Machine Learning

CSE 6363 (Fall 2019)

Lecture 5 Dimension Reduction

Dajiang Zhu, Ph.D.

Department of Computer Science and Engineering

Slides of this course (CSE6363) courtesy: Dr. Heng Huang, Dr. Aarti Singh

High-Dimensional data

- High-Dimensions = Lot of Features

Document classification
Features per document =
thousands of words/unigrams
millions of bigrams, contextual
information

- Surveys –Netflix

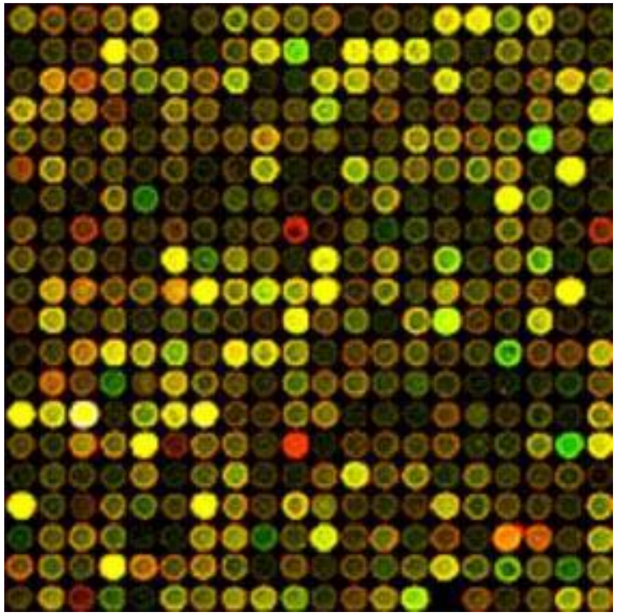
480189 users x 17770 movies

High-Dimensional data

High-Dimensions = Lot of Features

- Discovering gene networks

10,000 genes x 1000 drugs x several species



High-Dimensional data

High-Dimensions = Lot of Features

Functional resonance imaging (fMRI) data

x=90	y=104	z=72		673,920
x=90	y=104	z=72	t=284	**191,393,280**

We have **N** participants!

High-Dimensional data

Why are more features bad?

- Redundant features (not all words are useful to classify a document)
- more noise added than signal
- Hard to interpret and visualize
- Hard to store and process data (computationally challenging)
- Complexity of decision rule tends to grow with # features.

High-Dimensional data

Overall Strategies

- Feature Selection–Only a few features are relevant to the learning task (same space)

- Latent features–Some linear/nonlinear combination of features provides a more efficient representation than observed features (different space)

High-Dimensional data

Overall Strategies

- **Feature Selection–Only a few features are relevant to the learning task (same space)**

- Latent features–Some linear/nonlinear combination of features provides a more efficient representation than observed features (different space)

Feature Selection

- **Score each feature and extract a subset**
  - Training or cross-validated accuracy of single-feature classifiers f_i: $X_i \rightarrow Y$
  - Estimated mutual information between X_i and Y:

$$\hat{I}(X_i, Y) = \sum_k \sum_y \hat{P}(X_i = k, Y = y) \log \frac{\hat{P}(X_i = k, Y = y)}{\hat{P}(X_i = k)\hat{P}(Y = y)}$$

  - χ^2 statistic to measure independence between X_i and Y
  - Doman specific criteria

    -Text: Score some words such as “the”, “of”, … as zero

    -fMRI: Score some regions with higher weight (pre-knowledge)

Feature Selection

- **Score each feature and extract a subset**
  - Simple: select k highest scoring features
  - Iterative:
    - Choose single highest scoring feature X_k
    - Rescore all features, conditioned on the set of already-selected features
      - E.g., $Score(X_i \mid X_k) = I(X_i, Y \mid X_k)$
      - E.g, $Score(X_i \mid X_k) = Accuracy(\text{predicting } Y \text{ from } X_i \text{ and } X_k)$
    - Repeat, calculating new scores on each iteration, conditioning on set of selected features

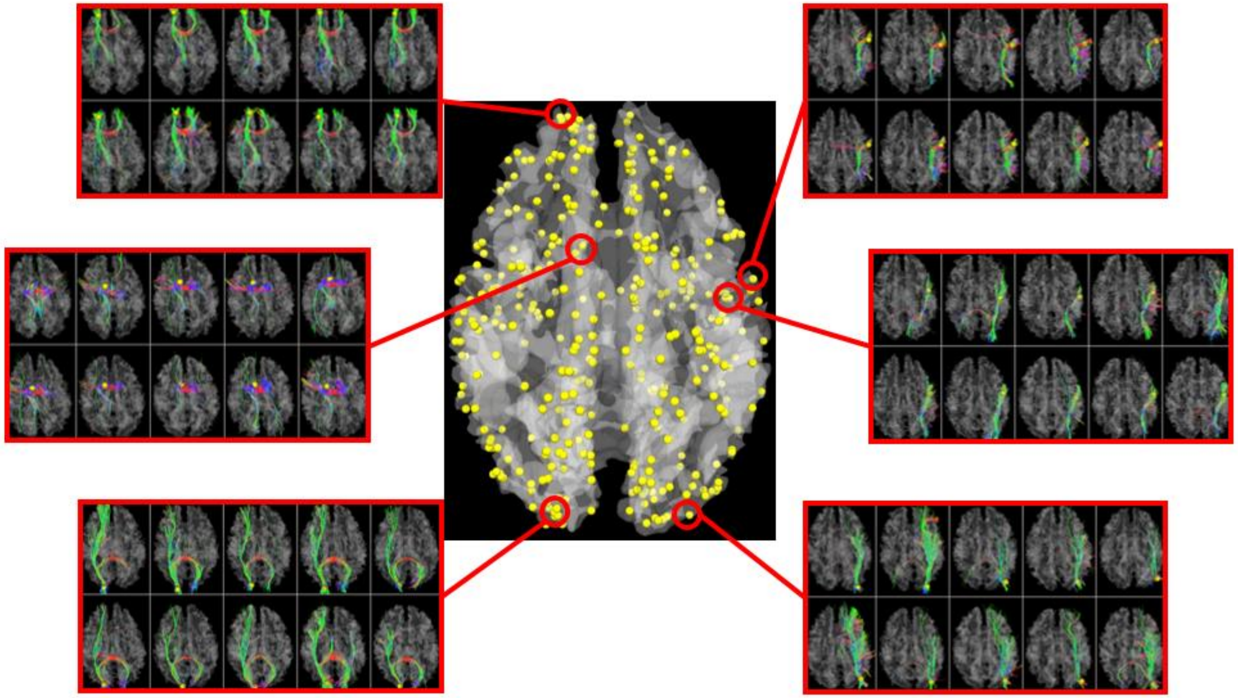
Feature Selection

- **An example**
  - t-Test
    - ✓Using two sample t-Test
    - ✓Set threshold, e.g. $p = 0.05$
  - CFS (Correlation-based Feature Selection)
    - ✓hypothesis: - A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

Feature Selection

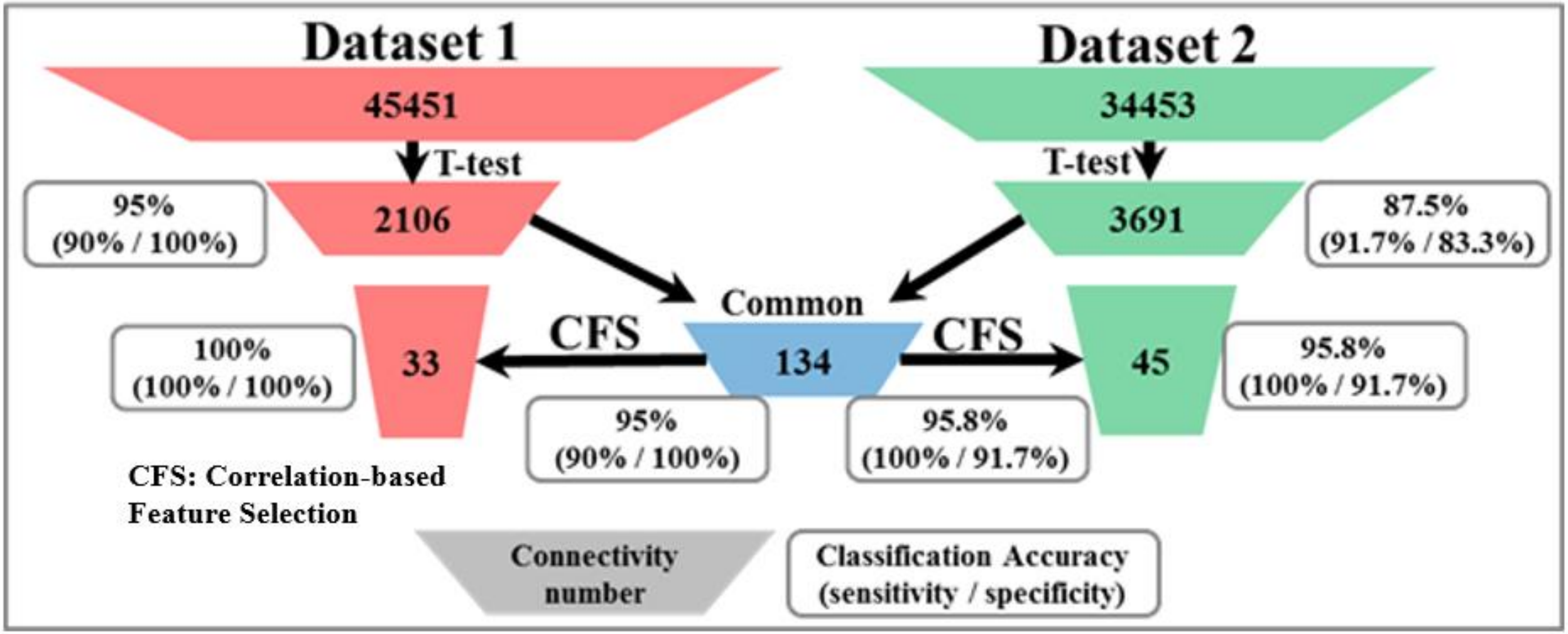
- **Score each feature and extract a subset**
  - ➢ Example 1



[*Cerebral Cortex*, 2012]

Feature Selection

- **Score each feature and extract a subset**
  - ➢ Example 1



[*Human Brain Mapping*, 2013]

Feature Selection

- **Score each feature and extract a subset**
  - **Regularization method**

Integrate feature selection into learning objective by penalizing number of features with non-zero weights

$$\widehat{W} = \arg\min_{W} \sum_{i=1}^{n} -\log P(Y_i | X_i; W) + \lambda \|W\|$$

Linear Regression Summary

$$\min_{\beta}(\mathbf{A}\beta - \mathbf{Y})^T(\mathbf{A}\beta - \mathbf{Y}) + \lambda \text{pen}(\beta) = \min_{\beta} J(\beta) + \lambda \text{pen}(\beta)$$

Least Square Solution

$$\text{pen}(\beta) = \|\beta\|_2^2$$

Ridge Regression

$$\text{pen}(\beta) = \|\beta\|_1$$

Lasso Regression

Lasso (L1 penalty) results in sparse solutions –vector with more zero coordinates. Will come to Lasso later!

Lasso Regression as feature selection

Tibshirani (*Journal of the Royal Statistical Society* 1996) introduced the **LASSO**: *least absolute shrinkage and selection operator*

J. R. Statist. Soc. B (1996)
58, *No.* 1, *pp.* 267–288

Regression Shrinkage and Selection via the Lasso

By ROBERT TIBSHIRANI†

University of Toronto, Canada

[Received January 1994. Revised January 1995]

SUMMARY

We propose a new method for estimation in linear models. The 'lasso' minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. Because of the nature of this constraint it tends to produce some coefficients that are exactly 0 and hence gives interpretable models. Our simulation studies suggest that the lasso enjoys some of the favourable properties of both subset selection and ridge regression. It produces interpretable models like subset selection and exhibits the stability of ridge regression. There is also an interesting relationship with recent work in adaptive function estimation by Donoho and Johnstone. The lasso idea is quite general and can be applied in a variety of statistical models: extensions to generalized regression models and tree-based models are briefly described.

Keywords: QUADRATIC PROGRAMMING; REGRESSION; SHRINKAGE; SUBSET SELECTION

Lasso Regression as feature selection

- LASSO coefficients are the solutions to the ℓ_1 optimization problem:

$$\text{minimize } (\mathbf{y} - \mathbf{Z}\beta)^\top(\mathbf{y} - \mathbf{Z}\beta) \text{ s.t. } \sum_{j=1}^{p} |\beta_j| \leq t$$

- This is equivalent to loss function:

$$\begin{aligned} PRSS(\beta)_{\ell_1} &= \sum_{i=1}^{n}(y_i - \mathbf{z}_i^\top\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j| \\ &= (\mathbf{y} - \mathbf{Z}\beta)^\top(\mathbf{y} - \mathbf{Z}\beta) + \lambda||\beta||_1 \end{aligned}$$

Lasso Regression as feature selection

- Again, we have a tuning parameter λ that controls the amount of regularization
- One-to-one correspondence with the threshhold t: recall the constraint:

$$\sum_{j=1}^{p} |\beta_j| \leq t$$

- Hence, have a "path" of solutions indexed by t
- If $t_0 = \sum_{j=1}^{p} |\hat{\beta}_j^{\text{ls}}|$ (equivalently, $\lambda = 0$), we obtain no shrinkage (and hence obtain the LS solutions as our solution)
- Often, the path of solutions is indexed by a fraction of shrinkage factor of t_0

Lasso Regression as feature selection

Why Lasso?

- In most cases, we believe that many coeficients (weights) shoulbe be 0.
- We seek a set of sparse solutions
- And large enough λ or small enough t will set some coefficients exactly equal to 0!

❖ **So, Lasso will perform model selection/feature selection/dimension reduction for us!**

Lasso Regression as feature selection

Computation of Lasso

- Unlike ridge regression, $\hat{\beta}_\lambda^{\text{lasso}}$ has no closed form
- Original implementation involves quadratic programming techniques from convex optimization
- But Efron et al. (*Annals of Statistics* 2004) proposed LARS (**least angle regression**), which computes the LASSO path efficiently

Lasso Regression as feature selection

Comparing Ridge and Lasso

- Even though $Z^T Z$ may not be of full rank, both ridge regression and Lasso admit solutions
- We always have issue that p $\gg$ n which means much more predictor variables than observations
  - ✓ Both ridge regression and Lasso have solutions
  - ✓ Regularization tends to reduce prediction error
- The ridge and Lasso solutions are both indexed by the continuous parameter λ

Lasso Regression as feature selection

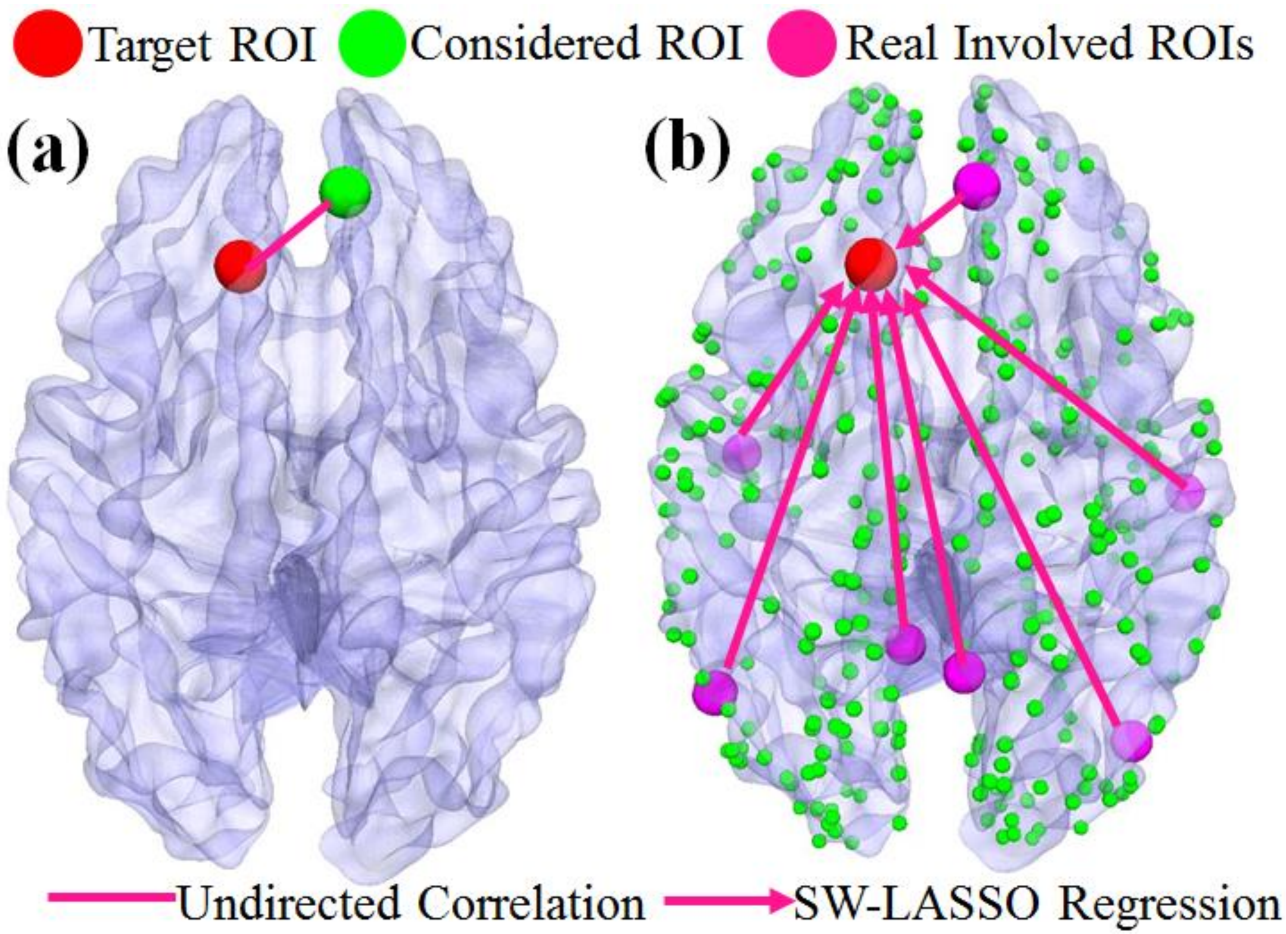
Suppose $p \gg n$

- Ridge regression produces coefficient values for each of the p variables
- Because of L1 penalty, Lasso will set many of the variables exactly equal to 0

=> Lasso produce **sparse solutions** and takes care of **model selection/feature selection/dimension reduction** for us!

Lasso Application - 1

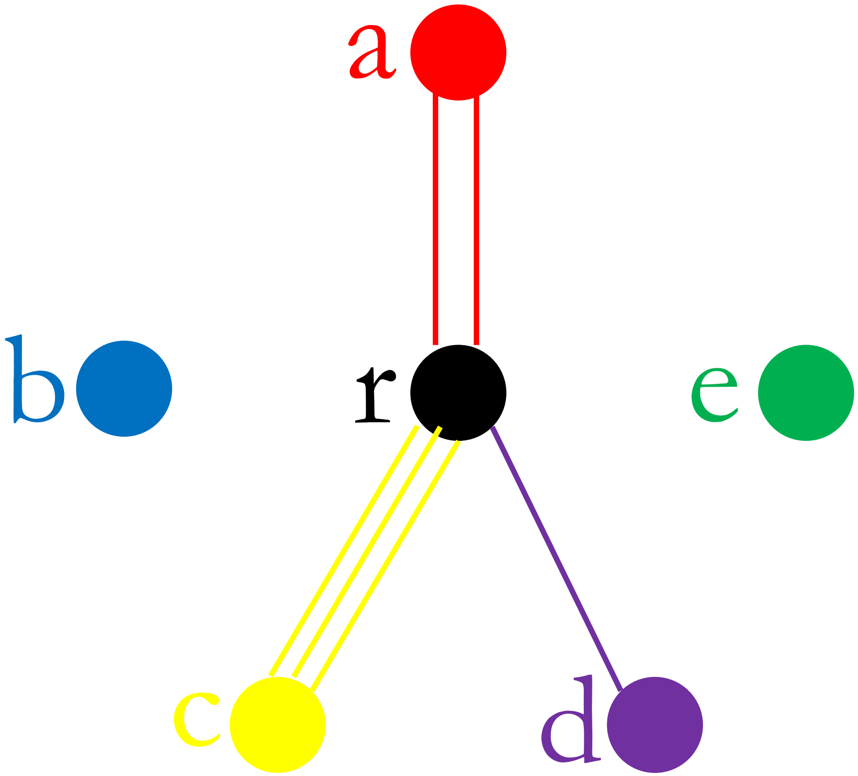
Simple idea of Structural Weighted-LASSO



[*IPMI*' 13]

Lasso Application - 1

Using structural regulation as penalty



$$f_{r,a} = \frac{||}{||+|||+|}$$

$$w_{r,a} = 1 - \frac{f_{r,a}}{p}$$

Lasso Application - 1

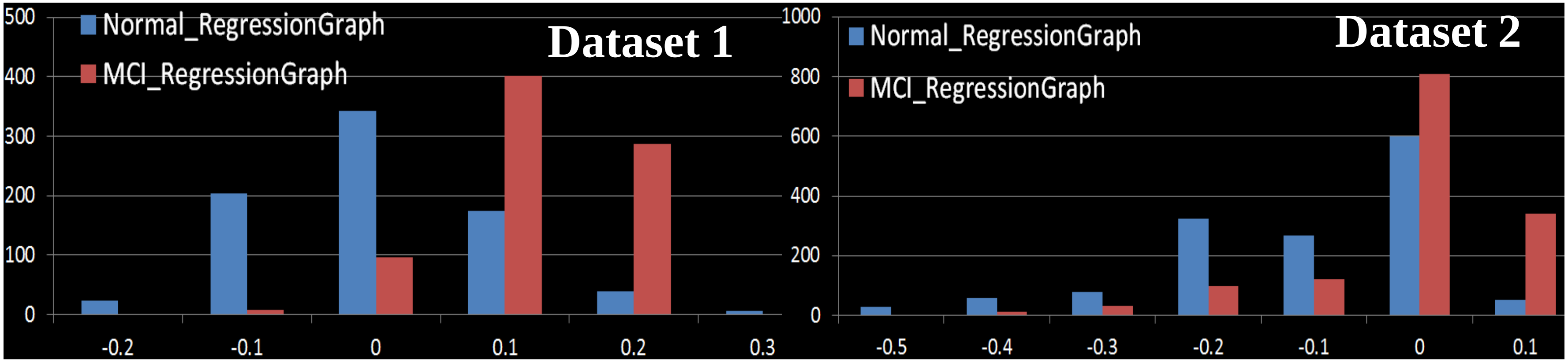
$$\hat{\beta}_{Lasso} = \arg\min\{\sum_{j=1}^{N}(y_r(j) - \sum_{i=1}^{k}\beta_i x_i(j))^2\} + \lambda\sum_{i=1}^{k}\boxed{(1 - f_{r,i}/p)}|\beta_i|$$

- y_r : Target. (one brain region)
- x_r :$\{x_1, x_2, \ldots, x_k\}$, k=357 (the other brain regions)

Lasso Application - 1

Assortative mixing

- From -1 (disassortative) to 1 (assortative)
- Reflect the preference of connecting to other nodes that either have similar (assortative) or dissimilar (disassortative) degree.



Lasso Application - 2

- Major Depressive Disorder (MDD)
  - affecting 350 million people globally

- Early diagnosis of MDD is challenging
  - ➢ Based on behavioral criteria
  - ➢ Rule out many factors (diabetes etc.)

- Study MDD from a more objective perspective
  - Magnetic Resonance Imaging (MRI)

Lasso Application - 2

MDD classification and potential bio-marker?

- Highly significant ≠ Highly predictive
- Small sample size, large number of features
- Single data source

Current MDD classification:

- ➢ Accuracy: 67% ~ 90% (only using T1 brain measures)
- ➢ Samples: most studies have 20 ~ 40 participants
- ➢ Data source: most are using data from single site

Lasso Application - 2

Overall strategy

- Machine learning
  - ➢ Feature selection + Classification

- Distributed- LASSO
  - ➢ A collaborative platform for multi-site data analysis
  - ➢ Each site DOES NOT need to share their raw data
  - ➢ Based on ENIGMA MDD Working Group

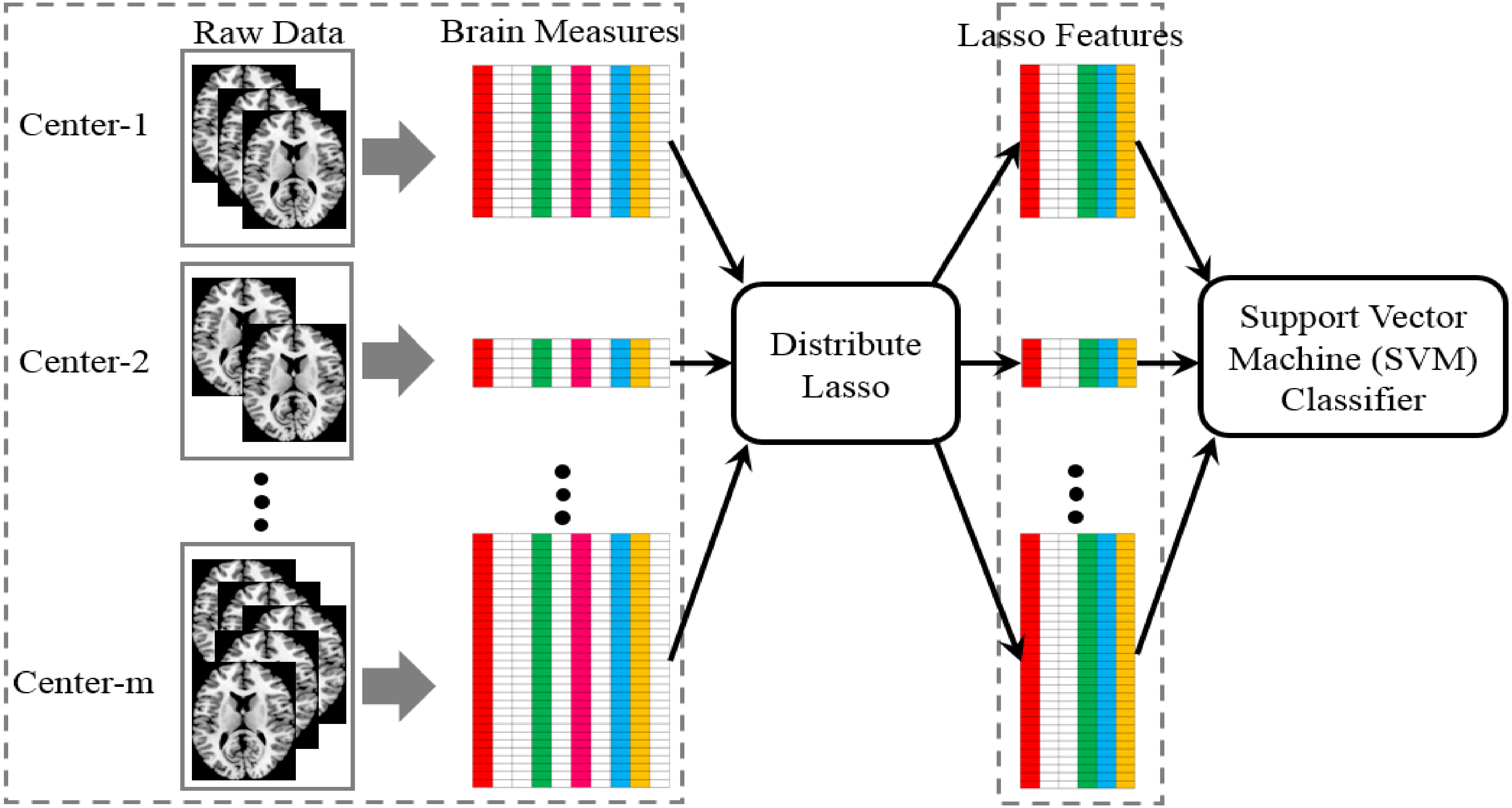
Lasso Application - 2

Healthy controls: ***1826***
MDD patients: ***1411***
Total: 3237

Demographics Across Sites

Sites & Location		Cognitively Normal				Major Depression			
		Males		Females		Males		Females	
		N	MeanAge(SD)	*N*	MeanAge(SD)	*N*	MeanAge(SD)	*N*	MeanAge(SD)
Amsterdam	*Netherlands*	*0*	--	*0*	--	*0*	--	*51*	29.4 (4.7)
Berlin	*Germany*	*31*	40.6 (12.4)	39	40.4 (13.0)	*36*	41.7 (11.3)	*65*	41.5 (12.0)
BRCDECC	*England*	*29*	49.2 (7.3)	*32*	49 (8.6)	*22*	50.3 (10.0)	*47*	50.1 (8.3)
Calgary	*Canada*	*4*	21.8 (0.8)	*2*	21.6 (0.0)	*6*	22.9 (0.0)	*4*	22.7 (1.0)
CLING	*Germany*	*117*	27.5 (5.0)	*164*	27.3 (5.3)	*23*	28.6 (9.7)	*20*	28.4 (11.0)
Dublin	*USA*	*23*	39.8 (12.7)	*24*	39.6 (12.3)	*20*	40.9 (11.5)	*33*	40.7 (10.4)
Groningen	*Netherlands*	*6*	42.6 (13.9)	*17*	42.4 (14.1)	*6*	43.7 (12.7)	*16*	43.5 (14.4)
Houston	*USA*	*33*	40.1 (11.5)	*67*	40 (12.3)	*20*	41.2 (12.0)	*48*	41.1 (11.9)
Magdeburg	*Germany*	*17*	35.9 (6.1)	*3*	35.7 (9.6)	*11*	37 (10.1)	*8*	36.8 (13.1)
Melbourne	*Australia*	*16*	22.9 (1.2)	*17*	22.7 (1.0)	*8*	24 (1.1)	*9*	23.8 (0.8)
MPIP	*Germany*	*90*	48.7 (12.7)	*124*	48.5 (12.1)	*158*	49.8 (13.1)	*199*	49.6 (13.4)
Munster	*Germany*	*304*	37.1 (11.3)	*385*	36.9 (12.0)	*114*	38.2 (10.7)	*150*	38 (11.5)
NESDA	*Netherlands*	*23*	38.7 (9.7)	*41*	38.5 (9.5)	*50*	39.8 (9.3)	*91*	39.6 (9.3)
Pedro	*Brazil*	*35*	32.1 (5.7)	*36*	31.9 (8.0)	*6*	33.2 (9.8)	*14*	33 (7.8)
Stanford	*USA*	*21*	37.5 (9.6)	*35*	37.4 (10.3)	*23*	38.6 (9.6)	*31*	38.5 (10.1)
Sydney	*Australia*	*42*	49.3 (22.6)	*49*	49.1 (22.2)	*45*	50.4 (21.6)	*77*	50.2 (19.2)
TOTAL		*791*	38 (13.7)	*1035*	37.7 (13.9)	*548*	42.6 (13.6)	*863*	42.3 (13.5)

Lasso Application - 2



Li, Q., et al., **MICCAI**, 2016 Zhu, D., et al., **SIPAIM**, 2016

Lasso Application - 2

$$\min_x \frac{1}{2} \|Ax - y\|_2^2 + \lambda \|x\|_1 : x \in \mathbb{R}^p$$

A: Brain measure

y: Labels indicating MDD/Control

Lasso Application - 2

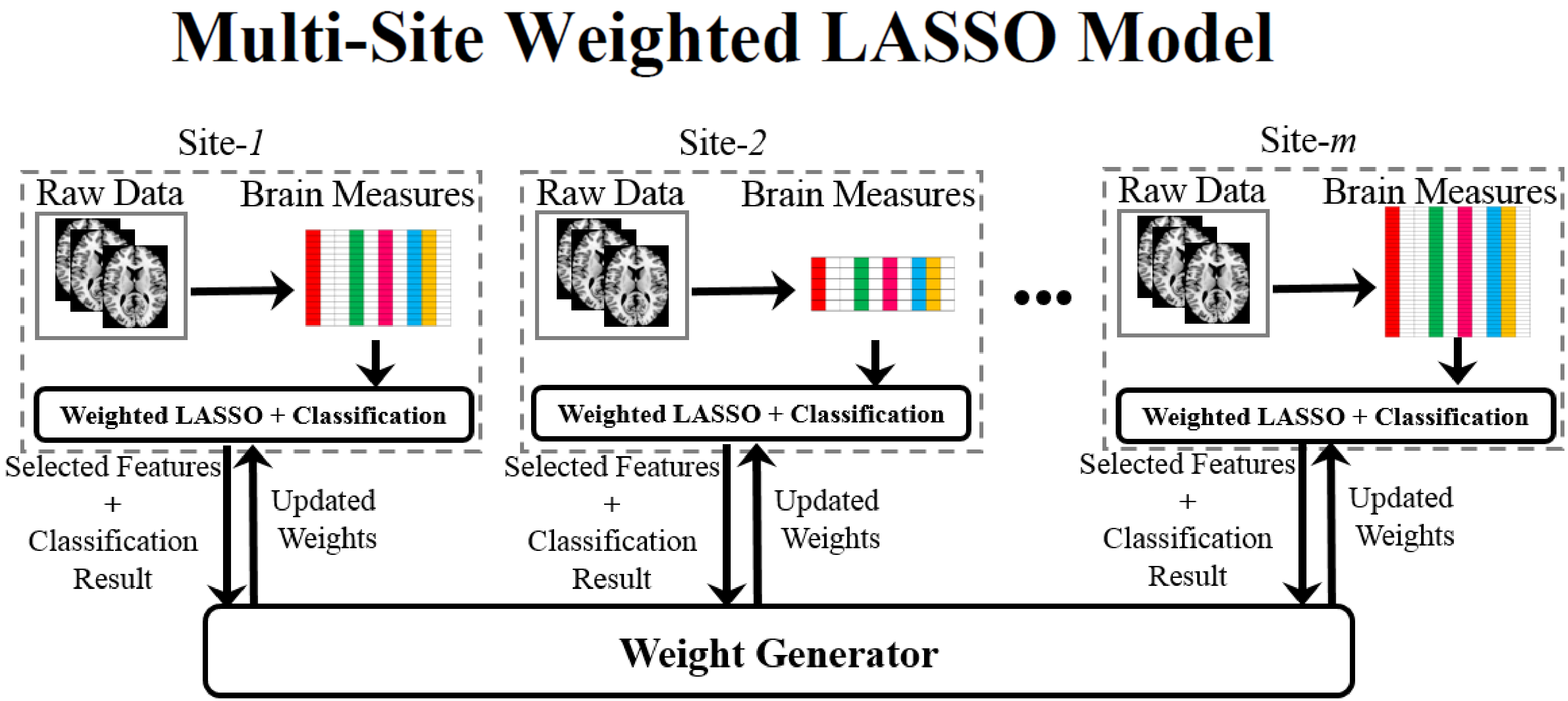
i^{th} **center:** (A_i, y_i) $\quad A_i \in \mathbb{R}^{n_i \times p} \quad y_i\ \mathbb{R}^{n_i \times 1}$

n_i is the number of participants at this center
p is the number of brain measures (all subjects are assumed to have the same number - p)

$$\begin{aligned}\nabla g &= A^T\ (\mathrm{Ax\text{-}y}) \\ &= \textstyle\sum_{i=1}^{m} A_i^T\ (A_i x - y_i) \\ &= \textstyle\sum_{i=1}^{m} \nabla g_i\end{aligned}$$

- The principle behind this formula is that it is possible to decompose the gradient computation on all the data into computing local gradients separately, which relate only to local data

Lasso Application - 3



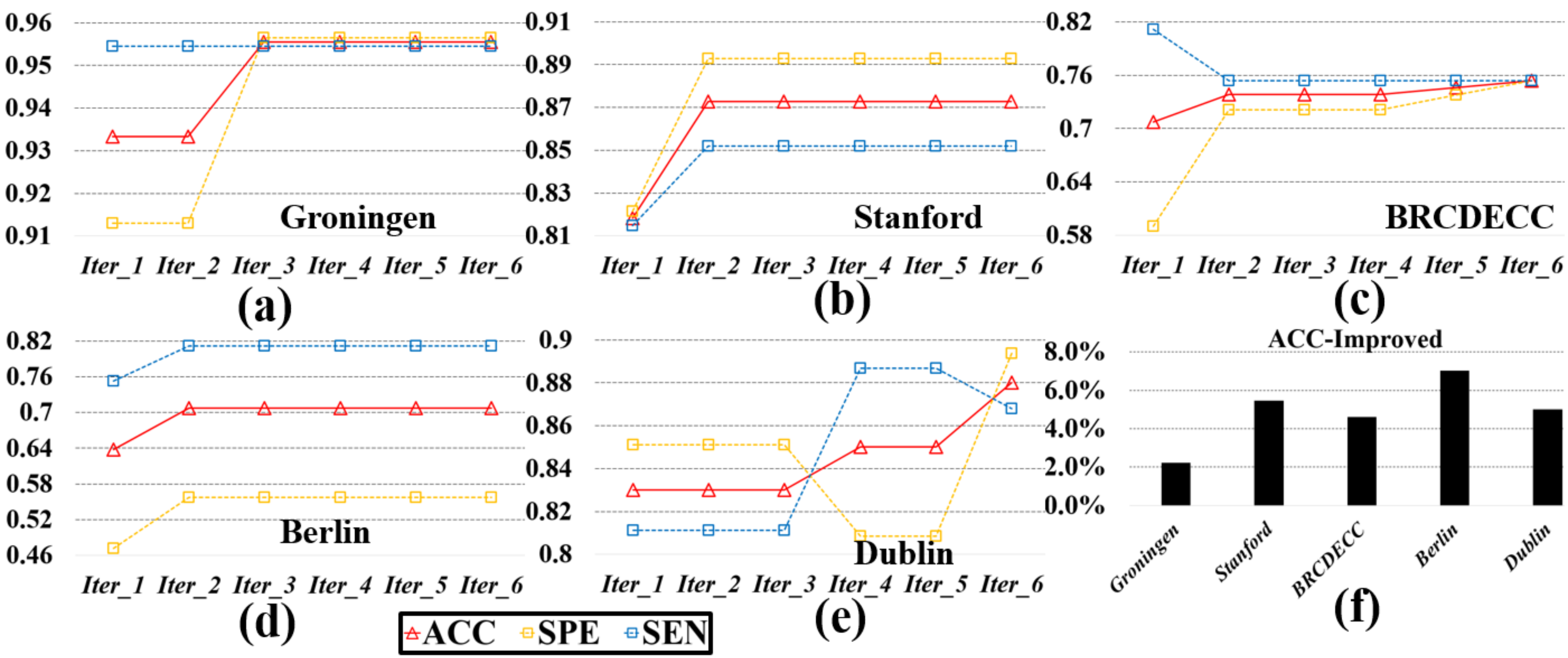
Lasso Application - 3

$$\hat{\beta}_{\mathrm{MSW-Lasso}} = \arg\min \|\mathrm{y} - \sum_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{x_i} \beta_{\mathrm{i}}\|^2 + \lambda \sum_{\mathrm{i}=1}^{\mathrm{n}} (1 - \sum_{s=1}^{m} \Psi_{s,i} A_s P_s / m) |\beta_{\mathrm{i}}|$$

$$W_f = \sum_{s=1}^{m} \Psi_{s,f} A_s P_s / m$$

$$\Psi_{s,f} = \begin{cases} 1, if \ \text{the } f^{th} \text{ feature was selected in site} - \mathrm{s} \\ 0, otherwise \end{cases}$$

Lasso Application - 3



High-Dimensional data

Overall Strategies

- Feature Selection–Only a few features are relevant to the learning task (same space)

- **Latent features–Some linear/nonlinear combination of features provides a more efficient representation than observed features (different space)**

Feature Selection

- **Latent Feature Extraction**
  - Combinations of observed features provide more efficient representation, and capture underlying relations that govern the data
- **Linear**
  - Principal Component Analysis (PCA)
  - Sparse Learning
  - Independent Component Analysis (ICA)
- **Nonlinear**
  - Laplacian Eigenmaps
  - ISOMAP

Sparse Learning

Whole brain fMRI signals

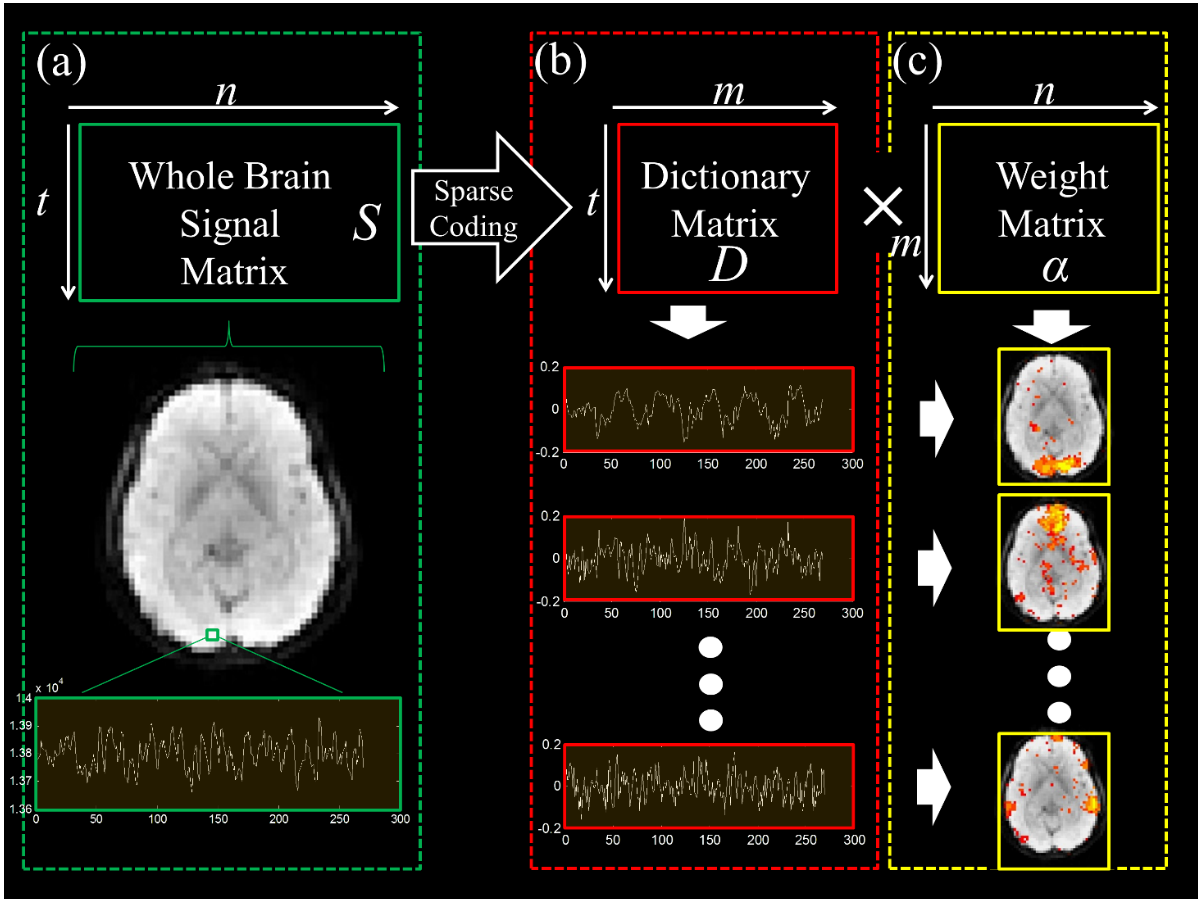
Coefficient

$$\ell(x, D) = \min_{\alpha \in \mathbb{R}^k} \frac{1}{2} ||x - D\alpha||_2^2 + \lambda ||\alpha||_1$$

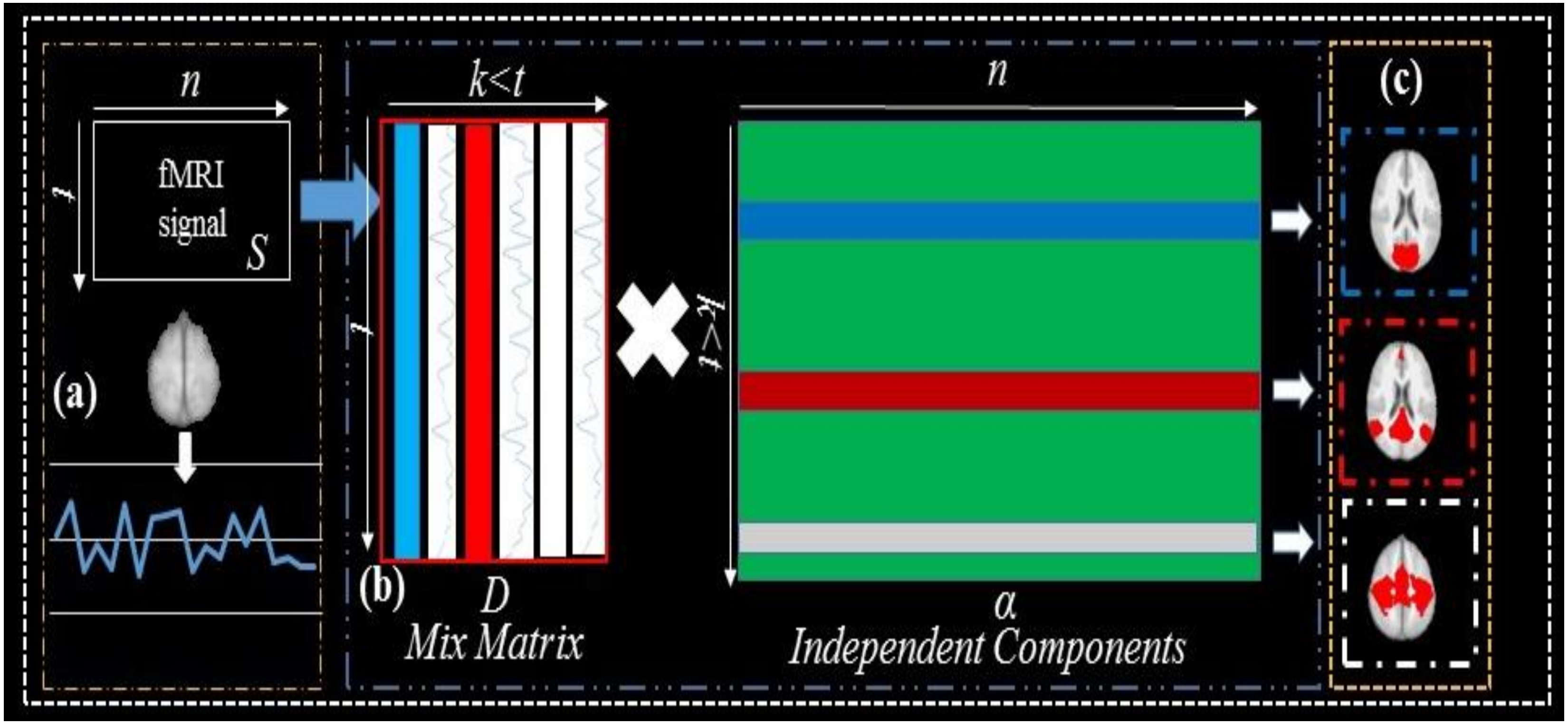
Dictionary

Regularization term

Sparse Learning



Independent Component Analysis



Comparing PCA, SL and ICA

PCA?

Sparse learning (dictionary learning)?

ICA?