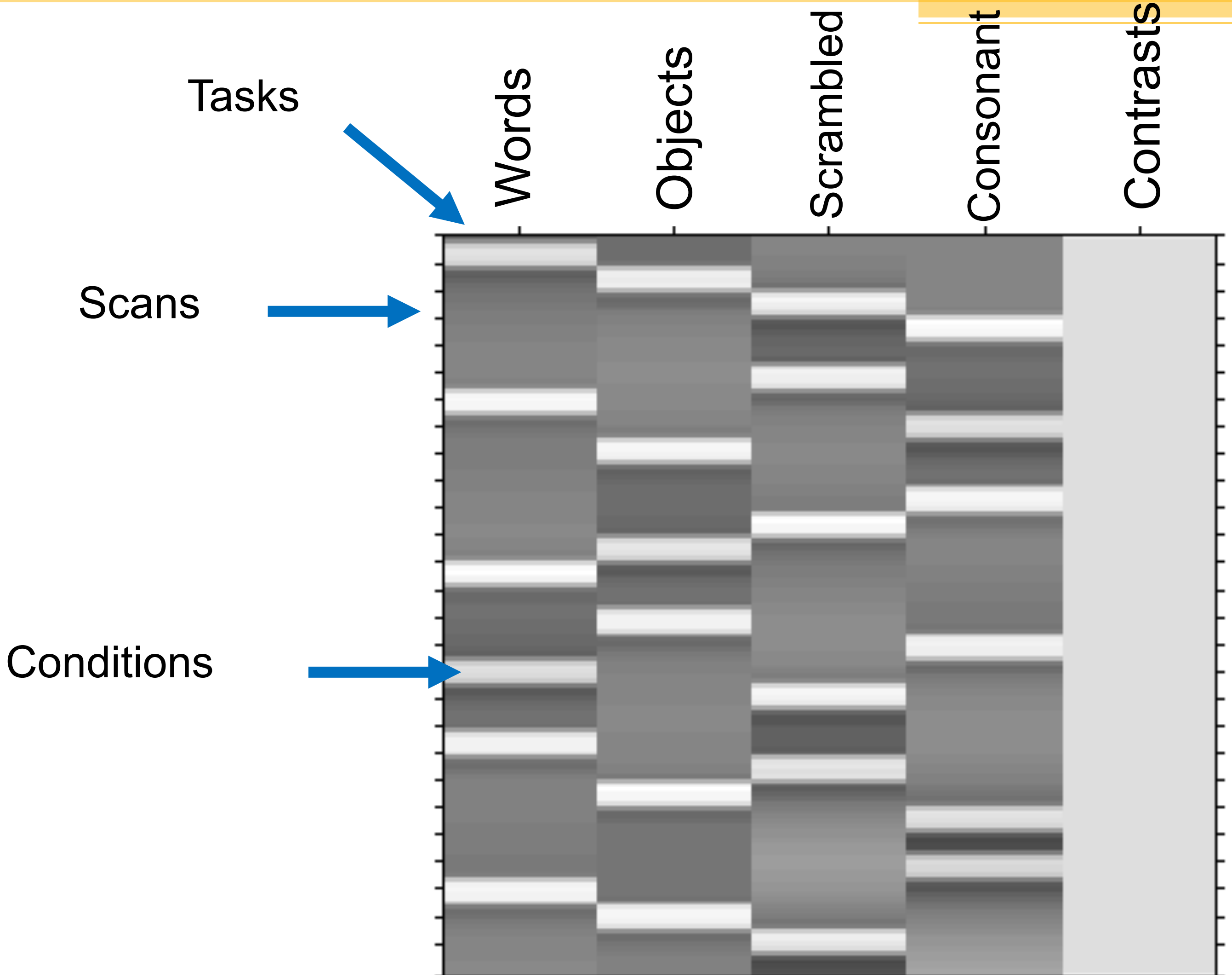
The first level analysis



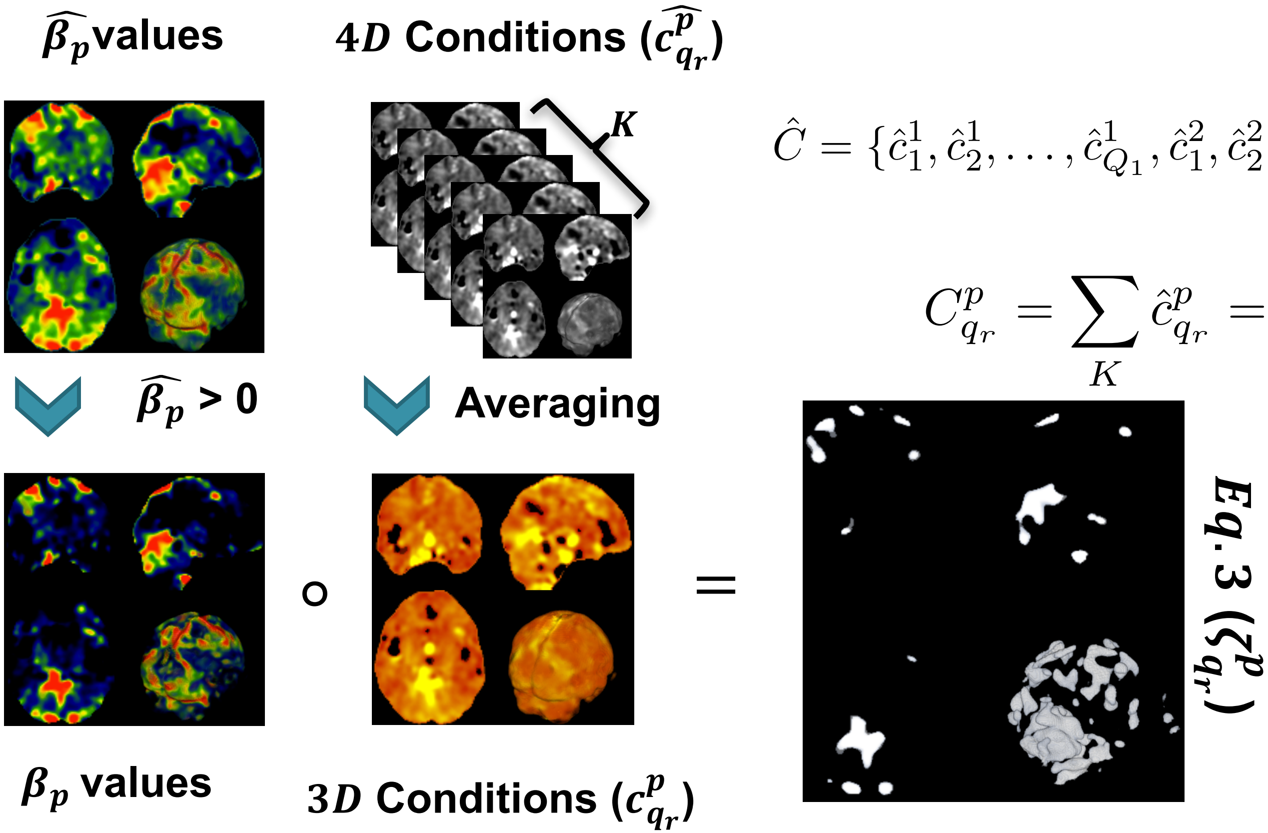
Example of Design Matrix



Example of Design Matrix (cont.)



Feature Extraction

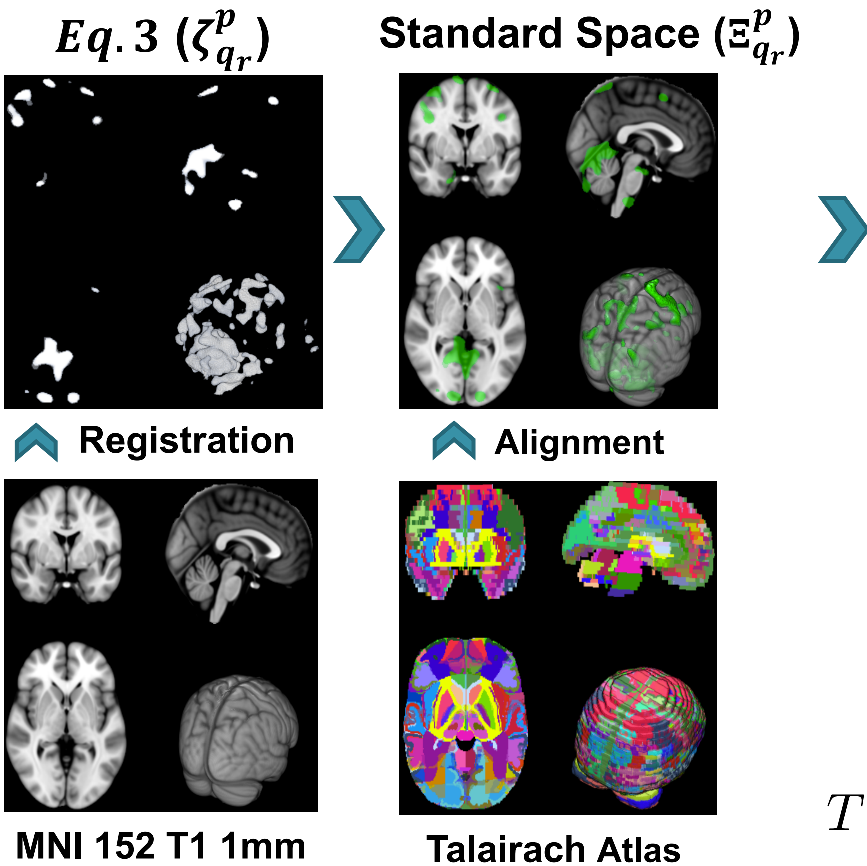


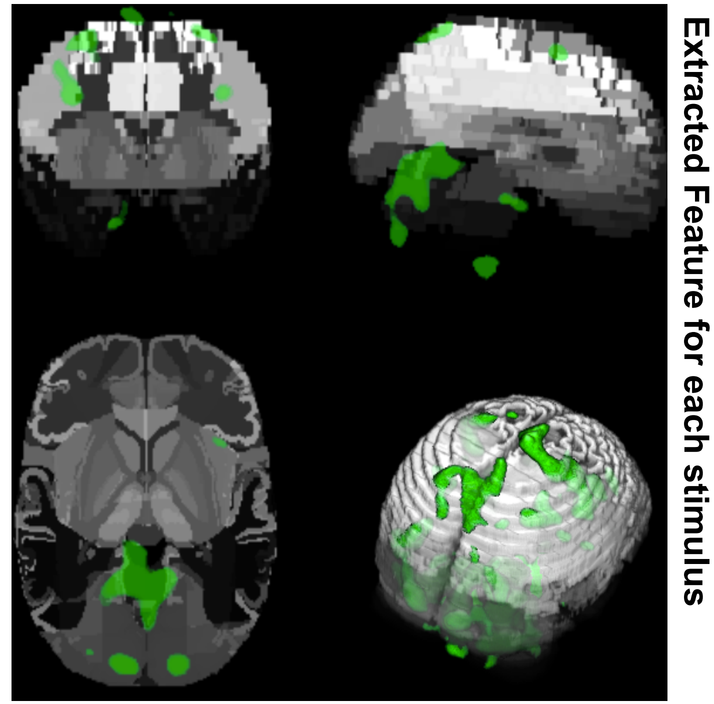
$$\hat{C} = \{\hat{c}^1_1, \hat{c}^1_2, \dots, \hat{c}^1_{Q_1}, \hat{c}^2_1, \hat{c}^2_2, \dots, \hat{c}^2_{Q_2}, \dots, \hat{c}^p_1, \hat{c}^P_2, \dots, \hat{c}^P_{Q_R}\}$$

$$C^p_{q_r} = \sum_K \hat{c}^p_{q_r} = \sum_{k=1}^{K^p_{q_r}} \hat{c}^p_{q_r}[k, :, :, :]$$

$$\zeta^p_{q_r} = \beta_p \circ C^p_{q_r} = \{\forall [x, y, z] \in C^p_{q_r} \implies (\zeta^p_{q_r})_{[x,y,z]} = (\beta_p)_{[x,y,z]} \times (C^p_{q_r})_{[x,y,z]}\} \quad (3)$$

Feature Extraction (cont.)





$$T^* = argmin_{T \in S_T}(NMI(Ref, \Xi^p_{q_r})) \quad (4)$$

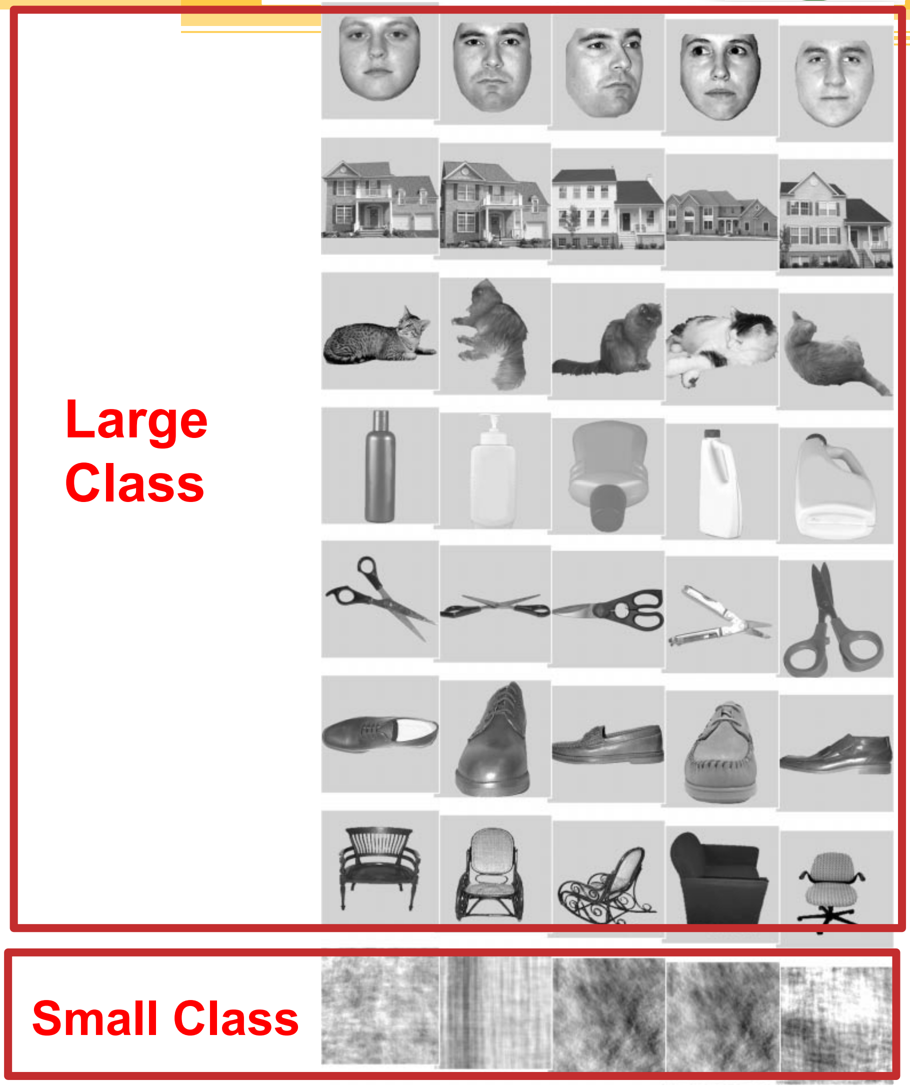
$$\forall a_v = [x_v, y_v, z_v] \in A_l \implies \Gamma^p_{q_r}(l) = \frac{1}{\mid A_l \mid} \sum_{v=1}^{|A_l|} (\Xi^p_{q_r})[a_v] = \frac{1}{\mid A_l \mid} \sum_{v=1}^{|A_l|} (\Xi^p_{q_r})[x_v, y_v, z_v] \quad (5)$$

Outline

One Vs. All (OVA) strategy

- Most of previous studies used One vs. All (OVA) strategy for training binary classification.

- While (non-)linear SVM is mostly used for creating classification method, the OVA strategy generate an imbalance sampling.



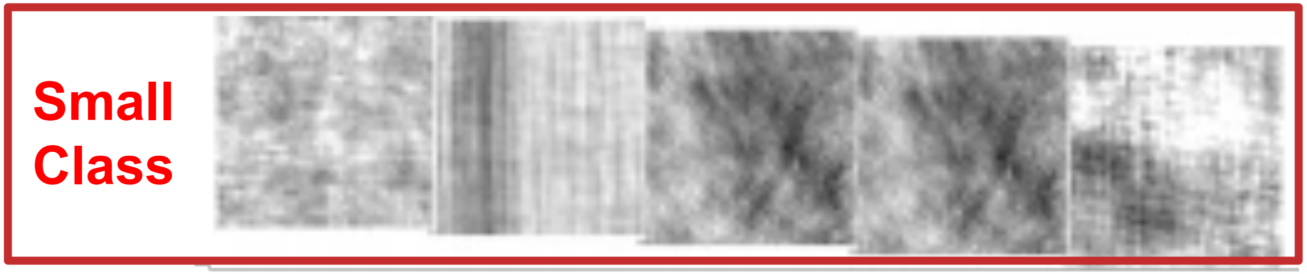
Ensemble Approach

- Each iteration contains all samples of small class, randomized selected samples from large class, and the samples that made errors in the pervious iteration.
- The number of randomized selected samples from large class is equal to the number of samples in the small class.
- Randomize sampling is without replacement.

Ensemble Approach

- Each iteration contains all samples of small class, randomized selected samples from large class, and the samples that made errors in the pervious iteration.
- The number of randomized selected samples from large class is equal to the number of samples in the small class.
- Randomize sampling is without replacement.

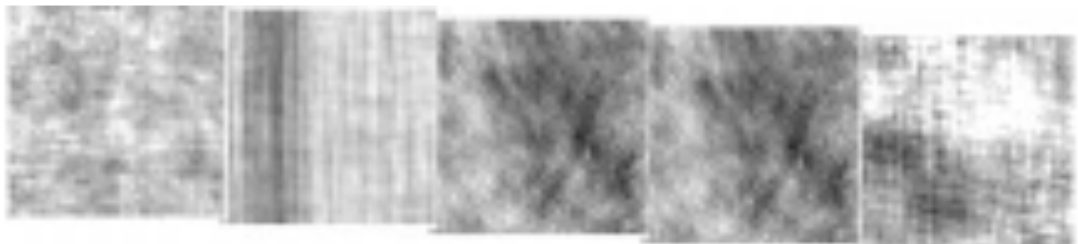
Iteration #1



Ensemble Approach

- Each iteration contains all samples of small class, randomized selected samples from large class, and the samples that made errors in the pervious iteration.
- The number of randomized selected samples from large class is equal to the number of samples in the small class.
- Randomize sampling is without replacement.

Iteration #1

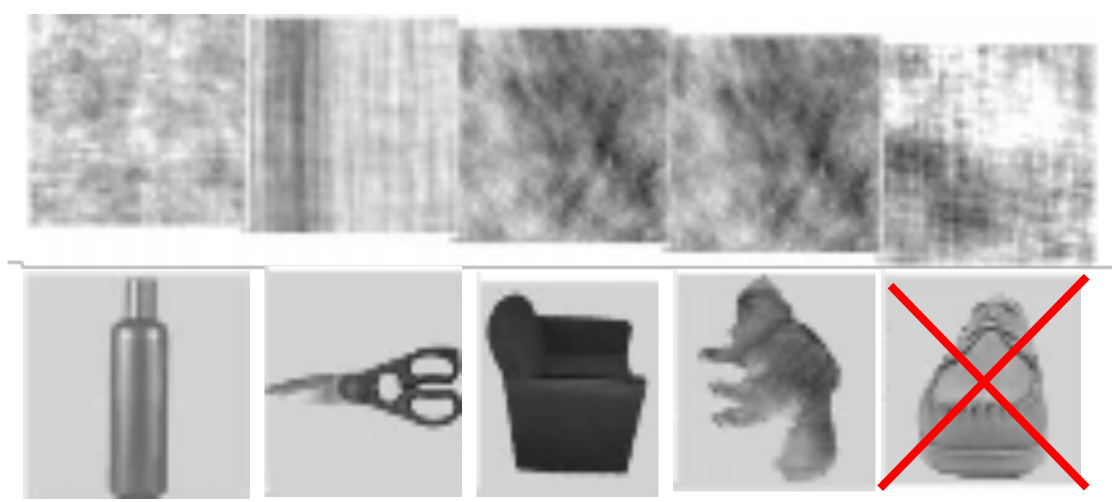


Selected Large Class

Ensemble Approach

- Each iteration contains all samples of small class, randomized selected samples from large class, and the samples that made errors in the pervious iteration.
- The number of randomized selected samples from large class is equal to the number of samples in the small class.
- Randomize sampling is without replacement.

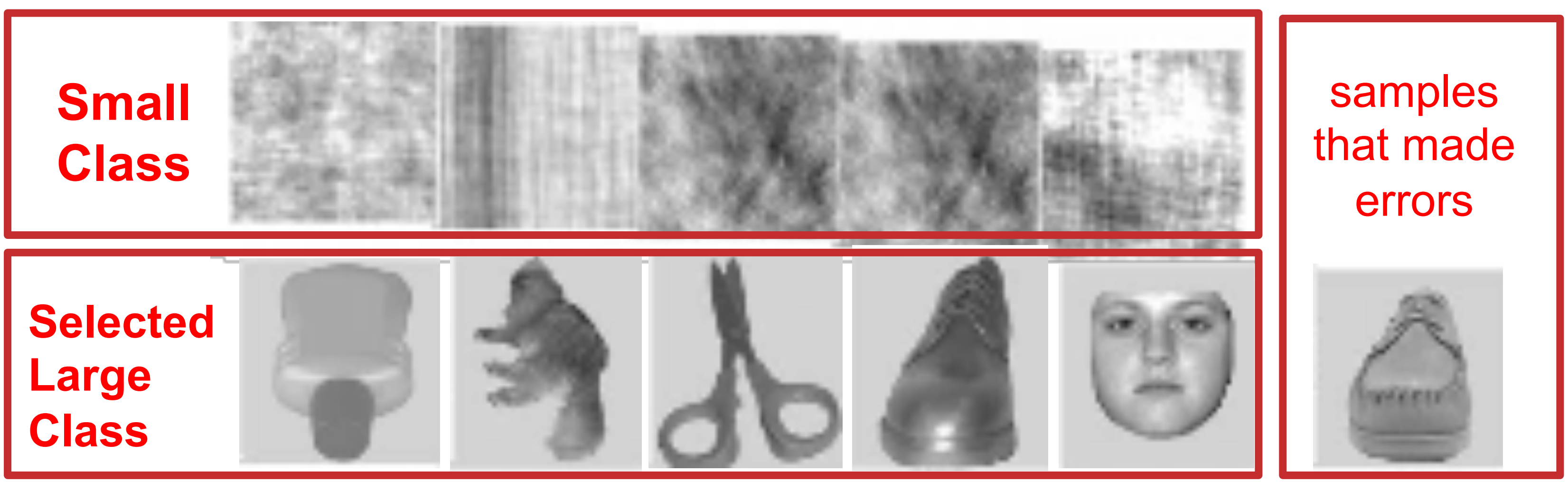
Iteration #1



Ensemble Approach

- Each iteration contains all samples of small class, randomized selected samples from large class, and the samples that made errors in the pervious iteration.
- The number of randomized selected samples from large class is equal to the number of samples in the small class.
- Randomize sampling is without replacement.

Iteration #2



A modified version of AdaBoost algorithm

Algorithm 1 The proposed binary classification algorithm

Input: Data set G_{tr} : is train set, I_{tr} : denotes real class labels of the train sets,
Output: Classifier E,
Method:

1. Partition $G_{tr} = \{G_{tr}^S, G_{tr}^L\}$, where G_{tr}^S, G_{tr}^L are Small and Large classes.
2. Calculate $J = Int(\mid G_{tr}^S \mid / \mid G_{tr}^L \mid)$ based on number of elements in classes.
3. Randomly sample the $G_{tr}^L = \{G_{tr}^L(1), \ldots, G_{tr}^L(J)\}$.
4. By considering $\bar{G}_1 = \bar{I}_1 = \emptyset$, generating $j = 1, \ldots, J+1$ classifiers:
5. Construct $G_j = \{G_{tr}^S, G_{tr}^L(j), \bar{G}_j\}$ and $I_j = \{I_{tr}^S, I_{tr}^L(j), \bar{I}_j\}$
6. Calculate $W_j = \{w_j\}_{|G_j|} = \begin{cases} 1 & \text{for instances of } G_{tr}^S \text{ or } \bar{G}_j \\ 1 - \mid corr(G_{tr}^S, G_{tr}^L) \mid & \text{for instances of } G_{tr}^L(j) \end{cases}$
7. Train $\theta_j = Classifier(G_j, I_j, W_j)$.
8. Construct $\bar{G}_{j+1}$, $\bar{I}_{j+1}$ as the set of instances cannot truly trained in θ_j.
9. **If** $(j \leq J+1)$: go to line 5; **Else:** return $\Theta_p = \{\theta_1, \ldots, \theta_{J+1}\}$ as final classifier.

A modified version of AdaBoost algorithm

Algorithm 1 The proposed binary classification algorithm

Input: Data set G_{tr} : is train set, I_{tr} : denotes real class labels of the train sets,
Output: Classifier E,
Method:

1. Partition $G_{tr} = \{G_{tr}^S, G_{tr}^L\}$, where G_{tr}^S, G_{tr}^L are Small and Large classes.
2. Calculate $J = Int(\mid G_{tr}^S \mid / \mid G_{tr}^L \mid)$ based on number of elements in classes.
3. Randomly sample the $G_{tr}^L = \{G_{tr}^L(1), \ldots, G_{tr}^L(J)\}$.
4. By considering $\bar{G}_1 = \bar{I}_1 = \emptyset$, generating $j = 1, \ldots, J+1$ classifiers:
5. Construct $G_j = \{G_{tr}^S, G_{tr}^L(j), \bar{G}_j\}$ and $I_j = \{I_{tr}^S, I_{tr}^L(j), \bar{I}_j\}$
6. Calculate $W_j = \{w_j\}_{|G_j|} = \begin{cases} 1 & \text{for instances of } G_{tr}^S \text{ or } \bar{G}_j \\ 1- \mid corr(G_{tr}^S, G_{tr}^L) \mid & \text{for instances of } G_{tr}^L(j) \end{cases}$
7. Train $\theta_j = Classifier(G_j, I_j, W_j)$.
8. Construct $\bar{G}_{j+1}$, $\bar{I}_{j+1}$ as the set of instances cannot truly trained in θ_j.
9. **If** $(j \leq J+1)$: go to line 5; **Else:** return $\Theta_p = \{\theta_1, \ldots, \theta_{J+1}\}$ as final classifier.

A modified version of AdaBoost algorithm

Algorithm 1 The proposed binary classification algorithm

Input: Data set G_{tr} : is train set, I_{tr} : denotes real class labels of the train sets,
Output: Classifier E,
Method:

1. Partition $G_{tr} = \{G_{tr}^S, G_{tr}^L\}$, where G_{tr}^S, G_{tr}^L are Small and Large classes.
2. Calculate $J = Int(\mid G_{tr}^S \mid / \mid G_{tr}^L \mid)$ based on number of elements in classes.
3. Randomly sample the $G_{tr}^L = \{G_{tr}^L(1), \dots, G_{tr}^L(J)\}$.
4. By considering $\bar{G}_1 = \bar{I}_1 = \emptyset$, generating $j = 1, \dots, J+1$ classifiers:
5. Construct $G_j = \{G_{tr}^S, G_{tr}^L(j), \bar{G}_j\}$ and $I_j = \{I_{tr}^S, I_{tr}^L(j), \bar{I}_j\}$

6. Calculate $W_j = \{w_j\}_{|G_j|} = \begin{cases} 1 & \text{for instances of } G_{tr}^S \text{ or } \bar{G}_j \\ 1- \mid corr(G_{tr}^S, G_{tr}^L) \mid & \text{for instances of } G_{tr}^L(j) \end{cases}$

7. Train $\theta_j = Classifier(G_j, I_j, W_j)$.
8. Construct $\bar{G}_{j+1}$, $\bar{I}_{j+1}$ as the set of instances cannot truly trained in θ_j.
9. **If** $(j \leq J+1)$: go to line 5; **Else:** return $\Theta_p = \{\theta_1, \dots, \theta_{J+1}\}$ as final classifier.

A modified version of AdaBoost algorithm

Algorithm 1 The proposed binary classification algorithm

Input: Data set G_{tr} : is train set, I_{tr} : denotes real class labels of the train sets,
Output: Classifier E,
Method:

1. Partition $G_{tr} = \{G^S_{tr}, G^L_{tr}\}$, where G^S_{tr}, G^L_{tr} are Small and Large classes.
2. Calculate $J = Int(\mid G^S_{tr} \mid / \mid G^L_{tr} \mid)$ based on number of elements in classes.
3. $\boxed{\text{Randomly sample the } G^L_{tr} = \{G^L_{tr}(1), \ldots, G^L_{tr}(J)\}.}$
4. By considering $\bar{G}_1 = \bar{I}_1 = \emptyset$, generating $j = 1, \ldots, J+1$ classifiers:
5. Construct $G_j = \{G^S_{tr}, G^L_{tr}(j), \bar{G}_j\}$ and $I_j = \{I^S_{tr}, I^L_{tr}(j), \bar{I}_j\}$
6. Calculate $W_j = \{w_j\}_{|G_j|} = \begin{cases} 1 & \text{for instances of } G^S_{tr} \text{ or } \bar{G}_j \\ 1- \mid corr(G^S_{tr}, G^L_{tr}) \mid & \text{for instances of } G^L_{tr}(j) \end{cases}$
7. Train $\theta_j = Classifier(G_j, I_j, W_j)$.
8. Construct $\bar{G}_{j+1}$, $\bar{I}_{j+1}$ as the set of instances cannot truly trained in θ_j.
9. **If** $(j \leq J+1)$: go to line 5; **Else:** return $\Theta_p = \{\theta_1, \ldots, \theta_{J+1}\}$ as final classifier.

A modified version of AdaBoost algorithm

Algorithm 1 The proposed binary classification algorithm

Input: Data set G_{tr} : is train set, I_{tr} : denotes real class labels of the train sets,
Output: Classifier E,
Method:

1. Partition $G_{tr} = \{G^S_{tr}, G^L_{tr}\}$, where G^S_{tr}, G^L_{tr} are Small and Large classes.
2. Calculate $J = Int(\mid G^S_{tr} \mid / \mid G^L_{tr} \mid)$ based on number of elements in classes.
3. Randomly sample the $G^L_{tr} = \{G^L_{tr}(1), \ldots, G^L_{tr}(J)\}$.
4. By considering $\bar{G}_1 = \bar{I}_1 = \emptyset$, generating $j = 1, \ldots, J+1$ classifiers:
5. Construct $G_j = \{G^S_{tr}, G^L_{tr}(j), \bar{G}_j\}$ and $I_j = \{I^S_{tr}, I^L_{tr}(j), \bar{I}_j\}$
6. Calculate $W_j = \{w_j\}_{|G_j|} = \begin{cases} 1 & \text{for instances of } G^S_{tr} \text{ or } \bar{G}_j \\ 1- \mid corr(G^S_{tr}, G^L_{tr}) \mid & \text{for instances of } G^L_{tr}(j) \end{cases}$
7. Train $\theta_j = Classifier(G_j, I_j, W_j)$.
8. Construct $\bar{G}_{j+1}$, $\bar{I}_{j+1}$ as the set of instances cannot truly trained in θ_j.
9. **If** $(j \leq J+1)$: go to line 5; **Else:** return $\Theta_p = \{\theta_1, \ldots, \theta_{J+1}\}$ as final classifier.

A modified version of AdaBoost algorithm

Algorithm 1 The proposed binary classification algorithm

Input: Data set G_{tr} : is train set, I_{tr} : denotes real class labels of the train sets,
Output: Classifier E,
Method:

1. Partition $G_{tr} = \{G^S_{tr}, G^L_{tr}\}$, where G^S_{tr}, G^L_{tr} are Small and Large classes.
2. Calculate $J = Int(\mid G^S_{tr} \mid / \mid G^L_{tr} \mid)$ based on number of elements in classes.
3. Randomly sample the $G^L_{tr} = \{G^L_{tr}(1), \ldots, G^L_{tr}(J)\}$.
4. By considering $\bar{G}_1 = \bar{I}_1 = \emptyset$, generating $j = 1, \ldots, J+1$ classifiers:
5. Construct $G_j = \{G^S_{tr}, G^L_{tr}(j), \bar{G}_j\}$ and $I_j = \{I^S_{tr}, I^L_{tr}(j), \bar{I}_j\}$

6. Calculate $W_j = \{w_j\}_{|G_j|} = \begin{cases} 1 & \text{for instances of } G^S_{tr} \text{ or } \bar{G}_j \\ 1- \mid corr(G^S_{tr}, G^L_{tr}) \mid & \text{for instances of } G^L_{tr}(j) \end{cases}$

7.Train $\theta_j = Classifier(G_j, I_j, W_j)$.
8. Construct $\bar{G}_{j+1}$, $\bar{I}_{j+1}$ as the set of instances cannot truly trained in θ_j.
9. **If** $(j \leq J+1)$: go to line 5; **Else:** return $\Theta_p = \{\theta_1, \ldots, \theta_{J+1}\}$ as final classifier.

A modified version of AdaBoost algorithm

Algorithm 1 The proposed binary classification algorithm

Input: Data set G_{tr} : is train set, I_{tr} : denotes real class labels of the train sets,
Output: Classifier E,
Method:

1. Partition $G_{tr} = \{G_{tr}^S, G_{tr}^L\}$, where G_{tr}^S, G_{tr}^L are Small and Large classes.
2. Calculate $J = Int(\mid G_{tr}^S \mid / \mid G_{tr}^L \mid)$ based on number of elements in classes.
3. Randomly sample the $G_{tr}^L = \{G_{tr}^L(1), \ldots, G_{tr}^L(J)\}$.
4. By considering $\bar{G}_1 = \bar{I}_1 = \emptyset$, generating $j = 1, \ldots, J+1$ classifiers:
5. Construct $G_j = \{G_{tr}^S, G_{tr}^L(j), \bar{G}_j\}$ and $I_j = \{I_{tr}^S, I_{tr}^L(j), \bar{I}_j\}$
6. Calculate $W_j = \{w_j\}_{|G_j|} = \begin{cases} 1 & \text{for instances of } G_{tr}^S \text{ or } \bar{G}_j \\ 1- \mid corr(G_{tr}^S, G_{tr}^L) \mid & \text{for instances of } G_{tr}^L(j) \end{cases}$
7. Train $\theta_j = Classifier(G_j, I_j, W_j)$.
8. Construct $\bar{G}_{j+1}$, $\bar{I}_{j+1}$ as the set of instances cannot truly trained in θ_j.
9. **If** $(j \leq J+1)$: go to line 5; **Else:** return $\Theta_p = \{\theta_1, \ldots, \theta_{J+1}\}$ as final classifier.

Multi-class Approach

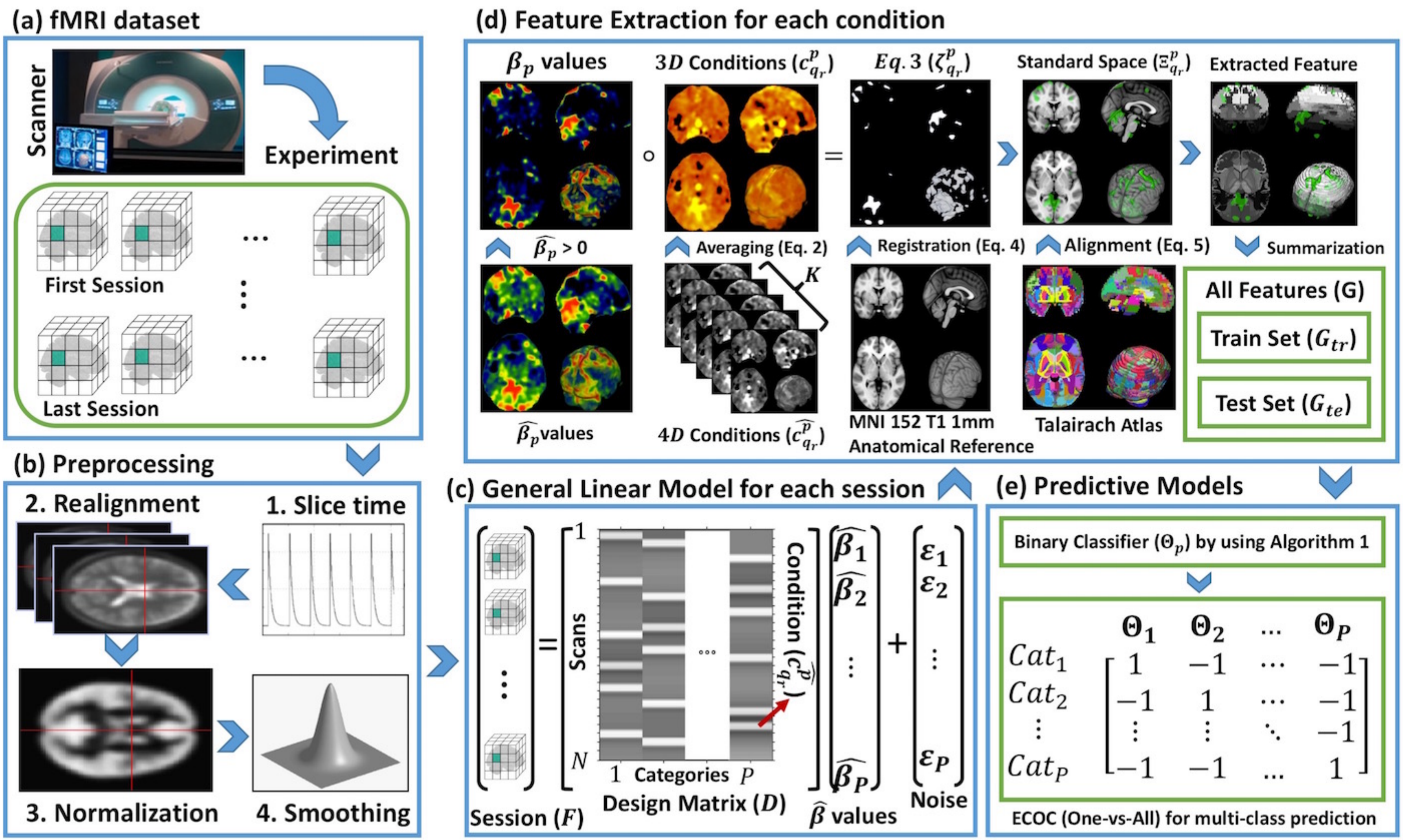
- This paper utilizes Error-Correcting Output Codes (ECOC) method for applying multi-class classification.

- Our method uses a one-versus-all encoding strategy for training the ECOC method, where an independent category of the visual stimuli is compared with the rest of categories.

- Indeed, the number of classifiers in this strategy is exactly equal to the number of categories.

- This method also assigns the brain response to the category with closest hamming distance in decoding stage.

Summery



Decoding visual stimuli in human brain by using Anatomical Pattern Analysis on fMRI images

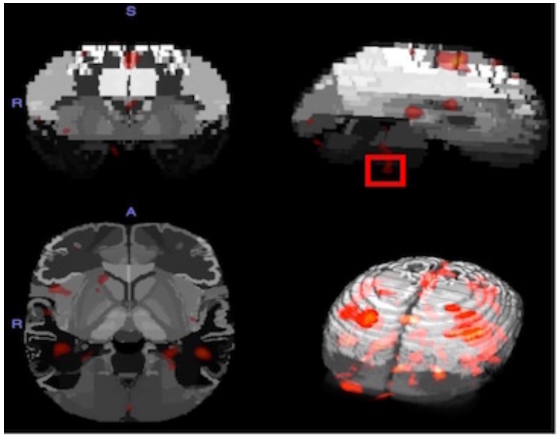
Outline

Data Sets

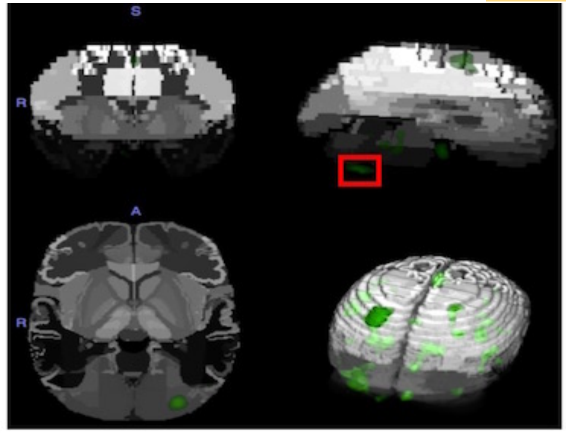
- This paper employs two datasets, shared by openfmri.org, for running empirical studies.

Title	ID	# of Subjects	# of Samples	# of Stimuli
Visual Object Recognition	DS105	6	568	8
Word and Object Processing	DS107	49	1568	4

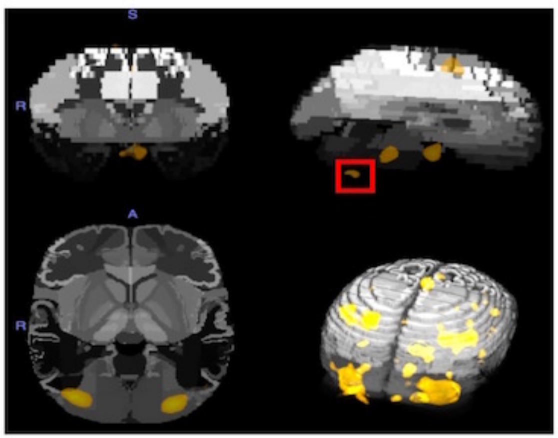
The Error of Registration



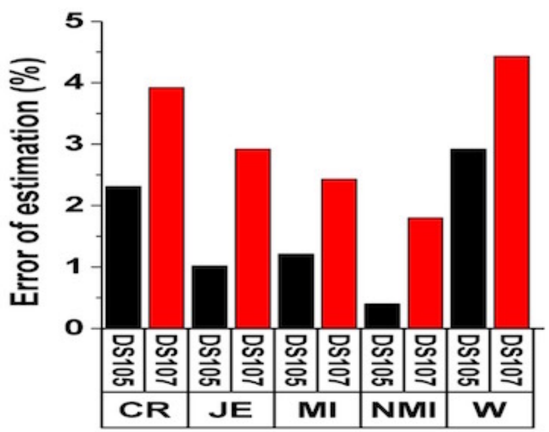
(a) Word



(b) Object



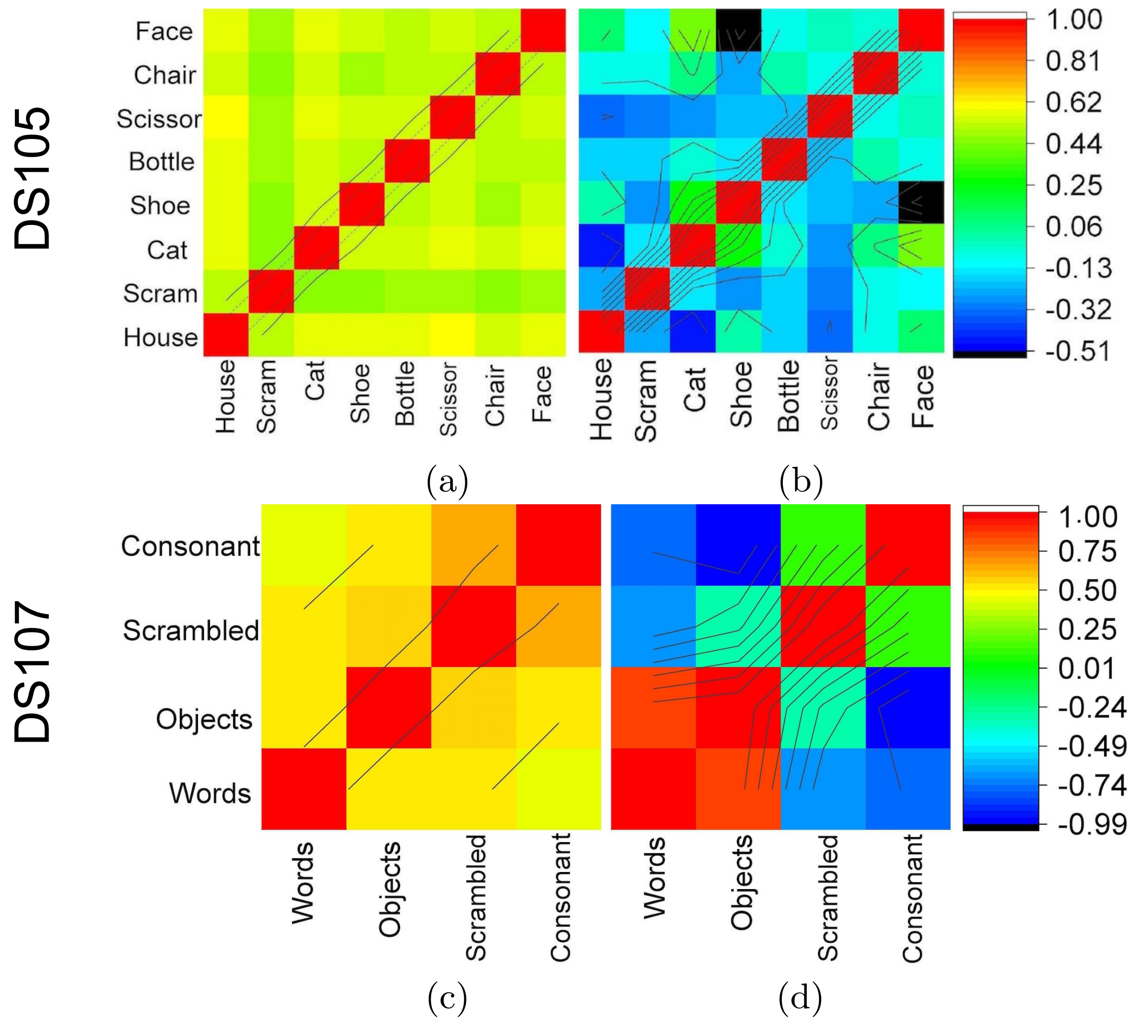
(c) Scramble



(d) Registration

- Woods function (W)
- Correlation Ratio (CR)
- Joint Entropy (JE)
- Mutual Information (MI)
- Normalized MI (NMI)

Correlation Analysis



Binary Classification (OVA)

Table 1: Accuracy of binary predictors

Data Sets	Cox & Savoy	McMenamin et al.	Mohr el al.	Osher et al.	Binary-APA
DS105-Objects	71.65±0.97	83.06±0.36	85.29±0.49	90.82±1.23	**98.37±0.16**
DS107-Words	69.89±1.02	89.62±0.52	81.14±0.91	94.21±0.83	**97.67±0.12**
DS107-Consonants	67.84±0.82	87.82±0.37	79.69±0.69	95.54±0.99	**98.73±0.06**
DS107-Objects	65.32±1.67	84.22±0.44	75.32±0.41	**95.62±0.83**	95.06±0.11
DS107-Scramble	67.96±0.87	86.19±0.26	78.45±0.62	93.1±0.78	**96.71±0.18**

Table 2: Area Under the ROC Curve (AUC) of binary predictors

Data Sets	Cox & Savoy	McMenamin et al.	Mohr el al.	Osher et al.	Binary-APA
DS105-Objects	68.37±1.01	82.22±0.42	80.91±0.21	88.54±0.71	**96.25±0.92**
DS107-Words	67.76±0.91	86.35±0.39	78.23±0.57	93.61±0.62	**97.02±0.2**
DS107-Consonants	63.84±1.45	85.63±0.61	77.41±0.92	94.54±0.31	**96.92±0.14**
DS107-Objects	63.17±0.59	81.54±0.92	73.92±0.28	94.23±0.94	**95.17±0.03**
DS107-Scramble	66.73±0.92	85.79±0.42	76.14±0.47	92.23±0.38	**96.08±0.1**

Multi-class Classification

Table 3: Accuracy of multi-class predictors

Data Sets	Cox & Savoy	McMenamin et al.	Mohr el al.	Osher et al.	Multi-APA
DS105 (P=8)	18.03±4.07	38.34±3.21	29.14±2.25	50.61±4.83	**57.93±2.1**
DS107 (P=4)	38.01±2.56	71.55±2.79	64.71±3.14	89.69±2.32	**94.21±2.41**
ALL (P=4)	32.93±2.29	68.35±3.07	63.16±4	80.36±3.04	**95.67±1.25**

Outline

Conclusion

- This paper proposes Anatomical Pattern Analysis (APA) framework for decoding visual stimuli in the human brain.
  - ✓ A new feature extraction method.
  - ✓ A customized classification algorithm.

- In future, we plan to apply the proposed method to different brain tasks such as low-level visual stimuli, emotion and etc.

- M. Yousefnezhad, D. Zhang, Local Discriminant Hyperalignment for multi-subject fMRI data alignment. 34th AAAI Conference on Artificial Intelligence (AAAI-17), San Francisco, California, USA, February/4-9, 2017.

Thanks for your attention!

For more details, contact:

myousefnezhad@nuaa.edu.cn

myousefnezhad@outlook.com

https://myousefnezhad.github.io