Multi-Task Learning

Jiayu Zhou

Michigan State University

Road Map

- **Part I**: Multi-Task Learning (MTL) Background and motivation
- **Part II**: Overview of MTL Models
- **Part III**: Application of MTL on disease progression
- **Part IV**: MTL Software Package (MALSAR)

Multiple Tasks

- Examination Scores Prediction[1]

School 1 - Alverno High School

Student id	Birth year	Previous score	...	School ranking	...	Exam score
72981	1985	95	...	83%	...	?

student-dependent | school-dependent

School 138 - Jefferson Intermediate School

Student id	Birth year	Previous score	...	School ranking	...	Exam score
31256	1986	87	...	72%	...	?

student-dependent | school-dependent

School 139 - Rosemead High School

Student id	Birth year	Previous score	...	School ranking	...	Exam score
12381	1986	83	...	77%	...	?

student-dependent | school-dependent

[1]The Inner London Education Authority (ILEA)

Learning Multiple Tasks

- Learning each task independently

School 1 - Alverno High School

Student id	Birth year	Previous score	School ranking	...	Exam Score
72981	1985	95	83%	...	?

task 1st

Excellent

Excellent

task 138th

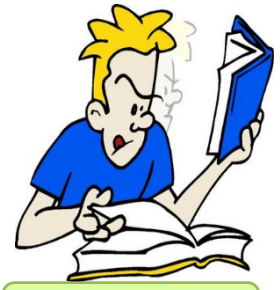
School 138 - Jefferson Intermediate School

Student id	Birth year	Previous score	School ranking	...	Exam Score
31256	1986	87	72%	...	?

School 139 - Rosemead High School

Student id	Birth year	Previous score	School ranking	...	Exam Score
12381	1986	83	77%	...	?

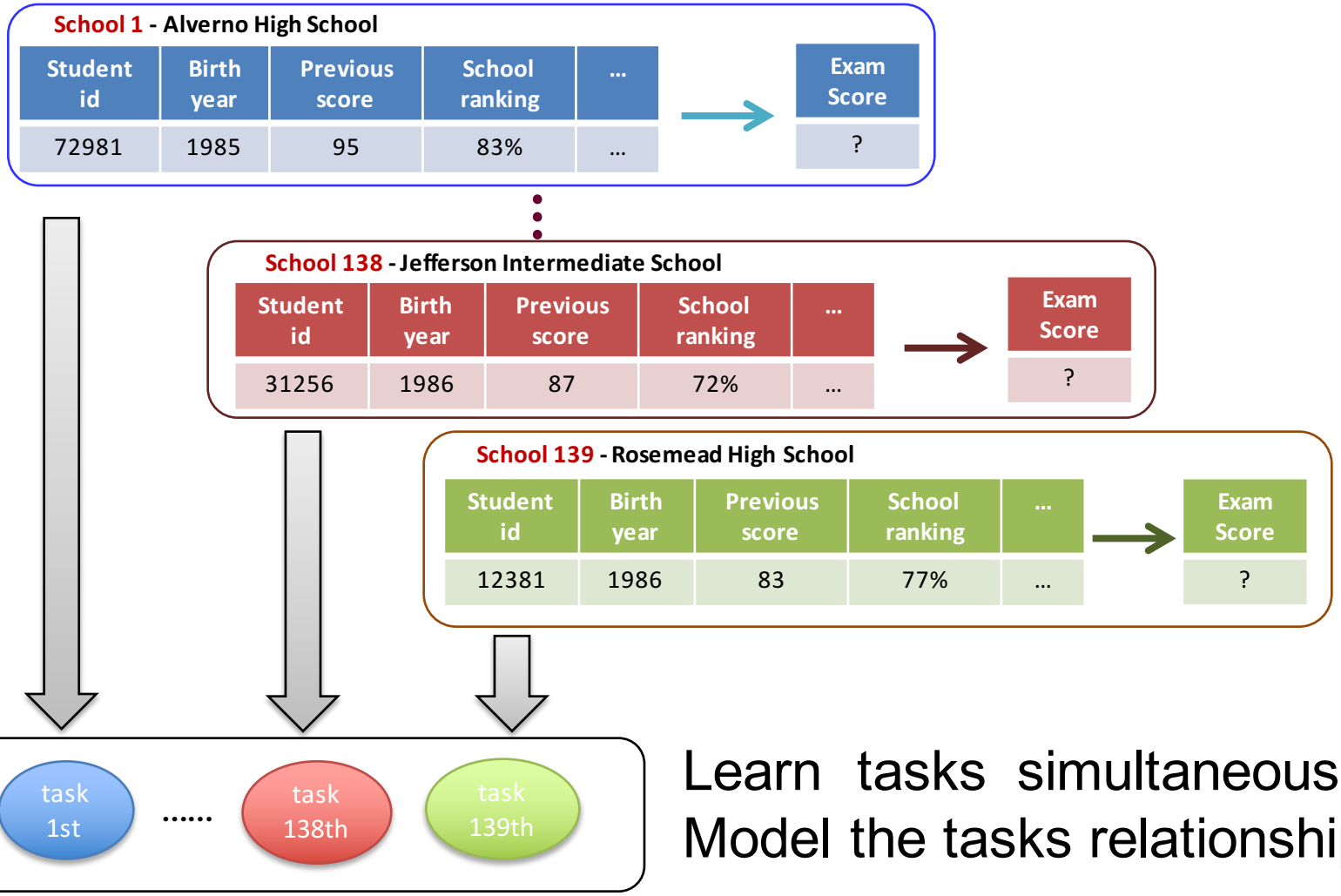
task 139th



Excellent

Learning Multiple Tasks

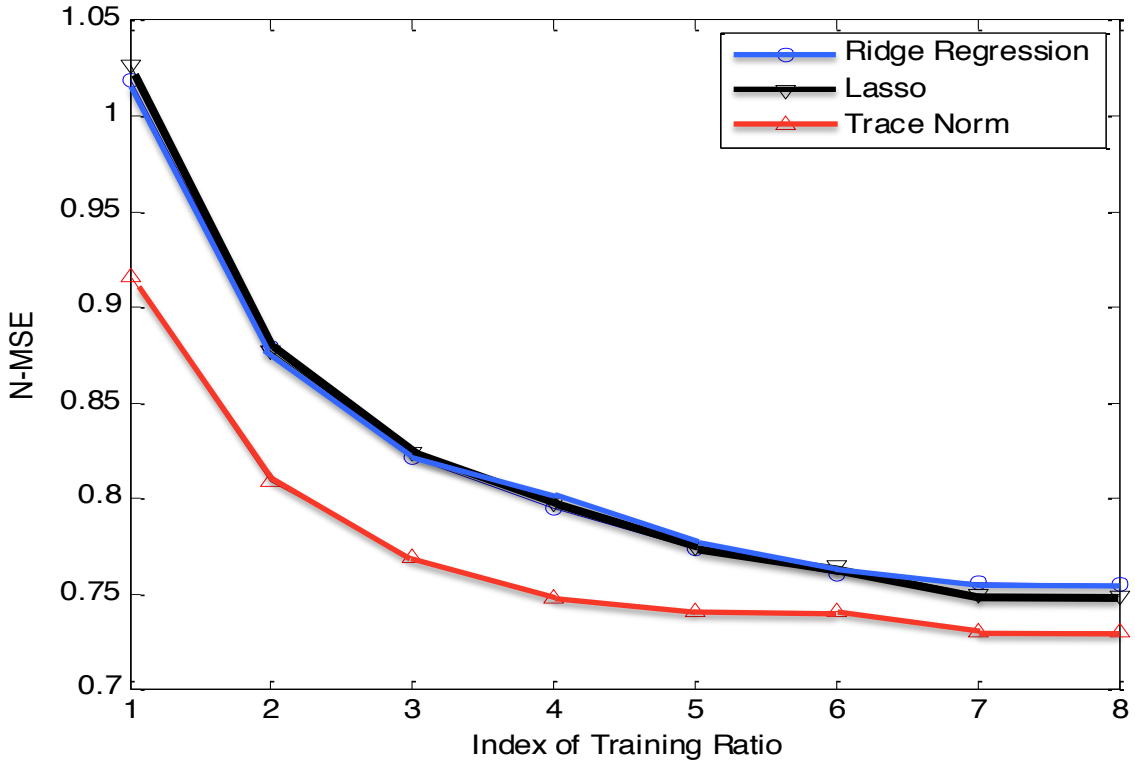
- Leaning multiple tasks simultaneously



Learn tasks simultaneously
Model the tasks relationship

Performance of MTL

- Evaluation on the *School* data:
  - Predict exam scores for 15362 students from 139 schools
  - Describe each student by 27 attributes
  - Multi-task learning performs significantly better than other single task learning approaches.



Performance measure:

$$\text{N-MSE} = \frac{\text{mean squared error}}{\text{variance (target)}}$$

More Applications of Multi-Task Learning

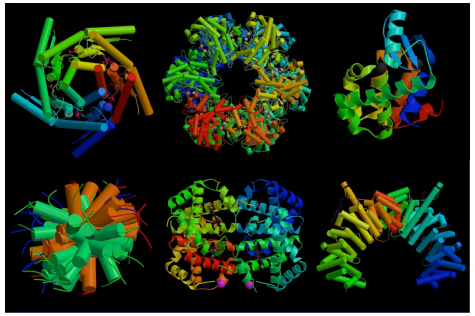
HIV Therapy Screening *[Bickel, ICML'08]*

Collaborative ordinal regression
[Yu et. al. NIPS'06]

Disease progression modeling
[Zhou et. al. KDD'11, 12]

Web image and video search
[Wang et. al. CVPR'09]

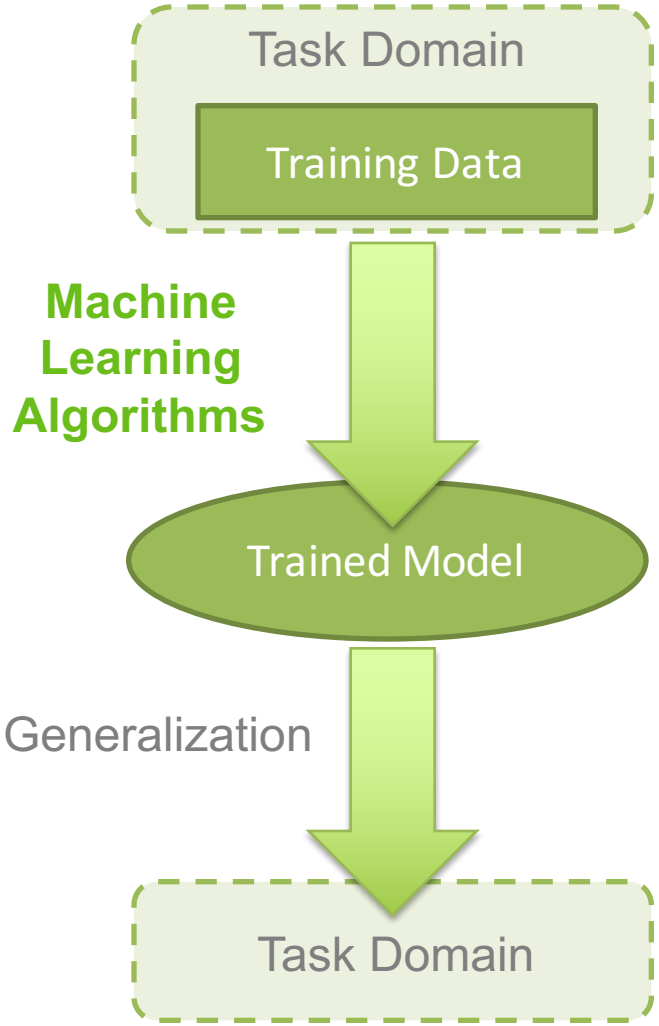
Disease prediction
[Zhang et. al. NeuroImage 12]



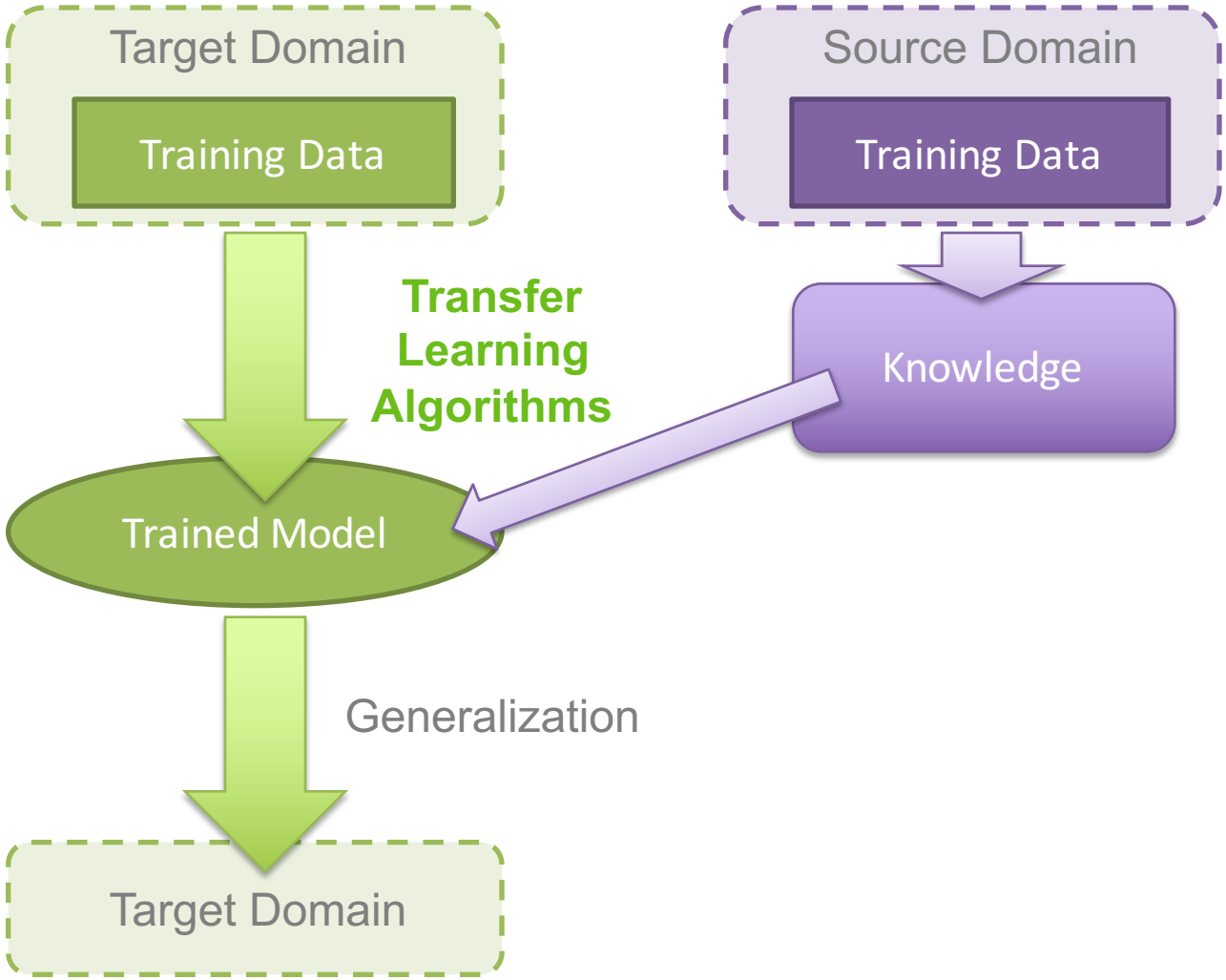
Protein classification
[Charuvaka et. al. ICDM'12]

Traditional Machine Learning

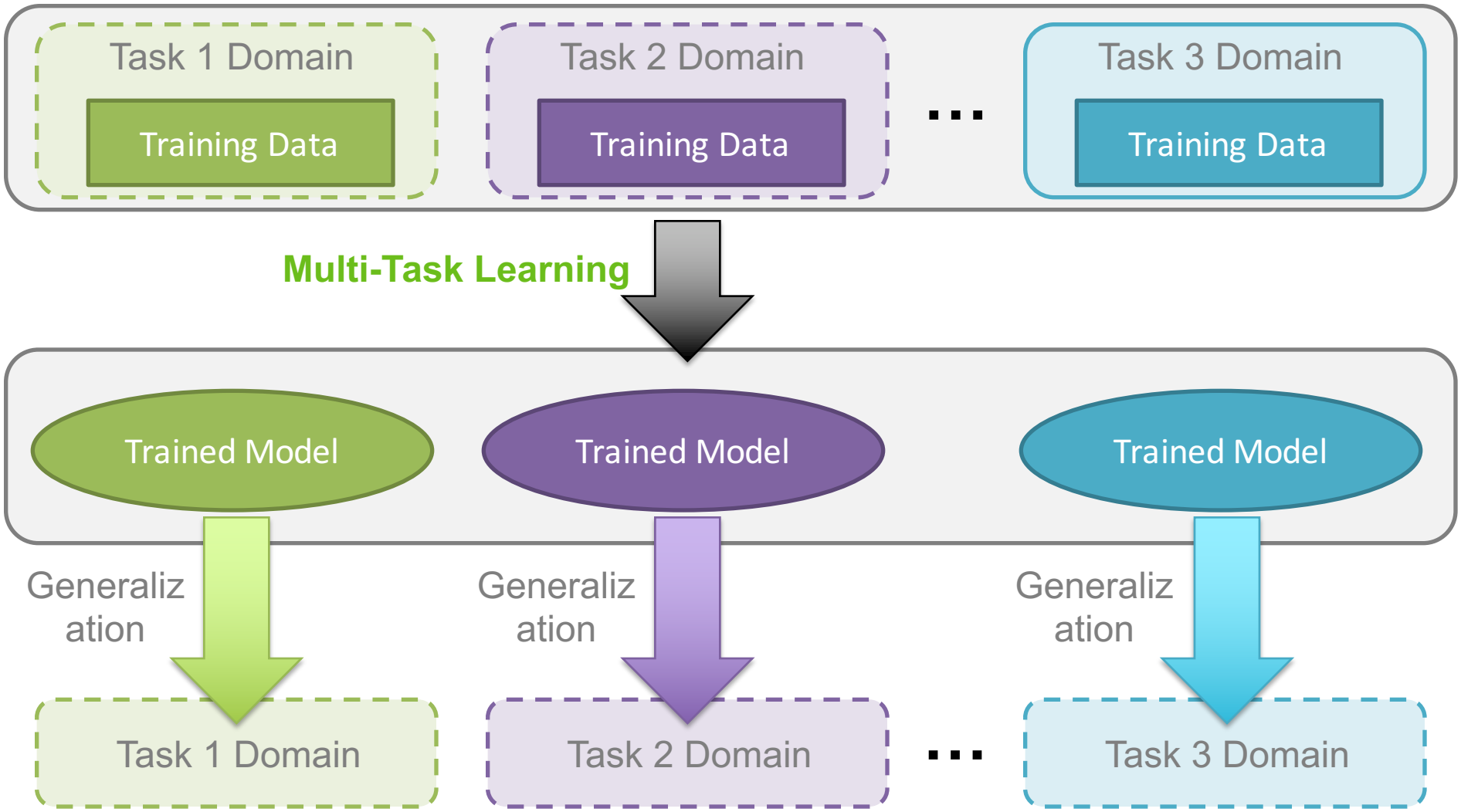
- Elements of machine learning on single task
  - The problem (task/domain)
  - Training data
  - Learning algorithms
  - Trained model
  - Applying model on unseen data (generalization)



Transfer Learning

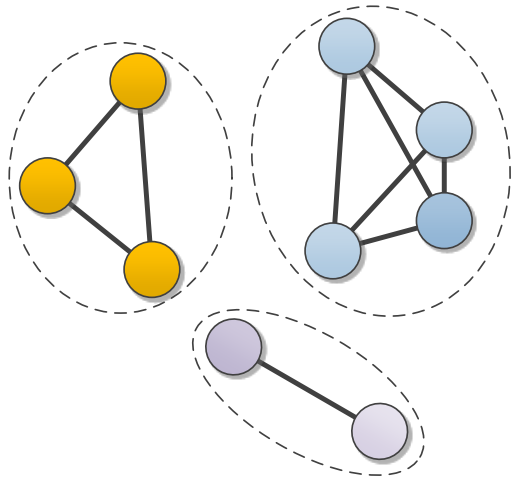


Multi-Task Learning



The Multi-Blah Family

- **Multi-Task Learning**
  - A set of related machine learning tasks
  - Different samples, (usually) same features for each task
- **Multi-View Learning**
  - A learning task involving a set of different views of the same set of objects (e.g., text and image descriptions)
  - Same samples, different features for each view
- **Multi-Label Learning**
  - A learning task where the prediction for each sample includes multiple labels (e.g., news categories)
  - Can be considered as multi-task with the same data matrices
- **Multi-Class Learning**
  - A classification task where the label can be multiple values (e.g., weather prediction)
  - Can be considered as multi-label with mutual exclusive labels.



Overview of MTL Models

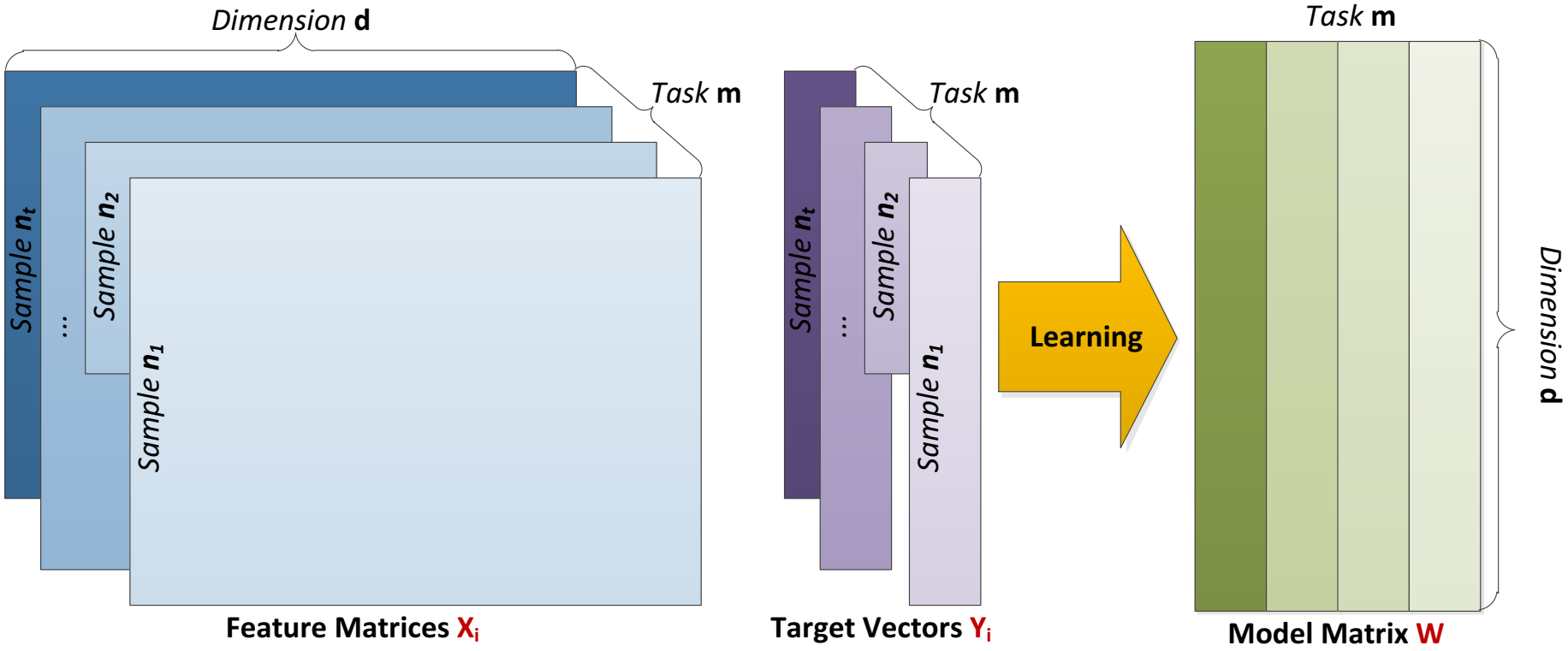
Achieve Multi-Task Learning

- Shared Hidden Nodes in Neural Network
- Shared Parameter Gaussian Process
- Multi-Task Regularization
  - Can be designed to incorporate various assumptions and domain knowledge
  - Can be trained using large-scale optimization algorithms on big data
  - The key is to design the regularization term that couples the tasks.

Representative Regularized MTL

- Mean-Regularized MTL
- MTL with High-Dimensional Features
  - Embedded Feature Selection
  - Low-Rank Subspace Learning
- Clustered MTL

Notation



- We focus on linear models:

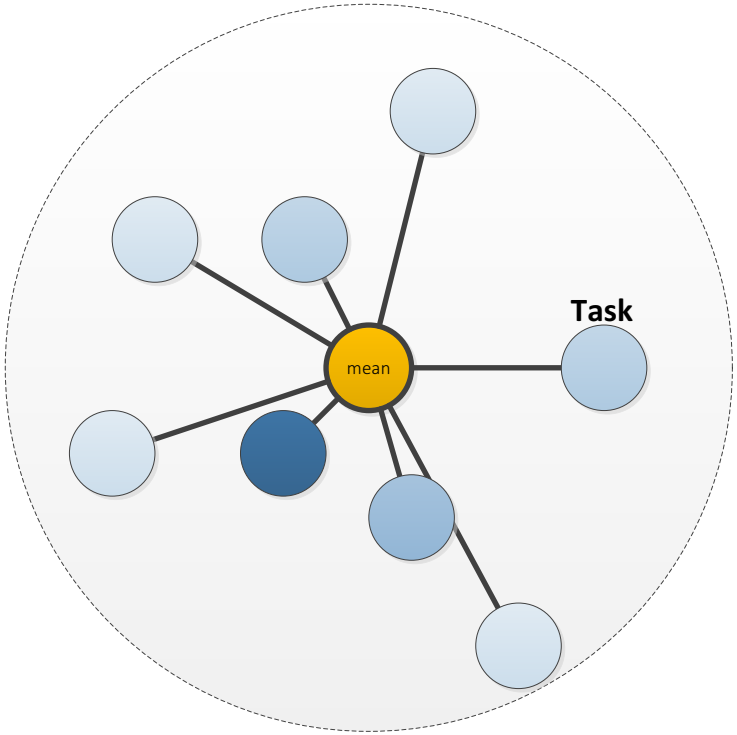
Mean-Regularized Multi-Task Learning

Evgeniou & Pontil, 2004 KDD

- Assumption: task parameter vectors of all tasks are close to each other.
  - Advantage: simple, intuitive, easy to implement
  - Disadvantage: may not hold in real applications.

Regularization
penalizes the deviation of each task from the mean

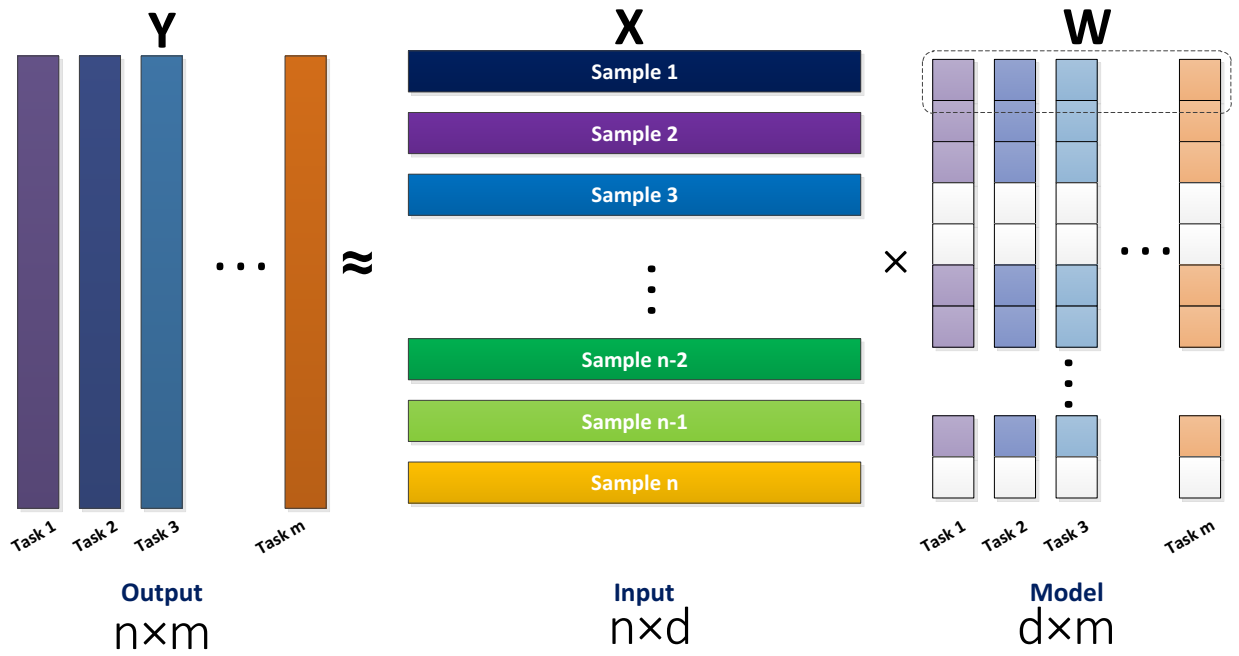
$$\min_W \frac{1}{2}\|XW - Y\|_F^2 + \lambda \sum_{i=1}^{m} \left\| W_i - \frac{1}{m}\sum_{s=1}^{m} W_s \right\|_2^2$$



Multi-Task Learning with Joint Feature Learning

Obozinski *et. al.* 2009 Stat Comput, Liu *et. al.* 2010 Technical Report

- Using group sparsity: ℓ_1/ℓ_q-norm regularization
- When q>1 we have group sparsity.



$$\min_W \frac{1}{2}\|XW - Y\|_F^2 + \lambda\|W\|_{1,q} \qquad \|W\|_{1,q} = \sum_{i=1}^{d}\|\boldsymbol{w}_i\|_q$$

Regularization

Encourages group sparsity

Writer-Specific Character Recognition

Obozinski, Taskar, and Jordan, 2006

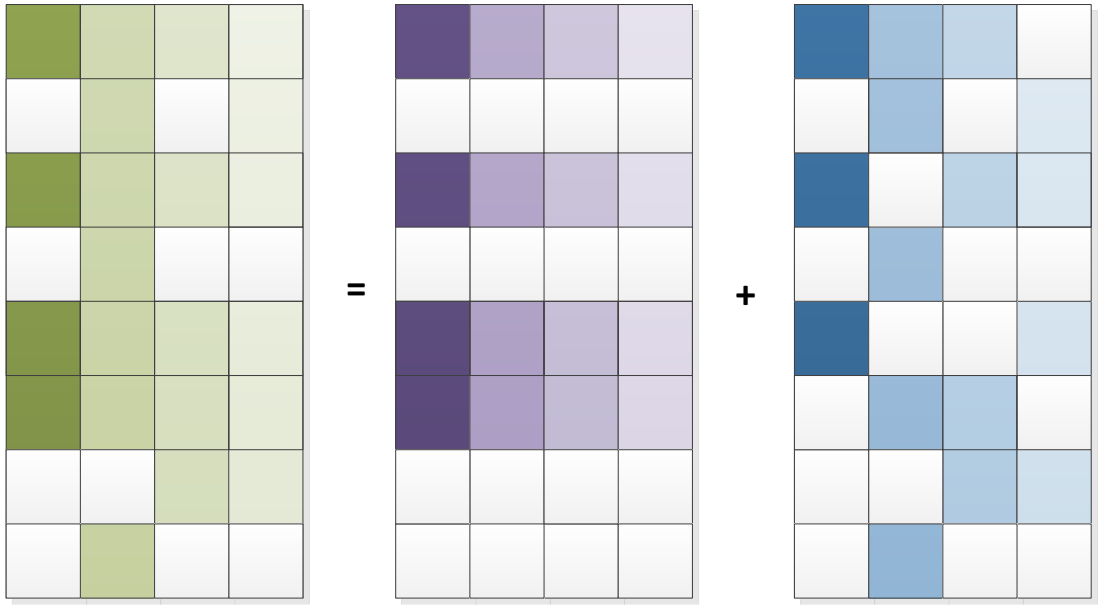
- Each task is a classification between two letters for one writer.

	pixels: error (%)			
Task	ℓ_1/ℓ_2	ℓ_1/ℓ_1	id.ℓ_1	pool
c/e	**4.0**	8.5	9.0	4.5
g/y	**11.4**	16.1	17.2	18.6
g/s	**4.4**	10.0	10.3	6.9
m/n	**2.5**	6.3	6.9	4.1
a/g	**1.3**	3.6	4.1	3.6
i/j	12.0	14.0	14.0	**11.3**
a/o	**2.8**	4.8	5.2	4.2
f/t	**5.0**	6.7	6.1	8.2
h/n	**3.2**	14.3	18.6	5.0

Dirty Model for Multi-Task Learning

Jalali *et. al.* 2010 NIPS

- In practical applications, it is too restrictive to constrain all tasks to share a single shared structure.



Model
W

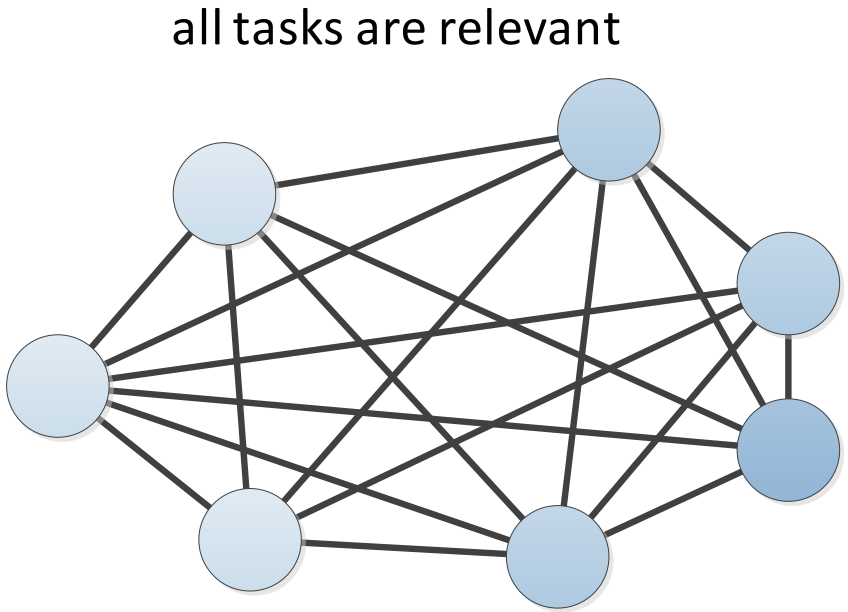
Group Sparse Component
P

Sparse Component
Q

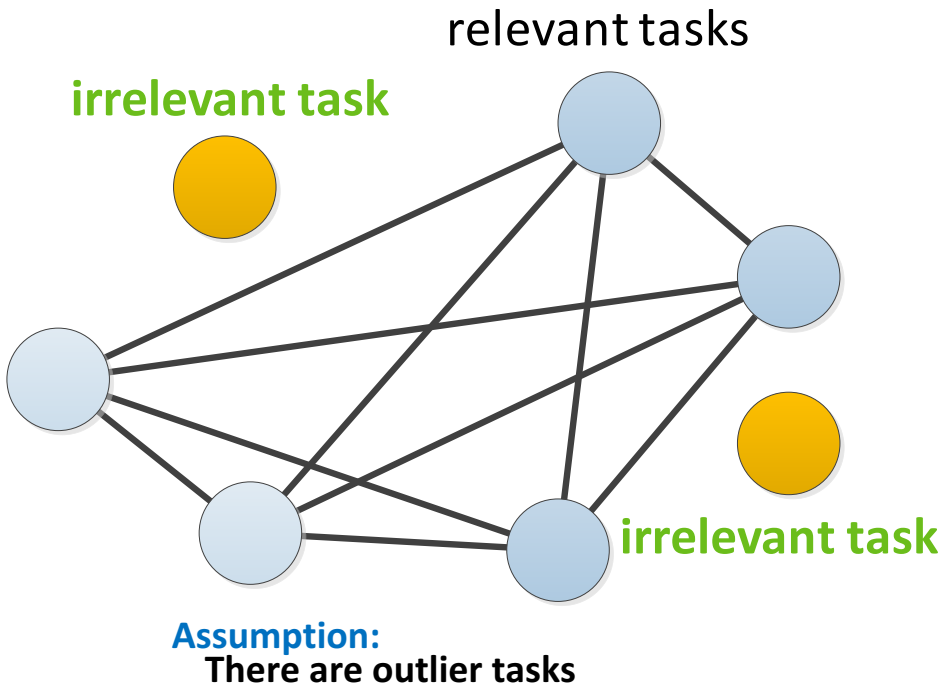
$$\min_{P,Q} \|Y - X(P+Q)\|_F^2 + \lambda_1 \|P\|_{1,q} + \lambda_2 \|Q\|_1$$

Robust Multi-Task Learning

- Most Existing MTL Approaches



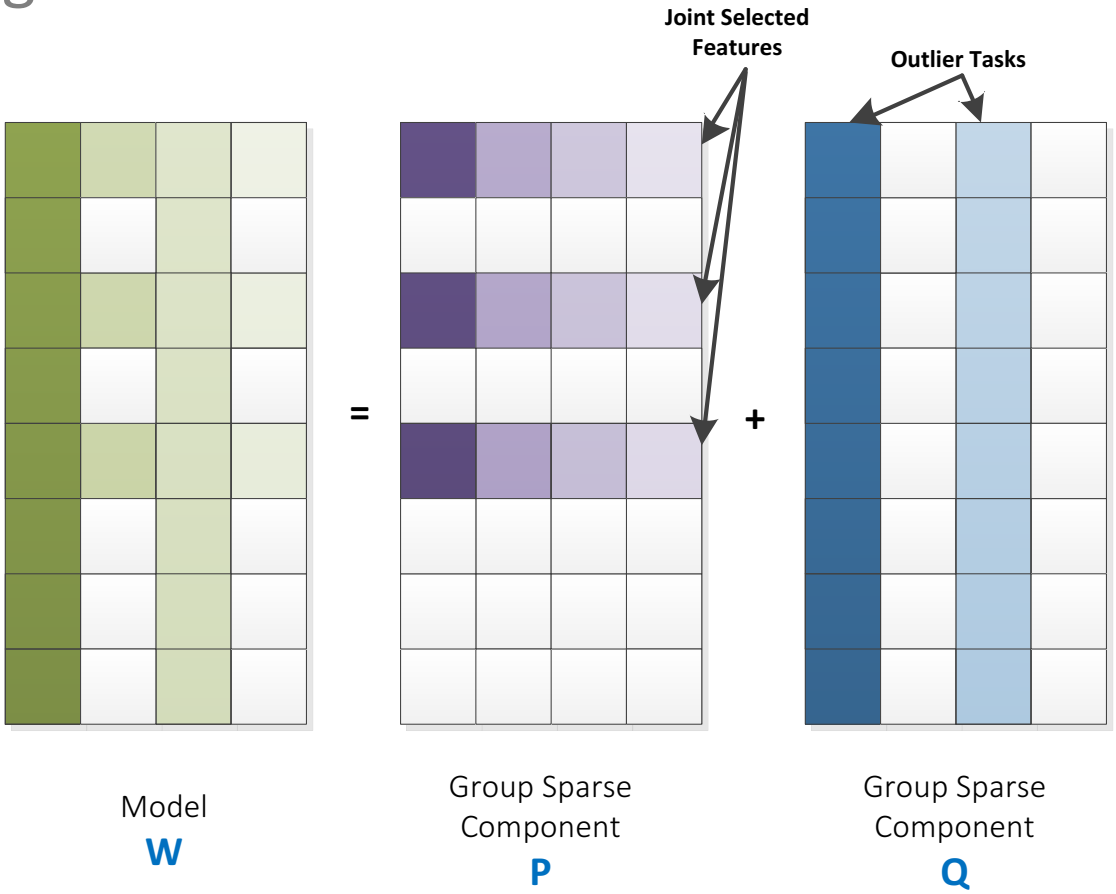
- Robust MTL Approaches



Robust Multi-Task Feature Learning

Gong *et. al.* 2012 KDD

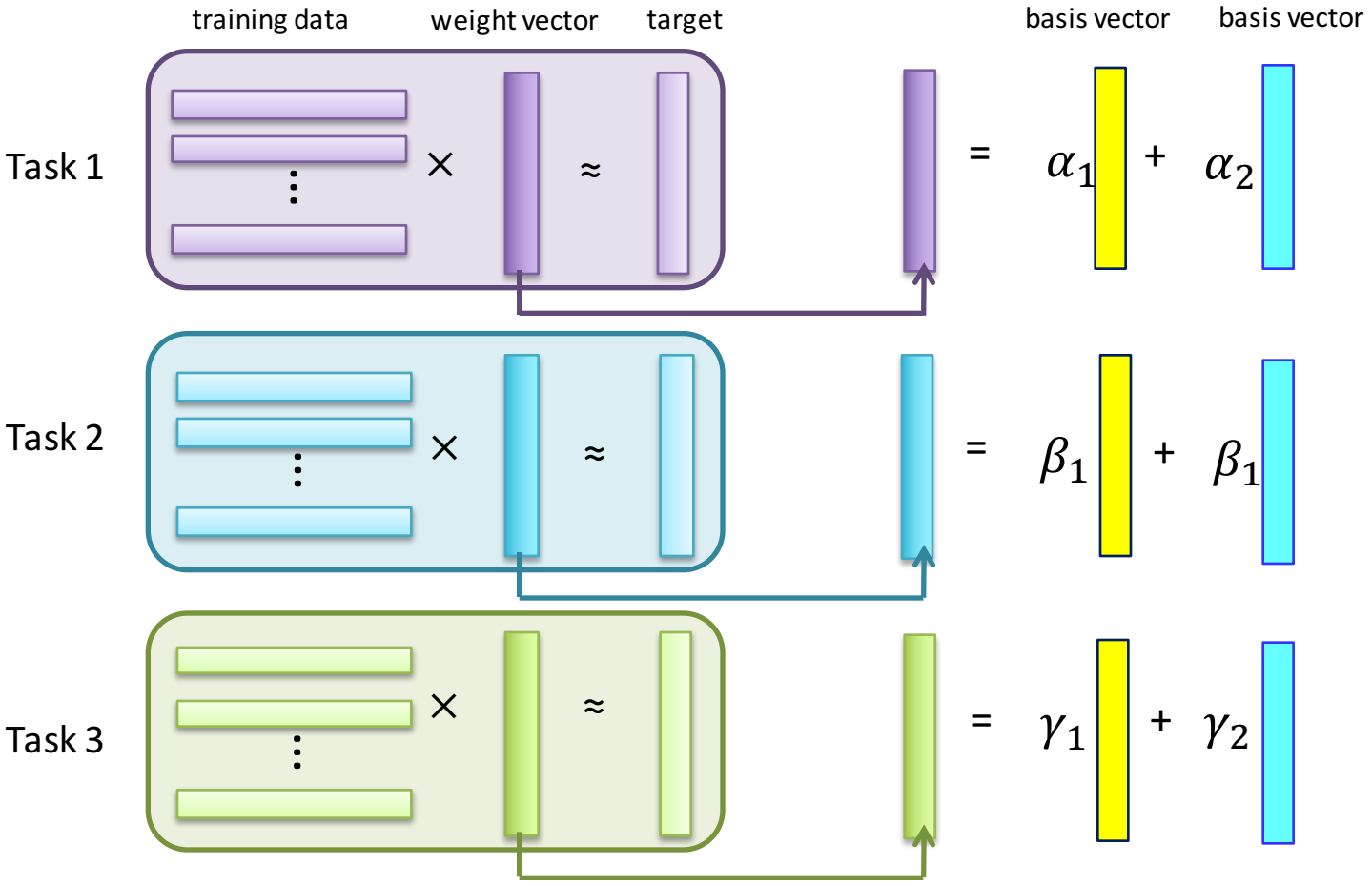
- Simultaneously captures a common set of features among relevant tasks and identifies outlier tasks.



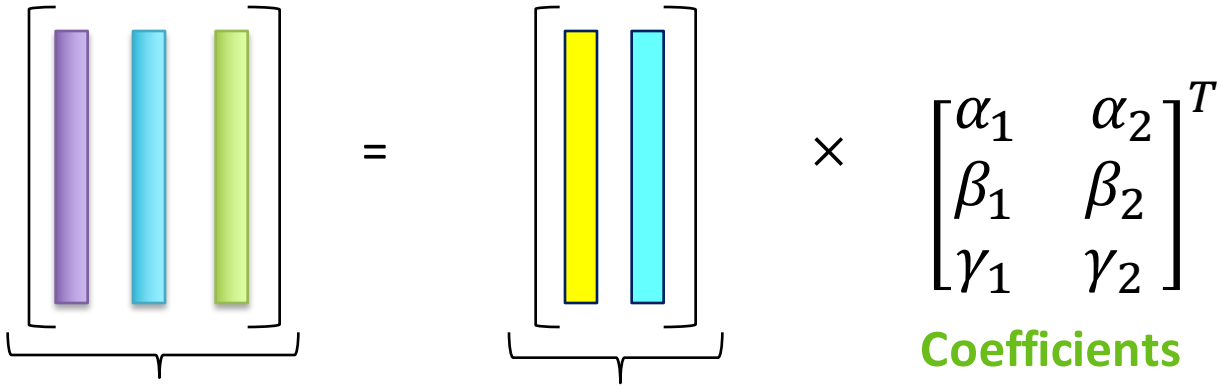
$$\min_{P,Q} \|Y - X(P+Q)\|_F^2 + \lambda_1 \|P\|_{1,q} + \lambda_2 \|Q^T\|_{1,q}$$

Low-Rank Structure for MTL

- Capture task relatedness via a shared low-rank structure



Low-Rank Structure for MTL (Cont.)



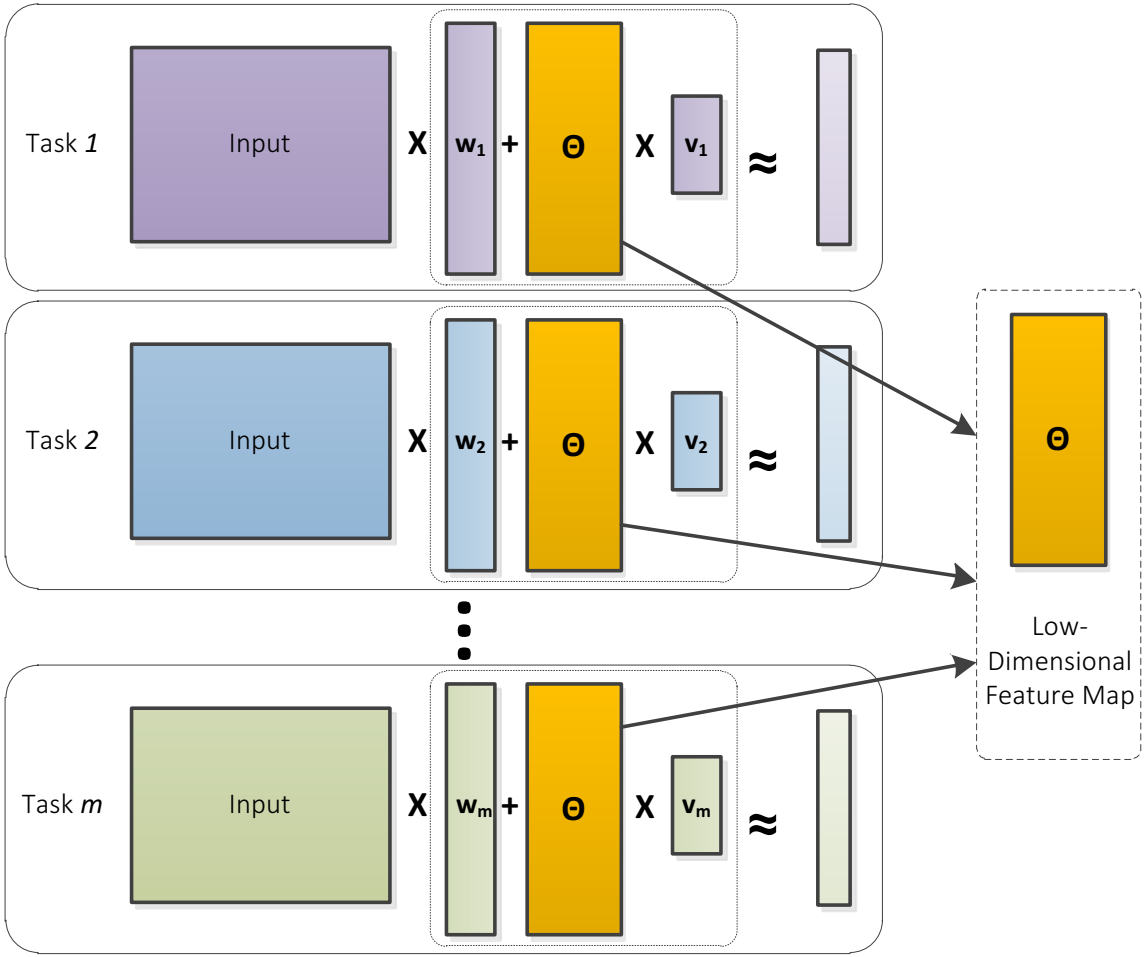
- Rank minimization formulation
  - $\min_W \text{Loss}(W) + \lambda \times \text{Rank}(W)$
- Rank minimization is *NP-Hard* for general loss functions thus we use convex relaxation: trace norm minimization
  - $\min_W \text{Loss}(W) + \lambda \times \|W\|_*$

Regularization
Encourages low-rank on the model matrix

Alternating Structure Optimization (ASO)

Ando and Zhang, 2005 JMRL

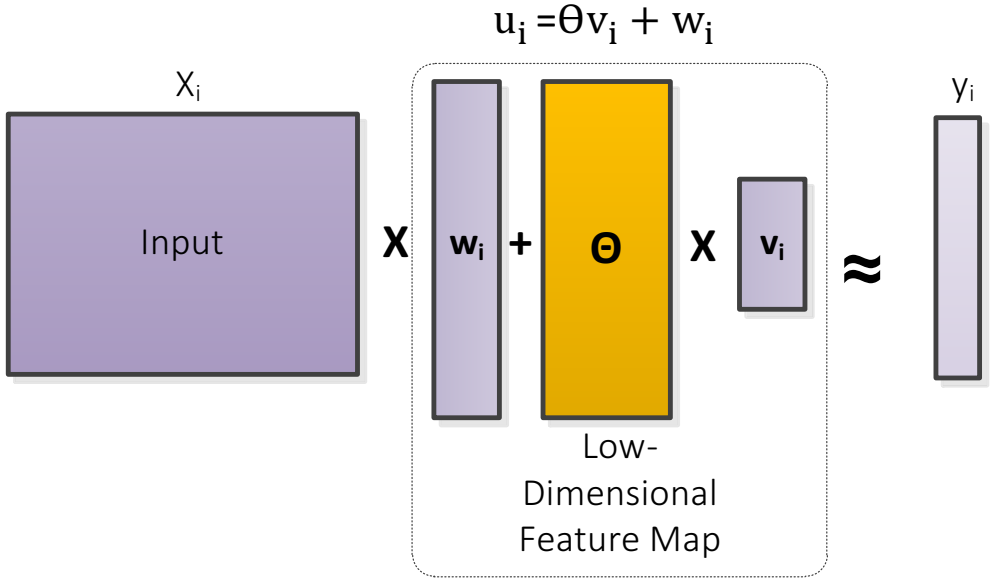
- ASO assumes that the model is the sum of two components: a task specific one and a shared low dimensional subspace.



Alternating Structure Optimization (ASO)

Ando and Zhang, 2005 JMRL

- Learning from the i-th task



$$\min_{\Theta,\{v_i,w_i\}} \sum_{i=1}^{m} \{\mathcal{L}_i(X_i(\Theta v_i + w_i), y_i) + \alpha \|w_i\|^2\}$$

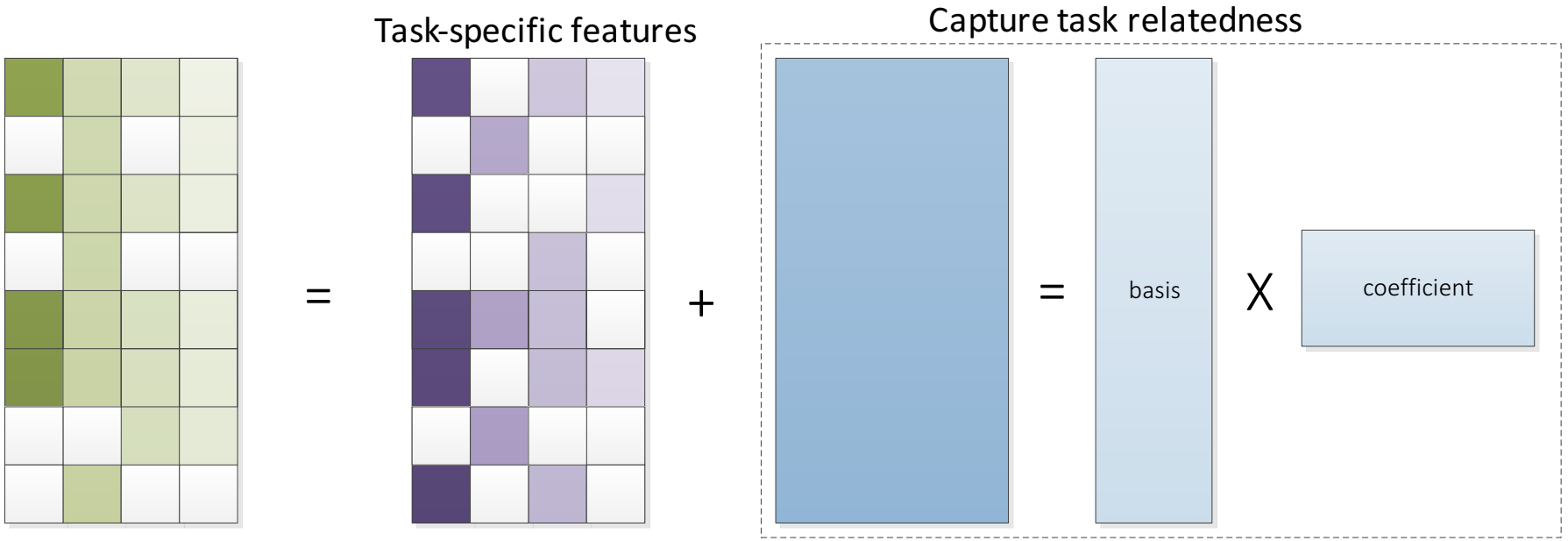
subject to $\Theta^T\Theta = I$

$$\mathcal{L}_i(X_i(\Theta v_i + w_i), y_i) = \|X_i(\Theta v_i + w_i) - y_i\|^2$$

Incoherent Low-Rank and Sparse Structures

Chen *et. al.* 2010 KDD

- ASO uses L2-norm on task-specific component, we can also use L1-norm to learn task-specific features.



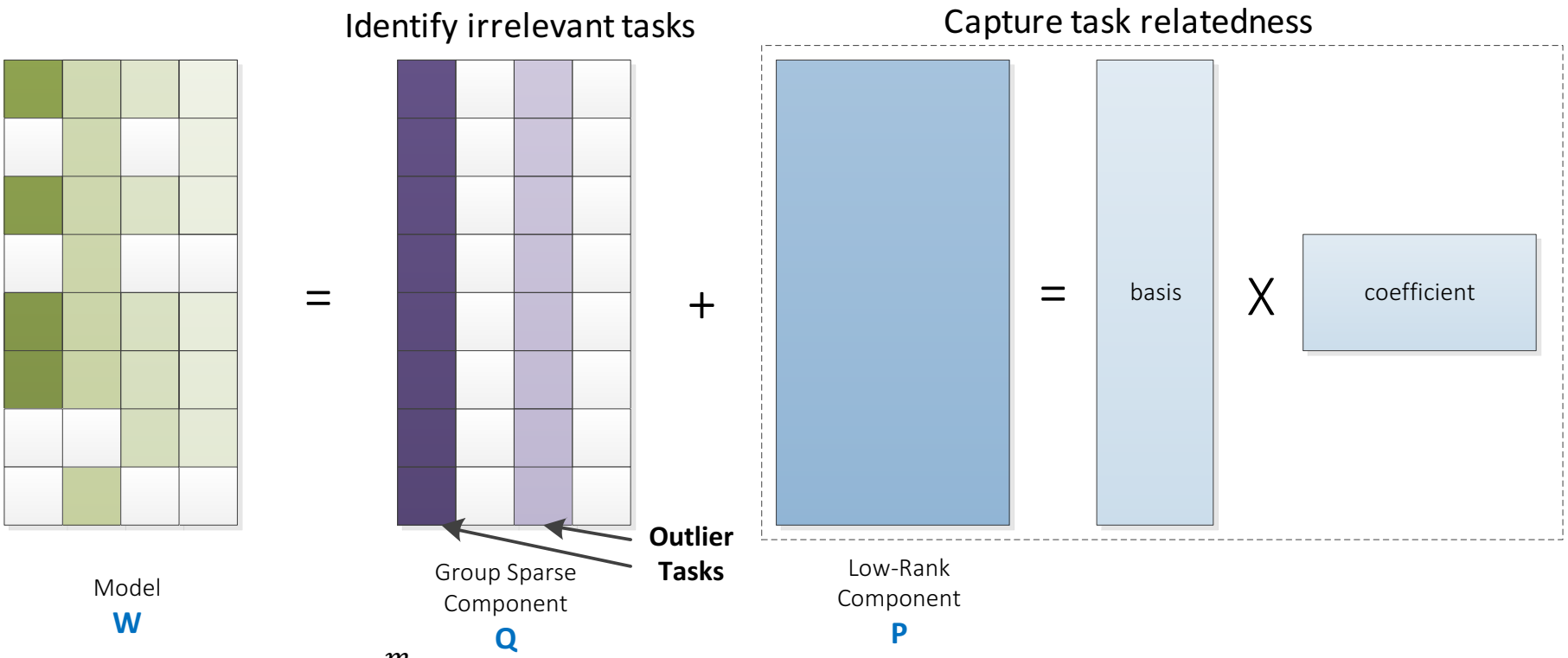
$$\min_{P,Q} \sum_{i=1}^{m} \mathcal{L}_i(X_i(P_i + Q_i), y_i) + \lambda \|Q\|_1$$
$$\text{subject to} \quad \|P\|_* \leqslant \eta$$

Convex formulation

Robust Low-Rank in MTL

Chen *et. al.* 2011 KDD

- Simultaneously perform low-rank MTL and identify outlier tasks.



$$\min_{P,Q} \sum_{i=1}^{m} \mathcal{L}_i\left(X_i(P_i + Q_i), y_i\right) + \alpha\|P\|_* + \beta\left\|Q^T\right\|_{1,q}$$

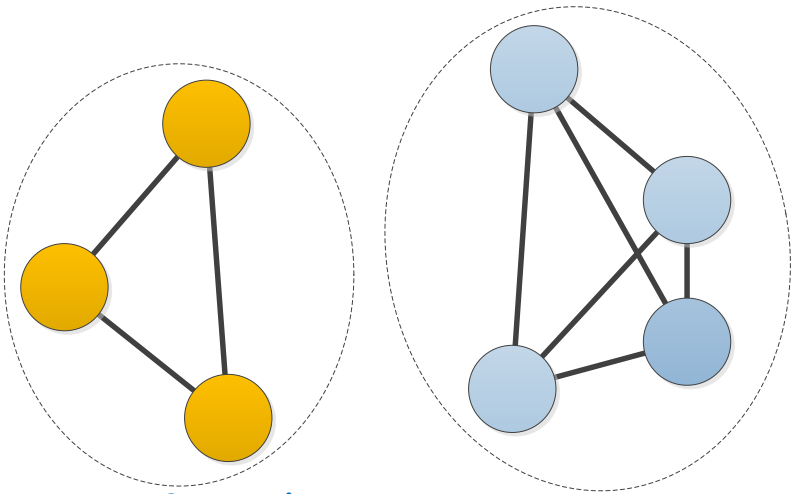
Summary

- All multi-task learning formulations discussed above can fit into the **W=P+Q** schema.
  - Component **P**: shared structure
  - Component **Q**: information not captured by the shared structure

Embedded Feature Selection	Shared Structure P	Component Q
L1/Lq	Feature Selection (L1/Lq Norm)	0
Dirty	Feature Selection (L1/Lq Norm)	L1-norm
rMTFL	Feature Selection (L1/Lq Norm)	Outlier (column-wise L1/Lq Norm)
Low-Rank Subspace Learning		
Trace Norm	Low-Rank (Trace Norm)	0
ISLR	Low-Rank (Trace Norm)	L1-norm
ASO	Low-Rank (Shared Subspace)	L2-norm on independent comp.
RMTL	Low-Rank (Trace Norm)	Outlier (column-wise L1/Lq Norm)

Multi-Task Learning with Clustered Structures

- Most MTL techniques assume all tasks are related
- Not true in many applications
- Clustered multi-task learning assumes
  - ❖ the tasks have a group structure
  - ❖ the models of tasks from the same group are closer to each other than those from a different group

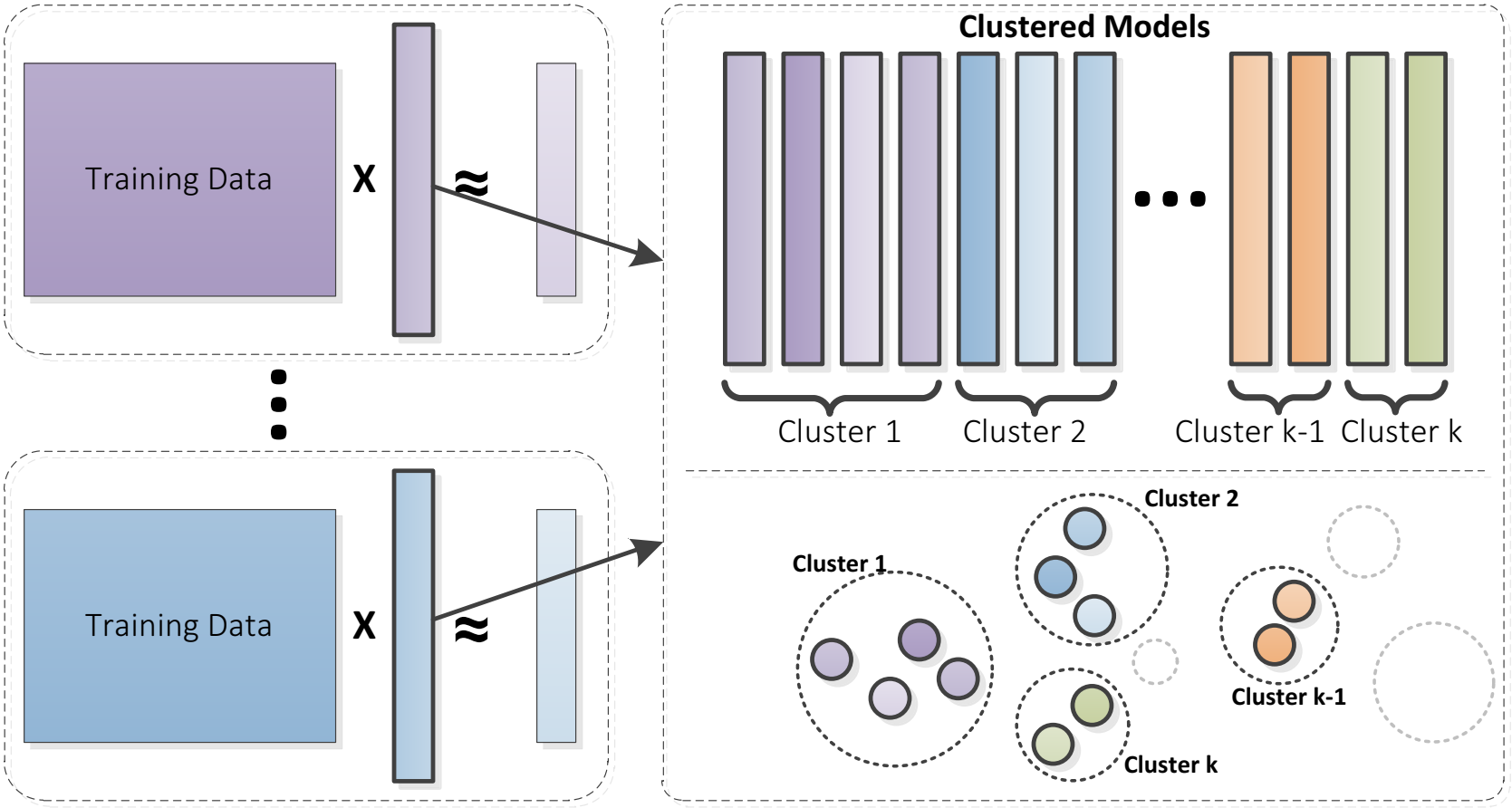


Assumption:
Tasks have group structures

Clustered Multi-Task Learning

Jacob *et. al.* 2008 NIPS, Zhou *et. al.* 2011 NIPS

- Use regularization to capture clustered structures.



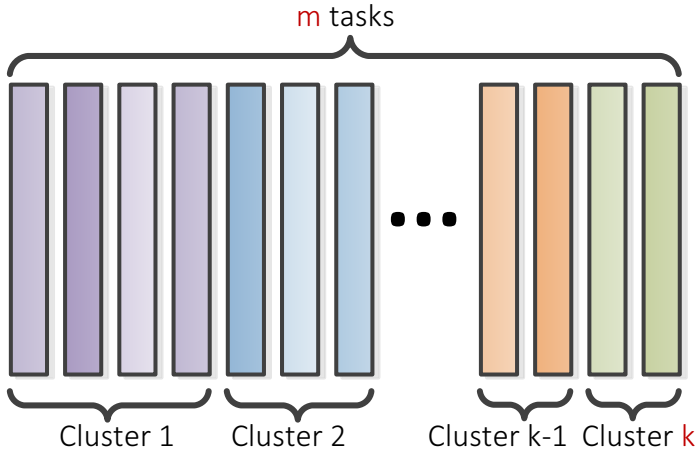
Clustered Multi-Task Learning

Zhou *et. al.* 2011 NIPS

- Capture structures by minimizing sum-of-square error (SSE) in K-means clustering:

$$\min_{I} \sum_{j=1}^{k} \sum_{v \in I_j} \left\| w_v - \bar{w}_j \right\|_2^2$$

I_j index set of j^{th} cluster



task number m > cluster number k

Equivalent

$$\min_{F} \operatorname{tr}(W^T W) - \operatorname{tr}(F^T W^T W F)$$

F : $m \times k$ orthogonal cluster indicator matrix

$F_{i,j} = 1/\sqrt{n_j}$ if $i \in I_j$ and 0 otherwise

Clustered Multi-Task Learning

Zhou *et. al.* 2011 NIPS

- Directly minimizing SSE is hard because of the non-linear constraint on F:

$$\min_F \operatorname{tr}(W^T W) - \operatorname{tr}(F^T W^T W F)$$

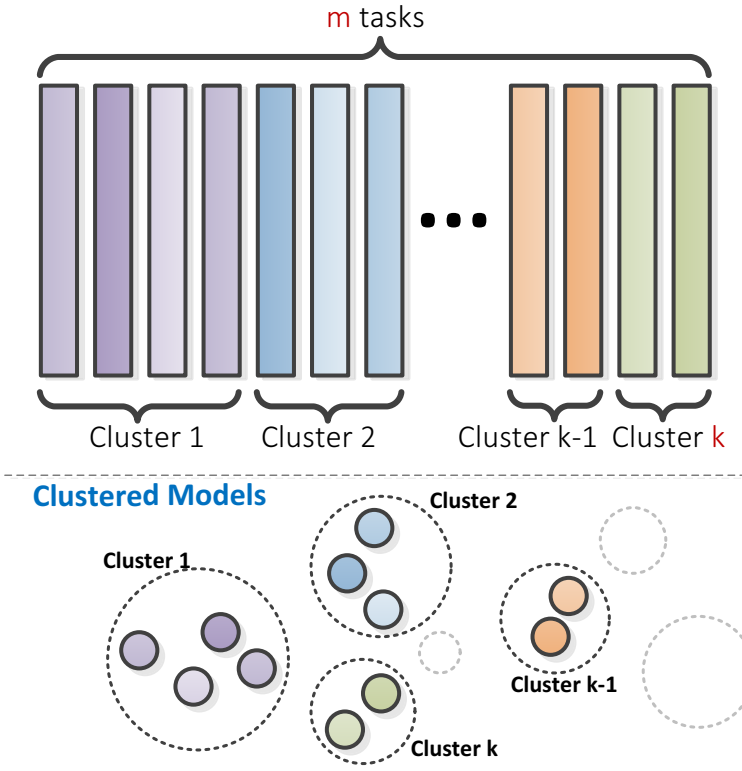
$F : m \times k$ orthogonal cluster indicator matrix

$F_{i,j} = 1/\sqrt{n_j}$ if $i \in I_j$ and 0 otherwise

Spectral Relaxation

$$\min_{F:F^T F = I_k} \operatorname{tr}(W^T W) - \operatorname{tr}(F^T W^T W F)$$

Zha et. al. 2001 NIPS



task number m > cluster number k

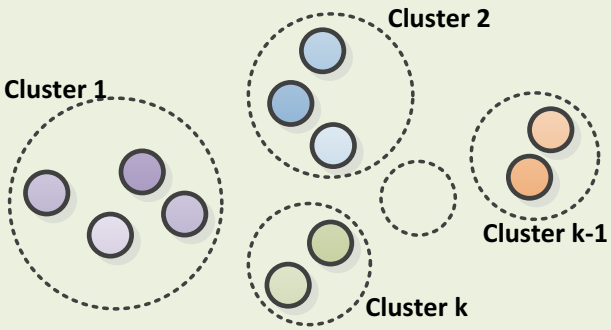
Clustered Multi-Task Learning

Zhou *et. al.* 2011 NIPS

- Clustered multi-task learning (CMTL) formulation

$$\min_{W,F:F^TF=I_k} \text{Loss(W)} + \alpha[\text{tr}(W^TW) - \text{tr}(F^TW^TWF)] + \beta\,\text{tr}(W^TW)$$

$\alpha[\text{tr}(W^TW) - \text{tr}(F^TW^TWF)]$: capture cluster structures

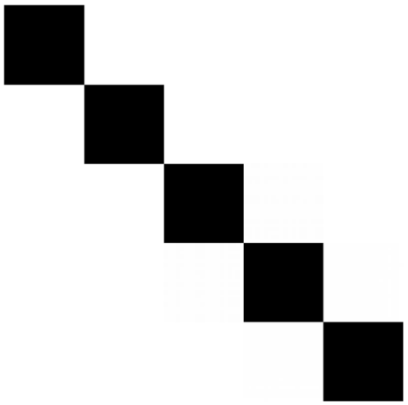


$\beta\,\text{tr}(W^TW)$: Improves generalization performance

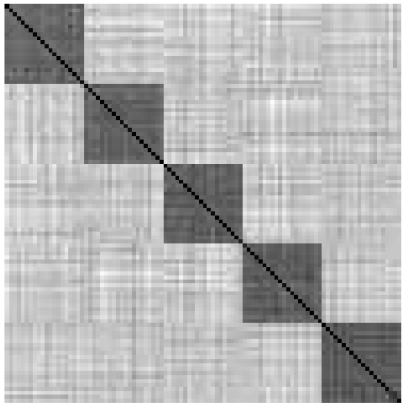
- CMTL has been shown to be equivalent to another class of MTL called ASO
  - Given the dimension of the shared low-rank subspace in ASO and the cluster number in clustered multi-task learning (CMTL) are the same.

Convex Clustered Multi-Task Learning

Zhou *et. al.* 2011 NIPS



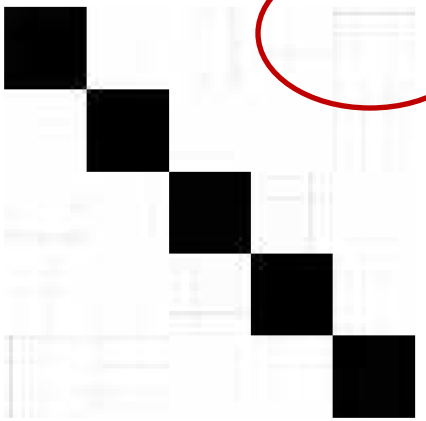
Ground Truth



Mean Regularized MTL

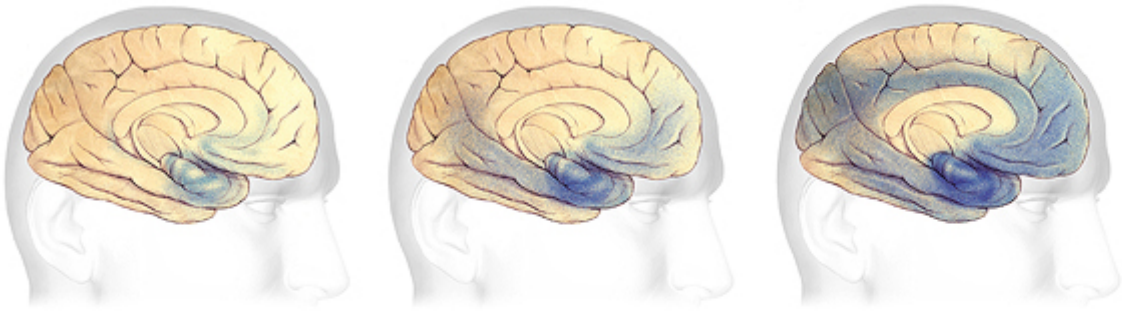
Low rank can also well capture cluster structure

Trace Norm Regularized MTL



noise introduced by relaxations

Convex Relaxed CMTL



Modeling Disease Progression via Multi-Task Learning

Multi-Task Learning Application


LEADING ACTRESS
JULIANNE MOORE
BAFTA NOMINEE FILM
GOLDEN GLOBE® AWARD
WINNER
JULIANNE MOORE - BEST ACTRESS, DRAMA
BEST ACTRESS
JULIANNE MOORE
ACADEMY AWARD® NOMINEE
"JULIANNE MOORE GIVES A SENSITIVE, SHATTERING AND BRILLIANT PERFORMANCE"
KATE MUIR, THE TIMES
"AN EFFORTLESSLY EXCELLENT FILM"
CATHERINE SHOARD, THE GUARDIAN
TIME OUT
THE TELEGRAPH
"EXTREMELY MOVING"
TIM ROBEY, THE TELEGRAPH
STILL ALICE
JULIANNE MOORE ALEC BALDWIN KRISTEN STEWART
A FILM BY RICHARD GLATZER AND WASH WESTMORELAND
StillAlice.co.uk
StillAliceFilm
#StillAlice
Alzheimer's Society
CURZON

Alzheimer's disease

Also called: senile dementia

ABOUT | SYMPTOMS | TREATMENTS

A progressive disease that destroys memory and other important mental functions.

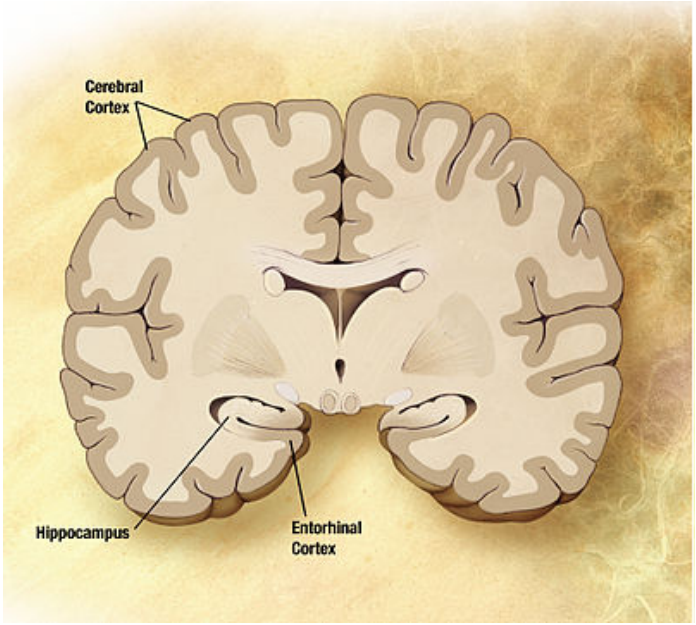
Very common

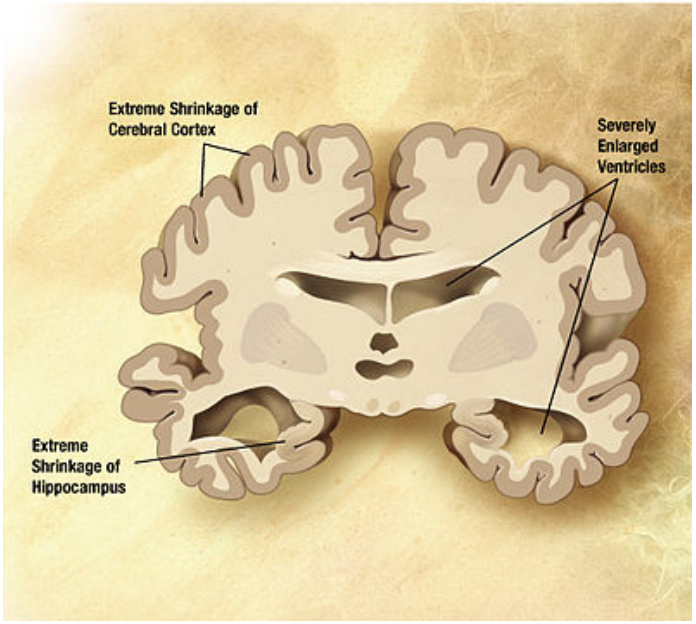
More than **3 million** US cases per year

- Requires a medical diagnosis
- Lab tests or imaging not required
- Chronic: can last for years or be lifelong

Consult a doctor for medical advice

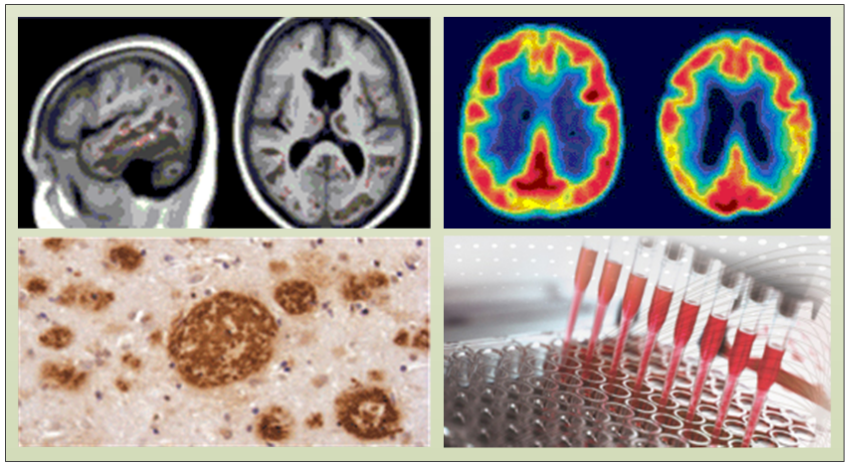
Sources: Mayo Clinic and others.

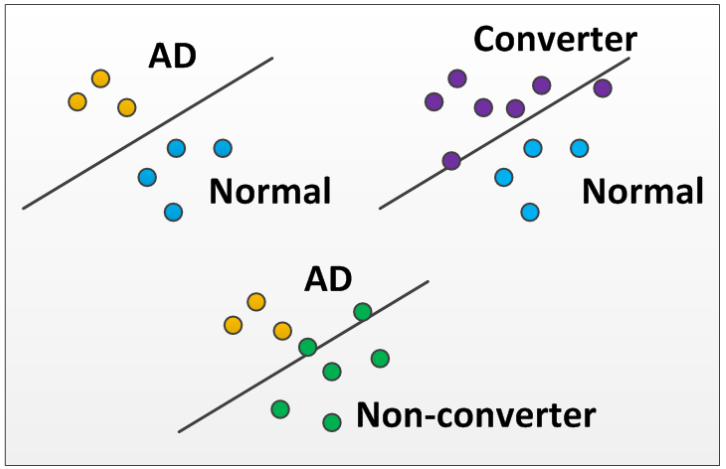




Background (cont.)

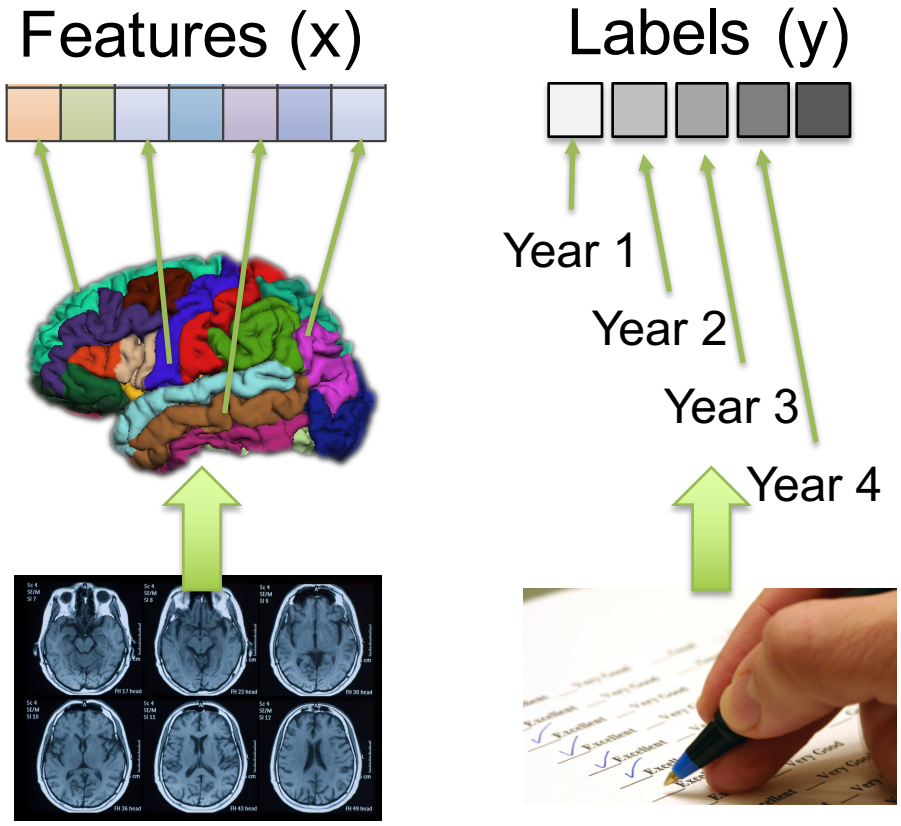
- NIH in 2003 funded the Alzheimer's Disease Neuroimaging Initiative (ADNI), facilitating a public available database for using neuroimaging data in predicting the progression of AD.





Disease Progression

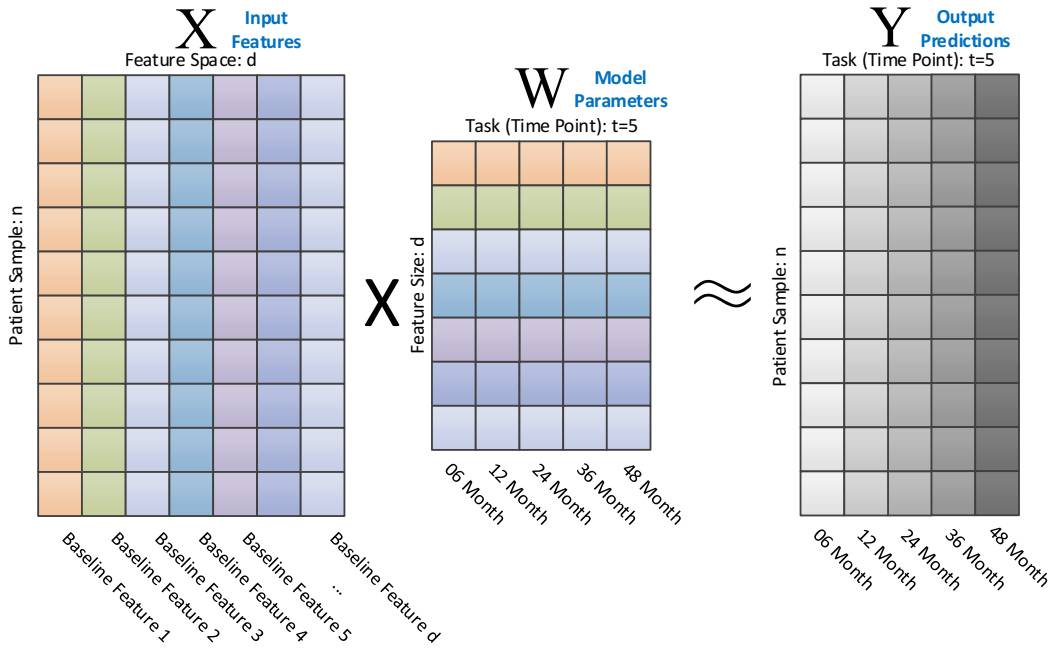
- Clinical scores are used to evaluate the cognitive status
  - MMSE, ADAS-Cog and etc.
- Disease progression
  - Prediction of clinical scores from neuroimaging features
  - Build one regression model at each time point.



Disease Progression (cont.)

- Disease progression as machine learning tasks
  - Build one regression model at each time point.

Regression minimize: $L(W) = \|(XW - Y)\|_F^2$



Model I: Temporal Group Lasso (TGL)

$$\min_W L(W) + \theta_1 \|W\|_F^2 + \theta_2 \|WH\|_F^2 + \delta \|W\|_{2,1}$$

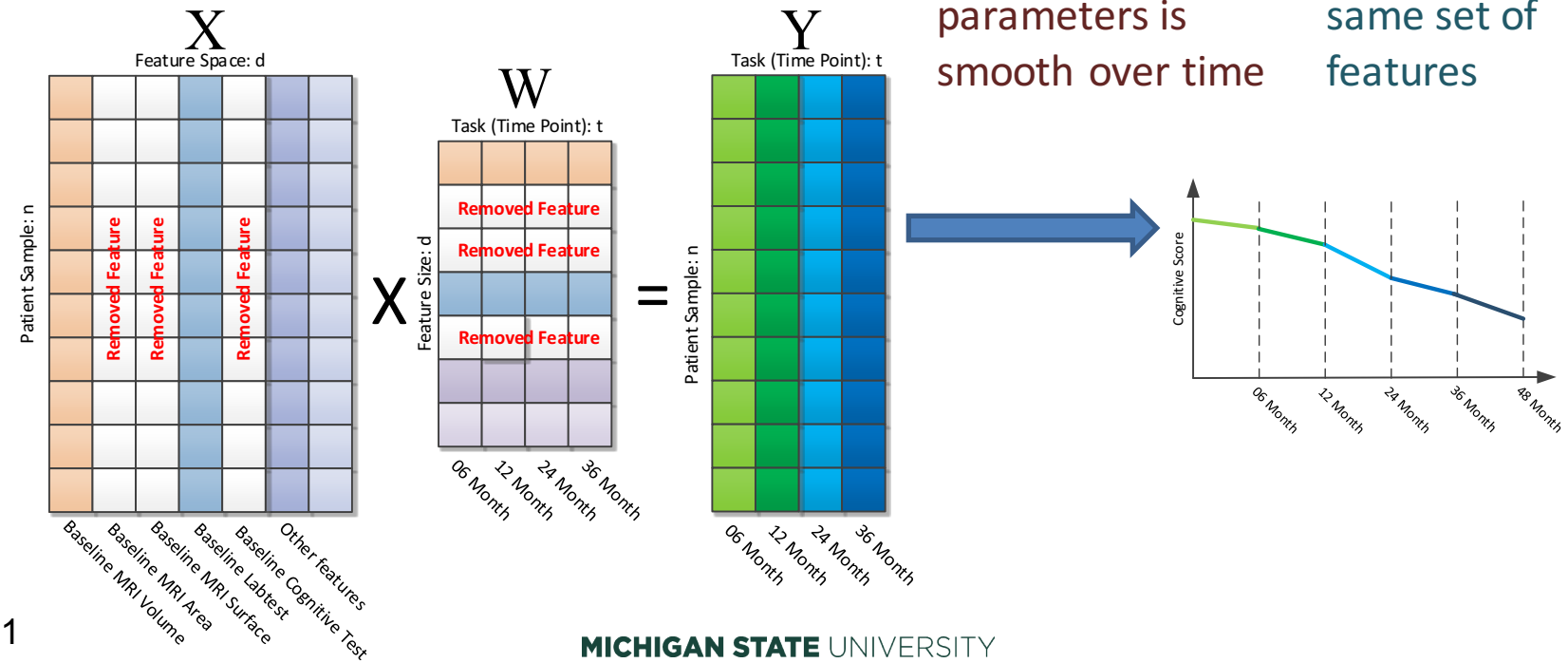
Loss Function
Performs regression

$$L(W) = \|(XW - Y)\|_F^2$$

Prevent Overfitting
Improves generalization performance

Temporal Smoothness
For each feature, the change of parameters is smooth over time

Group Sparse
Models at different time points share the same set of features



Model II: Fused Sparse Group Lasso (FSGL)

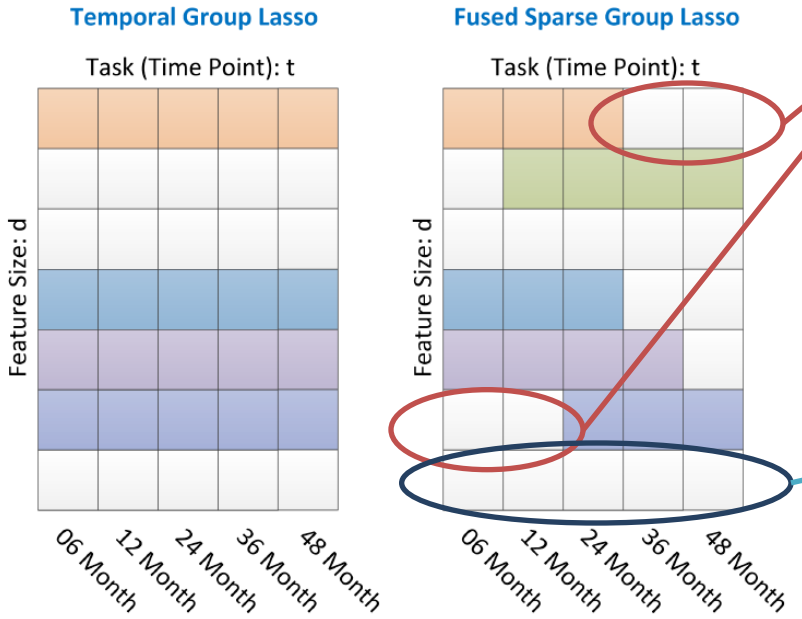
$$\min_W L(W) + \lambda_1 \|W\|_1 + \lambda_2 \|RW^T\|_1 + \lambda_3 \|W\|_{2,1}$$

Loss Function Performs regression

Element-wise Sparse Improves generalization performance

Sparse Temporal Smoothness via Fused Lasso
For each feature parameter, the change of values and sparse pattern of parameters is smooth over time

Group Sparse Models at different time points share the same set of features



Optimization Algorithm

- Objective is convex but non-smooth
  - Objective is smooth + non-smooth composite
  - Projected gradient/accelerated projected gradient
  - Key: proximal operator (Euclidean projection)

$$\pi(V) = \arg\min_W \frac{1}{2}\|W - V\|_F^2 + \lambda_1\|W\|_1 + \lambda_2\|RW^T\|_1 + \lambda_3\|W\|_{2,1}$$

Can be decomposed into two simpler problems and solved efficiently

THEOREM 1. *Define*

$$\pi_{\mathrm{FL}}(\mathbf{v}) = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{v}\|_2^2 + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|R\mathbf{w}\|_1 \quad (5)$$

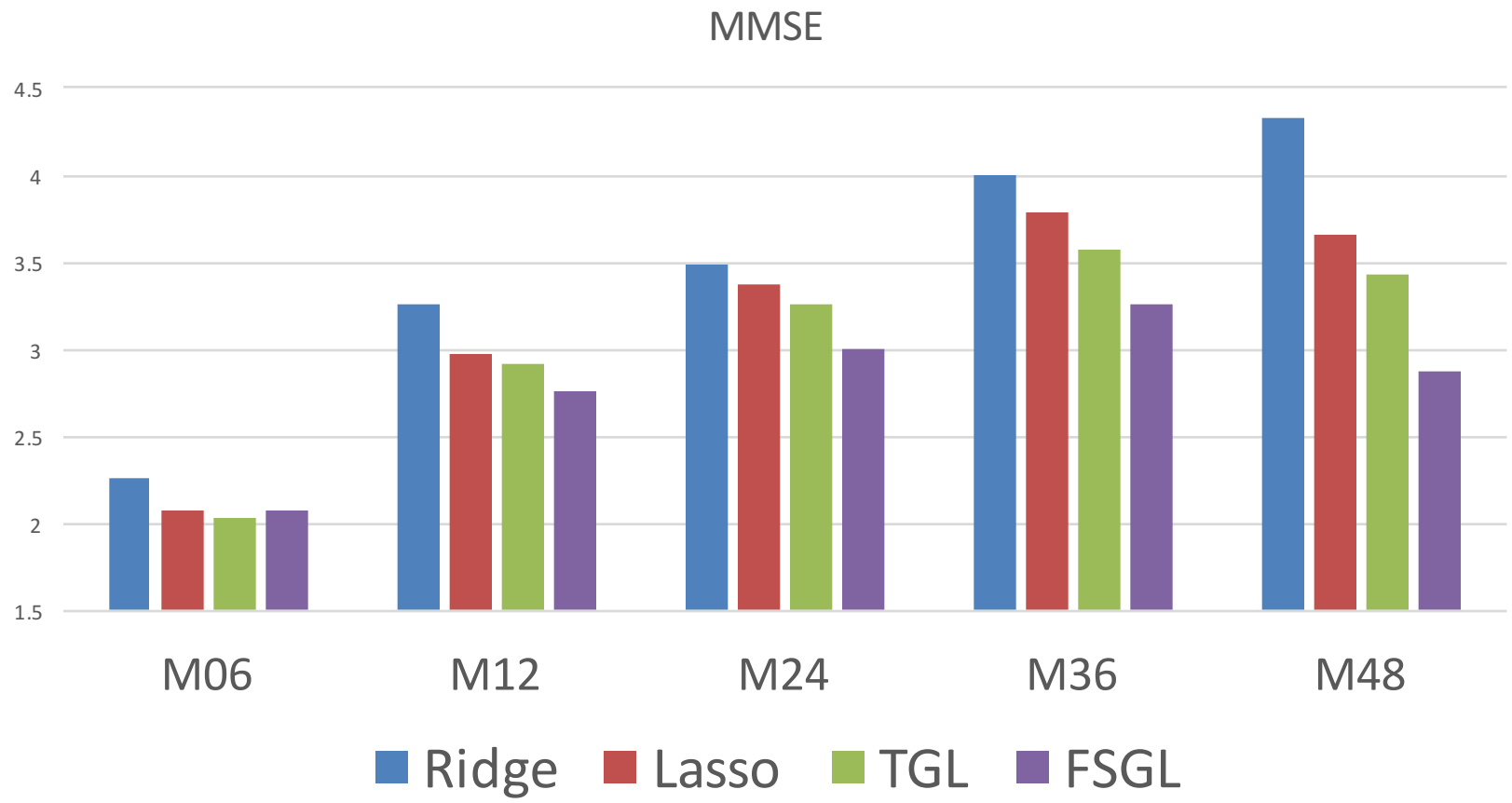
$$\pi_{\mathrm{GL}}(\mathbf{v}) = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{v}\|_2^2 + \lambda_3\|\mathbf{w}\|_2. \quad (6)$$

Then the following holds:

$$\pi(\mathbf{v}) = \pi_{\mathrm{GL}}(\pi_{\mathrm{FL}}(\mathbf{v})). \quad (7)$$

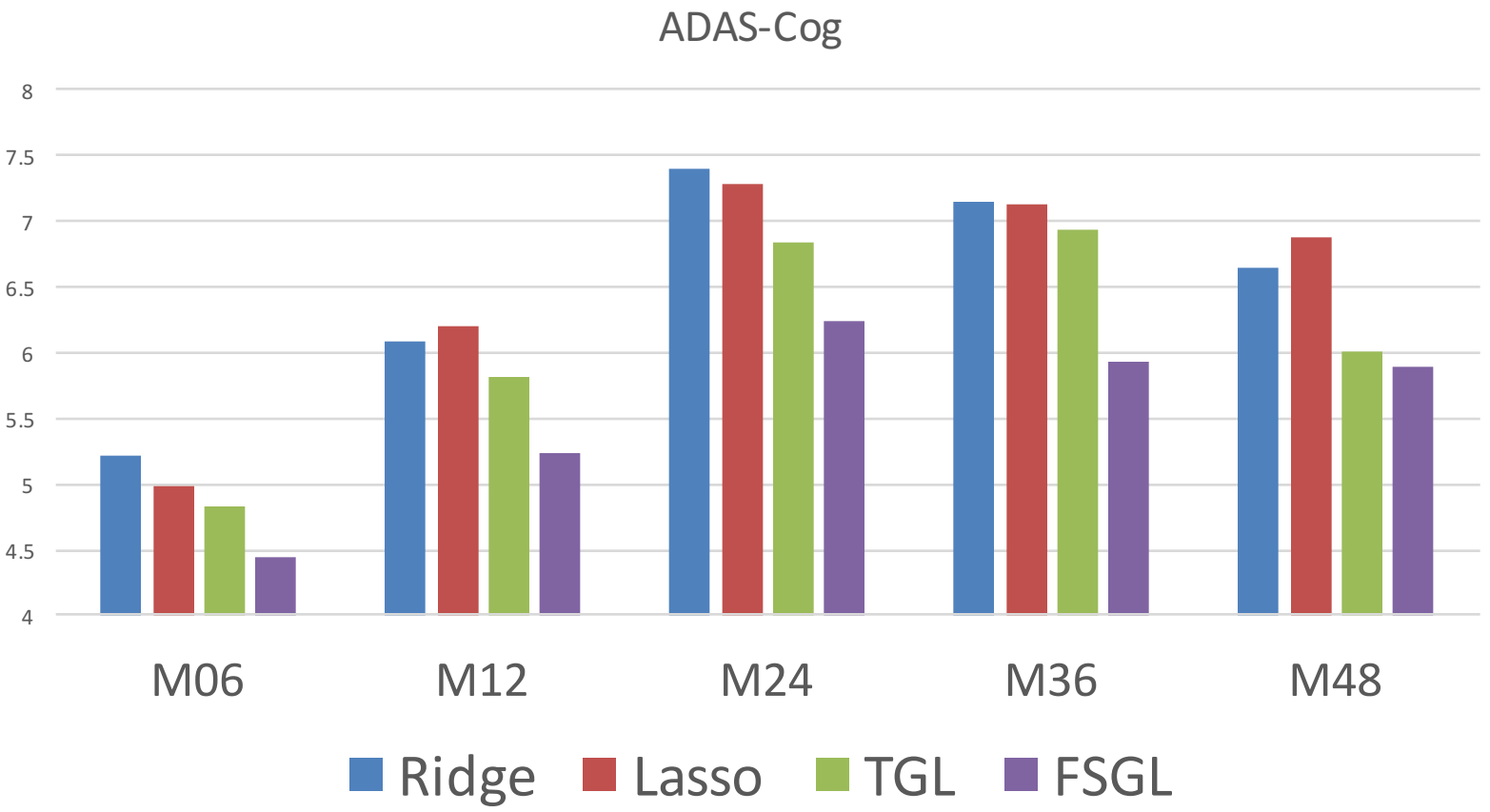
Performance

- Use baseline MRI feature to predict future MMSE score
- Average performance over 10 iterations



Performance (cont.)

- Use baseline MRI feature to predict ADAS-Cog score
- Average performance over 10 iterations

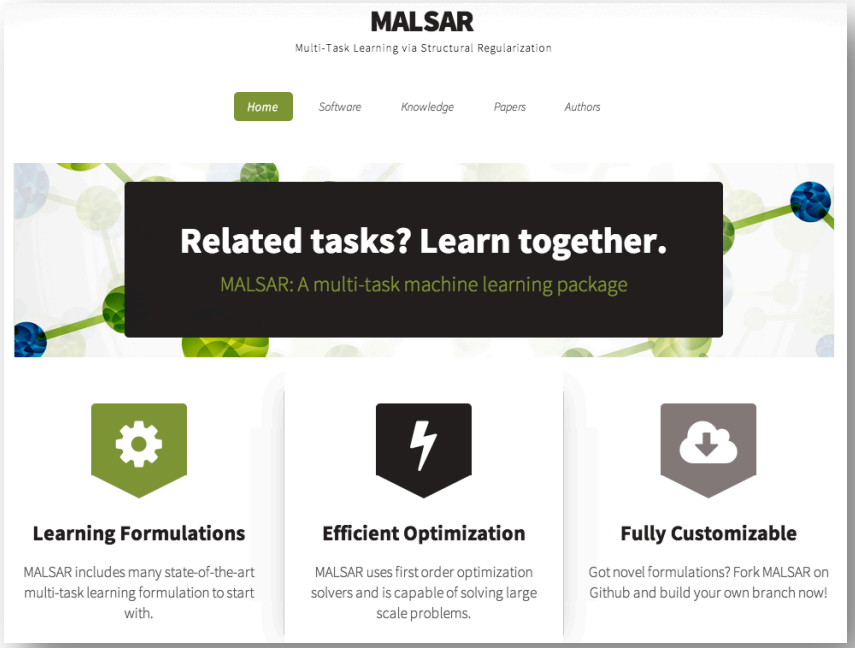


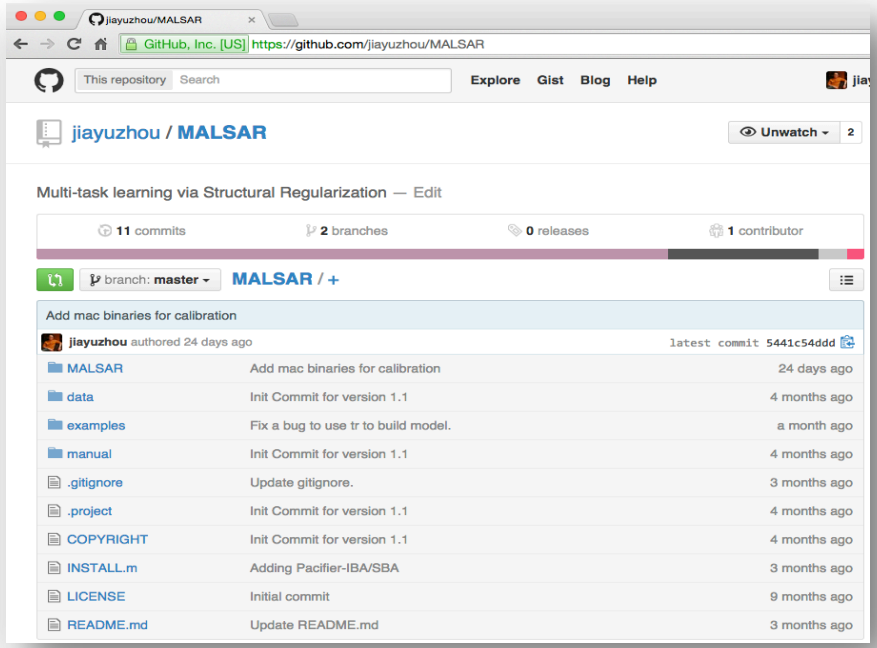
MALSAR: Multi-Task Learning via Structural Regularization

Multi-Task Learning Software

MALSAR

MULTI-TASK LEARNING VIA STRUCTURAL REGULARIZATION

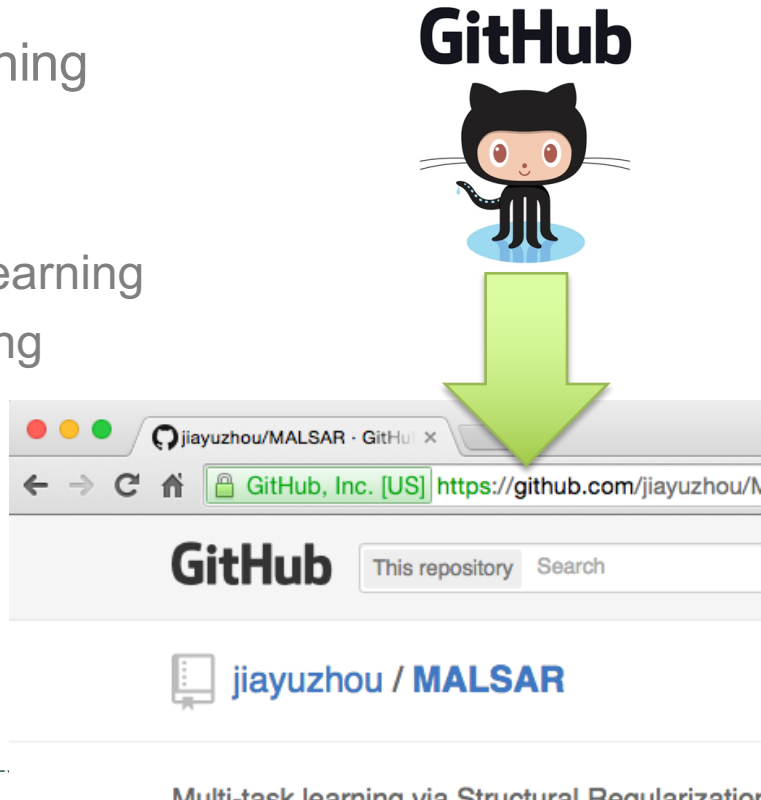




- Firstly introduced my MTL **tutorial** at **SDM** in 2012
- Many research works using MALSAR are published in KDD, NIPS, TPAMI, ICCV, ICDM, ICIP, COLING, MICCAI, ACM-MM, etc.
- Used as **course material** to analyze compound profiling in the *Strasbourg Summer School* in France

Some MTL Algorithms in MALSAR

- Mean-Regularized Multi-Task Learning
- MTL with Embedded Feature Selection
  - Joint Feature Learning
  - Dirty Multi-Task Learning
  - Robust Multi-Task Feature Learning
- MTL with Low-Rank Subspace Learning
  - Trace Norm Regularized Learning
  - Alternating Structure Optimization
  - Incoherent Sparse and Low Rank Learning
  - Robust Low-Rank Multi-Task Learning
- Clustered Multi-Task Learning
- Graph Regularized
- Many more…



An Example

Create a random MTL dataset

Invoke an MTL algorithm

MALSAR/example_Dirty.m

GitHub, Inc. [US] https://github.com/jiayuzhou/MALSAR/blob/mas

```
35
36  clear;
37  clc;
38  close;
39
40  addpath('../MALSAR/functions/dirty/'); % load function
41  addpath('../MALSAR/c_files/prf_lbm/'); % load projection c libraries.
42  addpath('../MALSAR/utils/'); % load utilities
43
44  %rng('default');     % reset random generator. Available from Matlab 201
45
46  %generate synthetic data.
47  dimension = 500;
48  sample_size = 50;
49  task = 50;
50  X = cell(task ,1);
51  Y = cell(task ,1);
52  for i = 1: task
53      X{i} = rand(sample_size, dimension);
54      Y{i} = rand(sample_size, 1);
55  end
56
57  opts.init = 0;      % guess start point from data.
58  opts.tFlag = 1;     % terminate after relative objective value does not
59  opts.tol = 10^-4;   % tolerance.
60  opts.maxIter = 500; % maximum iteration number of optimization.
61
62  rho_1 = 350;%   rho1: group sparsity regularization parameter
63  rho_2 = 10;%   rho2: elementwise sparsity regularization parameter
64
65  [W funcVal P Q] = Least_Dirty(X, Y, rho_1, rho_2, opts);
66
67
```

Thanks!