Mining Structured Sparsity Beyond Convexity

Jiayu Zhou
Michigan State University

Pinghua Gong, Zheng Wang, Jieping Ye
University of Michigan

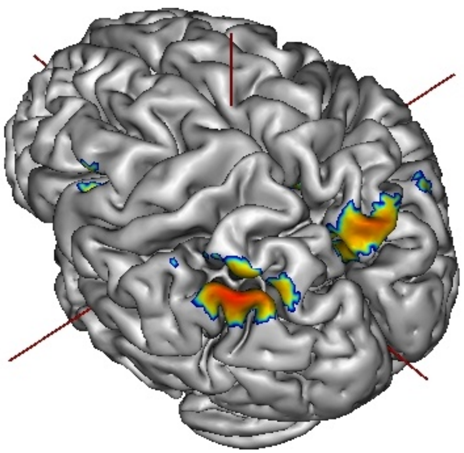
Road Map

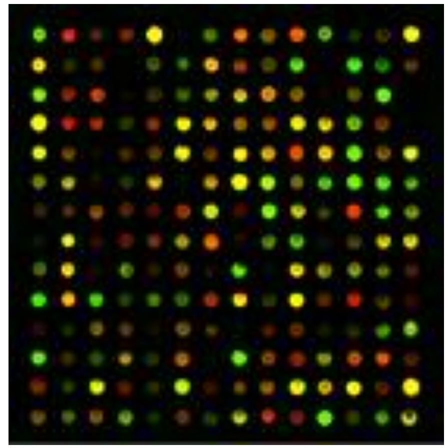
- Introduction to Sparsity
- Convex Approaches
- Non-Convex Approaches
- Topic: Matrix Completion
- Topic: Multi-task Learning

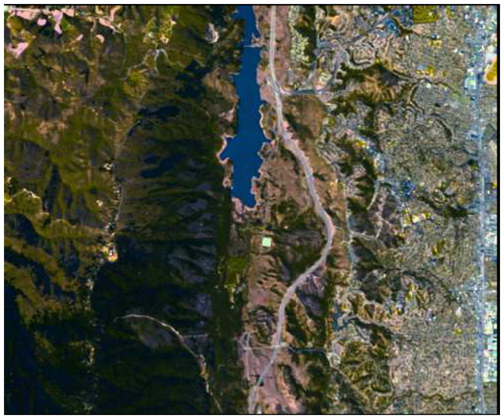
Road Map

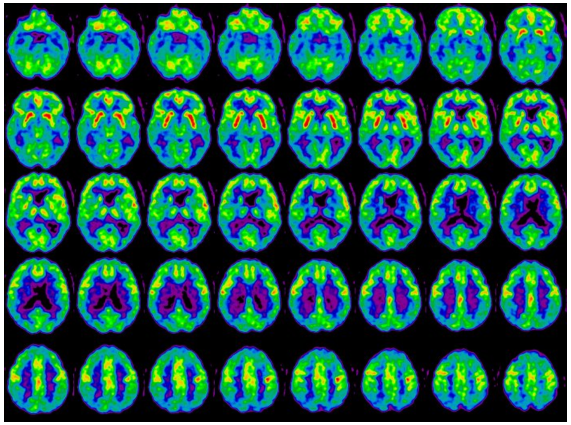
- **Introduction to Sparsity**
- Convex Approaches
- Non-Convex Approaches
- Topic: Matrix Completion
- Topic: Multi-task Learning

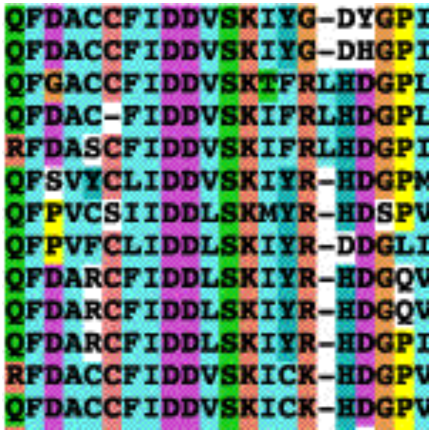
Mining High-Dimensional Data

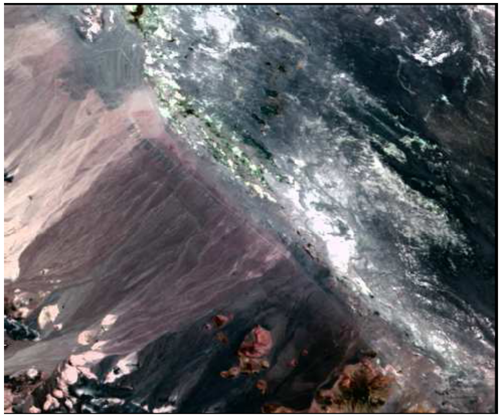












Dimensionality Reduction

- Dimensionality reduction algorithms
  - Feature Extraction
  - Feature Selection

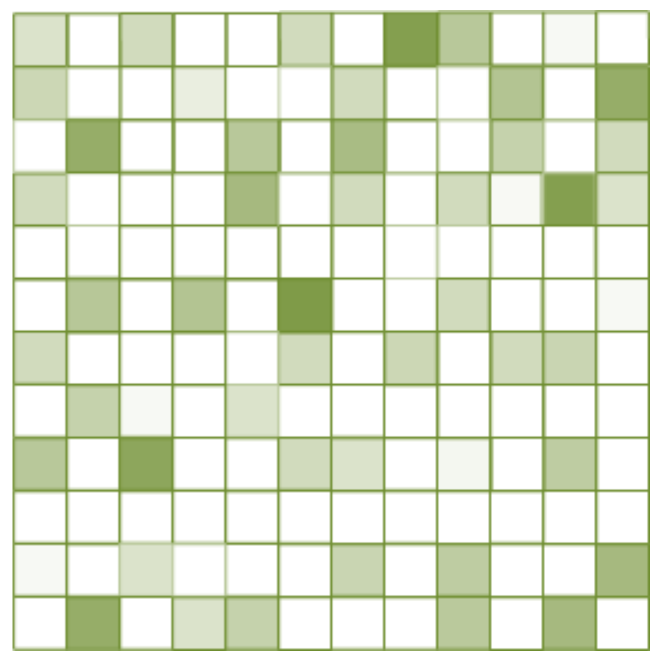
SIAM Data Mining 2007 Tutorial (Yu, Ye, and Liu):
"Dimensionality Reduction for Data Mining - Techniques, Applications, and Trends"

Sparse Learning

- We focus on sparse learning in this tutorial
  - Embed dimensionality reduction into data mining tasks
  - Flexible models for complex feature structures
  - Strong theoretical guarantee
  - Empirical success in many applications
  - Recent progress on efficient implementations

What is Sparsity

- Many data mining tasks can be represented using a vector or a matrix.
- “Sparsity” implies many zeros in a vector or a matrix.



Human Anatomy

Anatomy Lesson of Dr. Nicolaes Tulp by Rembrandt van Rijn, 1632.

Biomedical Imaging

X-Ray,1895

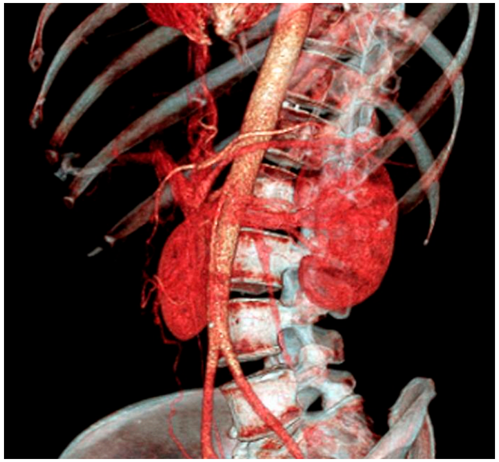
1901 Nobel Prize in Physics
Wilhelm Röntgen's

Biomedical Imaging

X-Ray,1895

1901 Nobel Prize in Physics
Wilhelm Röntgen's

Computed Tomography (CT), 1967



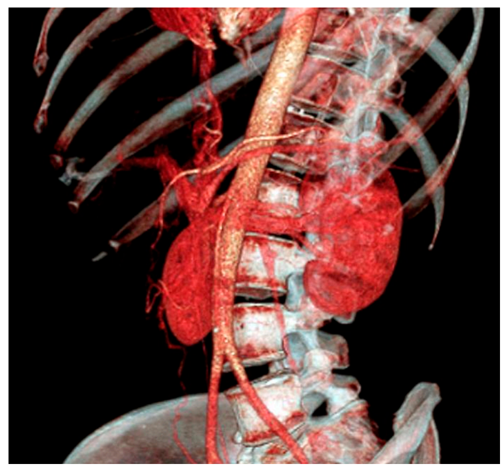
1979 Nobel Prize in Physiology or Medicine
Allan M. Cormack and Godfrey N. Hounsfield

Biomedical Imaging

X-Ray,1895

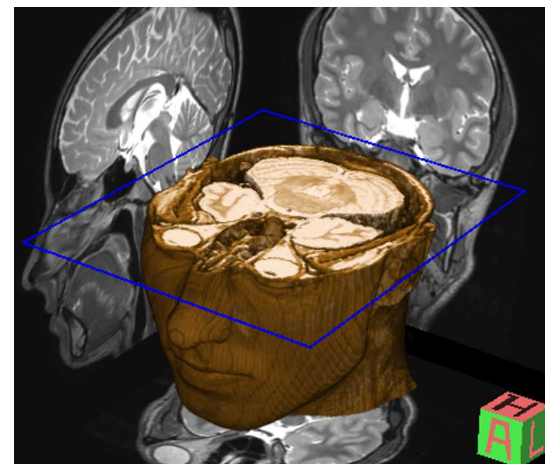
1901 Nobel Prize in Physics
Wilhelm Röntgen's

Computed Tomography (CT), 1967



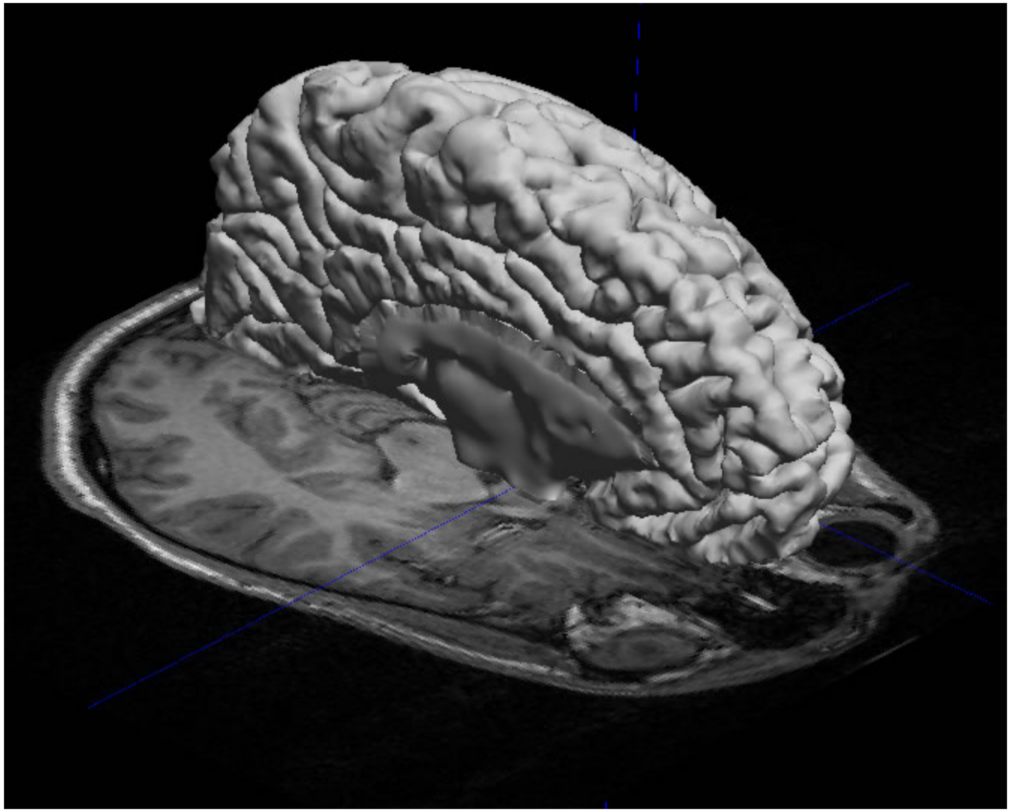
1979 Nobel Prize in Physiology or Medicine
Allan M. Cormack and Godfrey N. Hounsfield

Magnetic Resonance Imaging (MRI), 1971

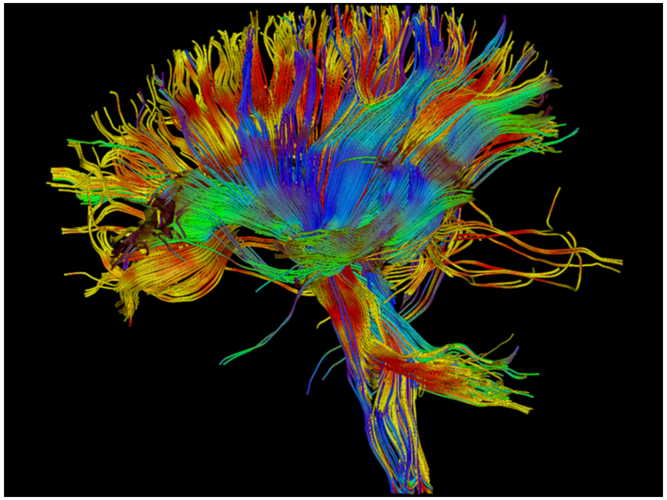


2003 Nobel Prize in Physiology or Medicine
Paul Lauterbur and Sir Peter Mansfield

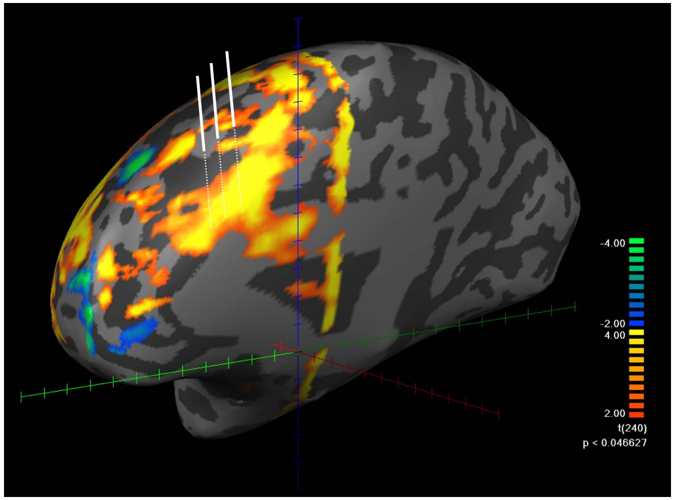
Magnetic Resonance Imaging



Structural



Diffusion



Functional

Magnetic Resonance Imaging (cont.)

- Acquire a digital object $x \in \mathbb{R}^p$ from *n* measurements:

$$y_i = \langle x, \varphi_i \rangle, i = 1, 2, \ldots, n$$

  - Waveforms φ_i : Sinusoids
    - *y* is a vector of Fourier coefficients (e.g., MRI)
- Recover the object from the measurements
  - Sovling a linear system of equations

Magnetic Resonance Imaging (cont.)

Compressive Sensing

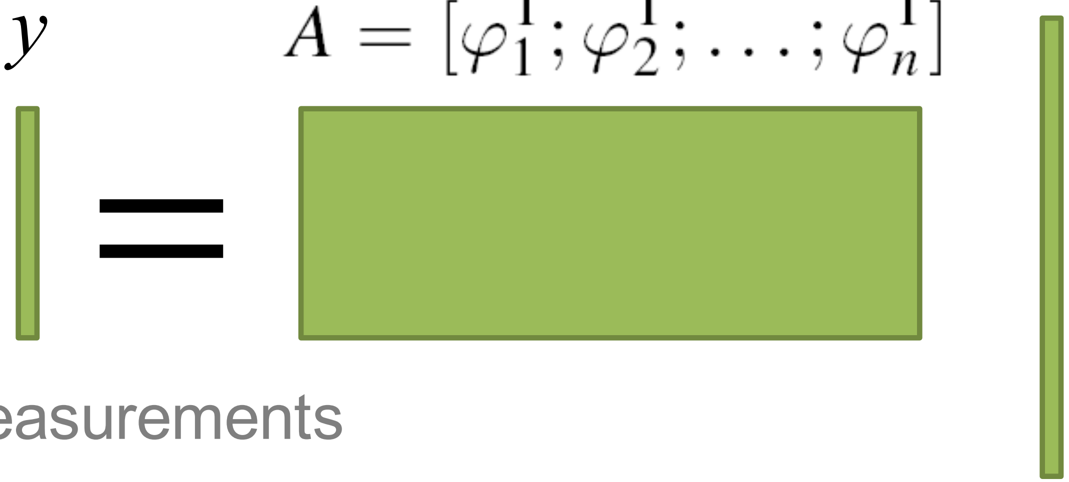
- Is accurate reconstruction possible from $n \ll p$ measurements only?
  - Few sensors
  - Measurements are very expensive
  - Sensing process is slow
  - Save lives

Motivation: Signal Acquisition

- Conventional wisdom: reconstruction is impossible
  - Number of measurements must match the number of unknowns

x

y $\quad A = [\varphi_1^{\mathrm{T}}; \varphi_2^{\mathrm{T}}; \ldots; \varphi_n^{\mathrm{T}}]$



$n \times 1$ measurements

If ***n<<p***, the system is underdetermined. $\quad$ *p*×1 signal

Generalization: Signal Acquisition

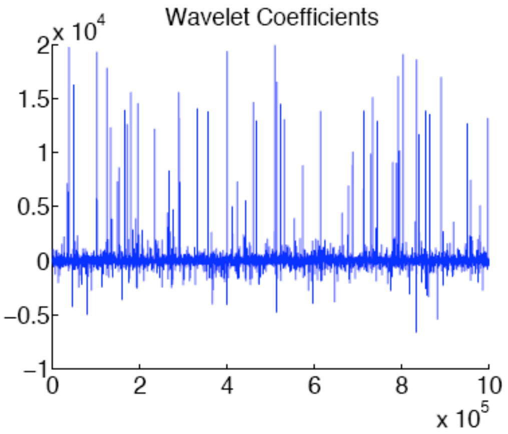
- Wish to acquire a digital object $x \in \mathbb{R}^p$ from *n* measurements:

$$y_i = \langle x, \varphi_i \rangle, i = 1, 2, \ldots, n$$

- Waveforms φ_i
  - Dirac delta functions (spikes)
    - *y* is a vector of sampled values of *x* in the time or space domain
  - Indicator functions of pixels
    - *y* is the image data typically collected by sensors in a digital camera
  - Sinusoids
    - *y* is a vector of Fourier coefficients (e.g., MRI)

Motivation: Signal Acquisition (cont.)

- Many natural signals are sparse or compressible in the sense that they have concise representations when expressed in the proper basis



Megapixel image represented as ***2.5%*** *largest wavelet coefficients*

(*Candes and Wakin, 2008*)

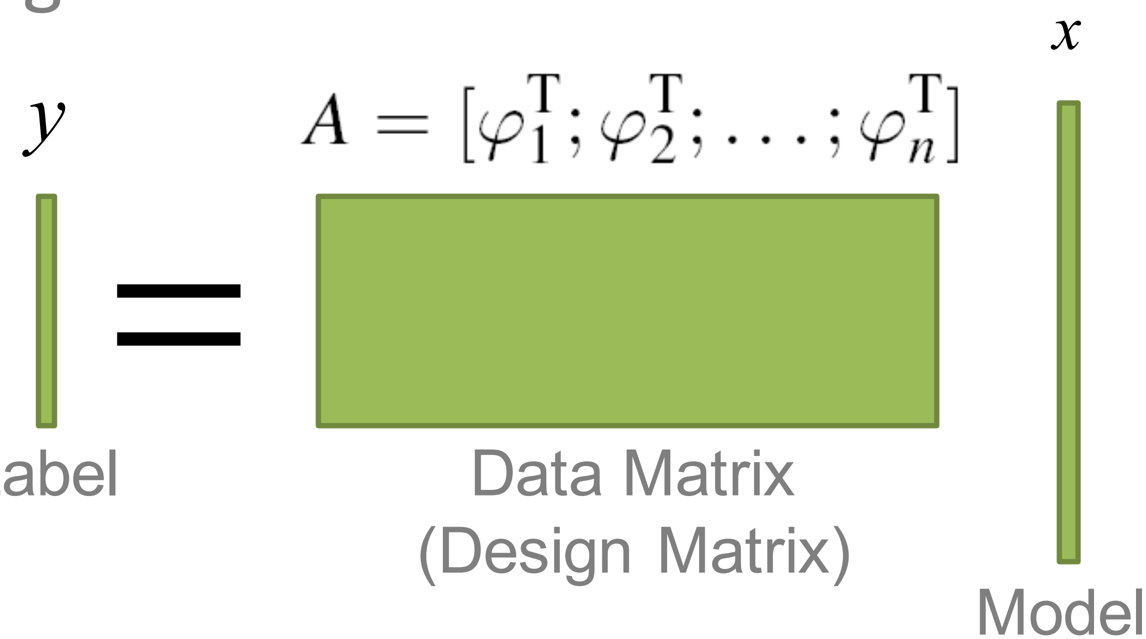
MRI by Compressive Sensing

Sparsity

- Dominant modeling tool
  - Genomics
  - Genetics
  - Signal and audio processing
  - Image processing
  - Neuroscience (theory of sparse coding)
  - Machine learning
  - Data mining
  - …

Sparsity in Data Mining

- Regression, classification, collaborative filtering…



Road Map

- Introduction to Sparsity
- **Convex Approaches**
- Non-Convex Approaches
- Topic: Matrix Completion
- Topic: Multi-task Learning

Convex Sparse Learning Models

- Let x be the model parameter to be estimated. A commonly employed model for estimating x is

$$\min \ \text{loss}(x) + \lambda \times \text{penalty}(x) \qquad (1)$$

- (1) is equivalent to the following model:

$$\begin{aligned} &\min \ \text{loss}(x) \\ &\text{s.t.} \quad \text{penalty}(x) \leq z \end{aligned} \qquad (2)$$

Convex Sparse Learning Models

- Let x be the model parameter to be estimated. A commonly employed model for estimating x is

$$\min \ \text{loss}(x) + \lambda \times \text{penalty}(x) \tag{1}$$

  - Sparsity via L_1
  - Sparsity via L_1/L_q
  - Sparsity via Fused Lasso
  - Sparse Inverse Covariance Estimation
  - Sparsity via Trace Norm

The L_1 Norm Penalty

$$\boxed{\min \ \text{loss}(x) + \lambda \, ||x||_0}$$

$$\boxed{\min \ \text{loss}(x) + \lambda \, ||x||_1}$$

The L_1 Norm Penalty

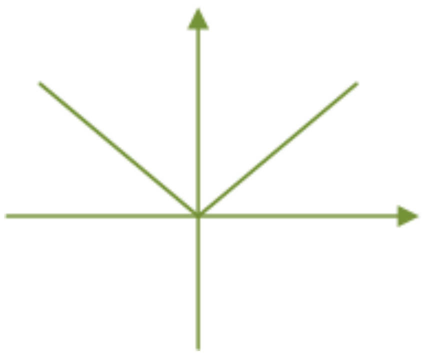
- penalty$(x)=||x||_1=\sum_i|x_i|$
  - Valid norm
  - Convex
  - Computationally tractable
  - Sparsity induced norm
  - Theoretical properties
  - Various Extensions

$$\boxed{\min \ \text{loss}(x) + \lambda \, ||x||_0}$$

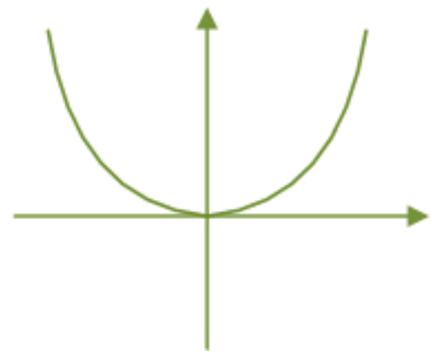
$$\boxed{\min \ \text{loss}(x) + \lambda \, ||x||_1}$$

Why does L_1 Induce Sparsity?

Analysis in 1D (comparison with L_2)



$0.5 \times (x-v)^2 + \lambda|x|$



$0.5 \times (x-v)^2 + \lambda x^2$

If $v \geq \lambda$, $x=v-\lambda$
If $v \leq -\lambda$, $x=v+\lambda$
Else, $x=0$

$x=v/(1+2\lambda)$

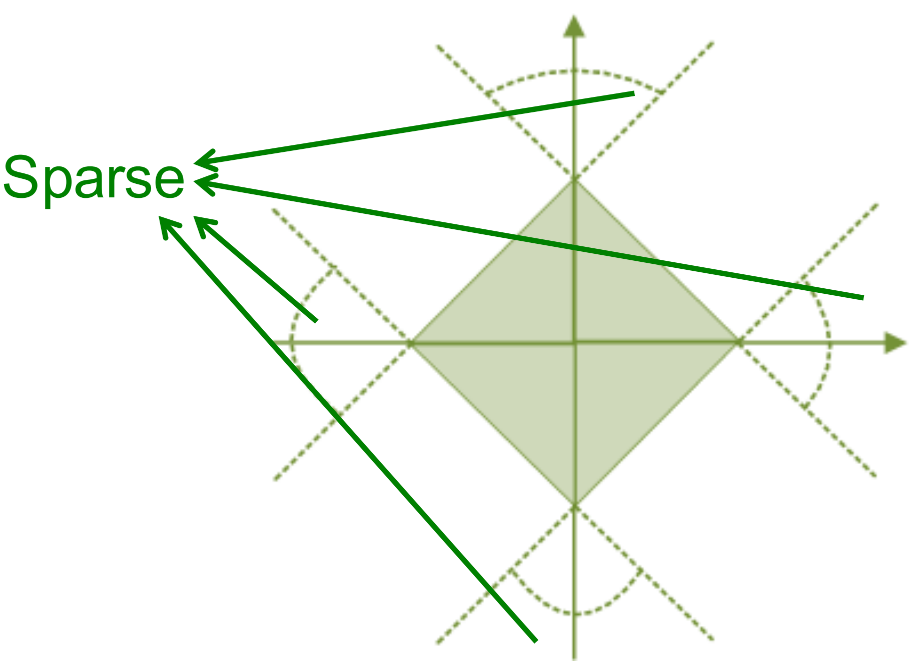
Nondifferentiable at 0

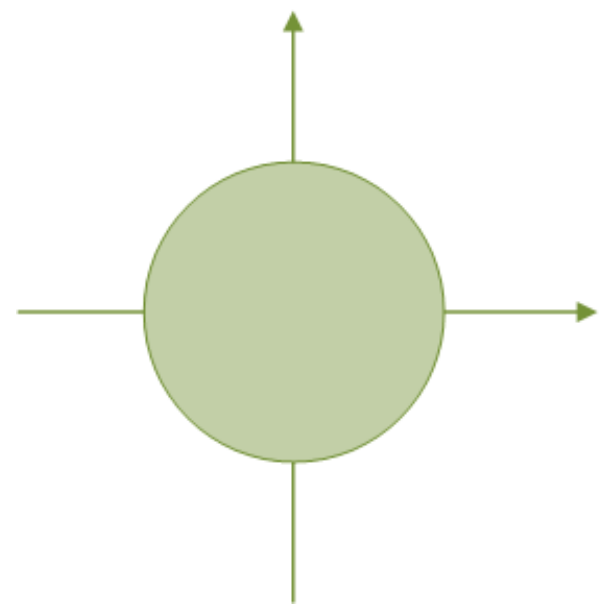
Differentiable at 0

Why does L_1 Induce Sparsity?

- Understanding from the projection

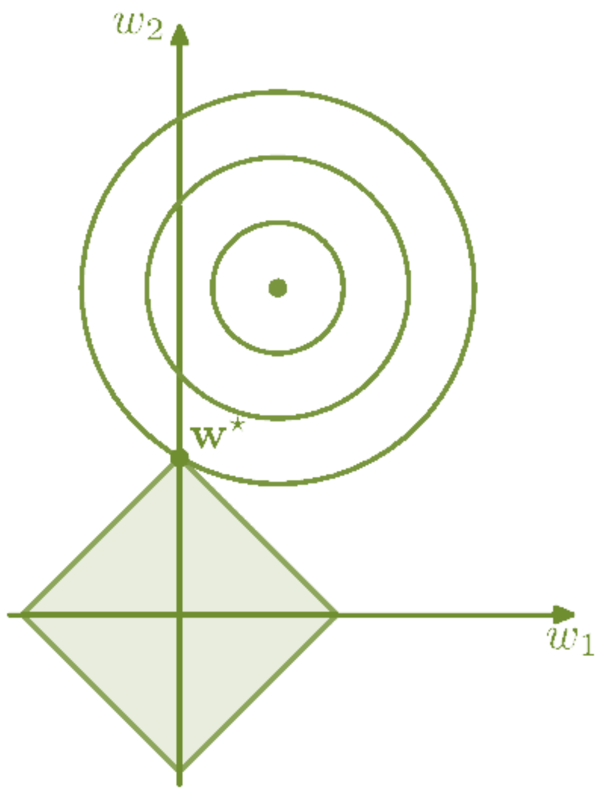
min loss(x)	min $0.5\|\|x-v\|\|^2$	min loss(x)	min $0.5\|\|x-v\|\|^2$
s.t. $\|\|x\|\|_1 \le 1$	s.t. $\|\|x\|\|_1 \le 1$	s.t. $\|\|x\|\|_2 \le 1$	s.t. $\|\|x\|\|_2 \le 1$

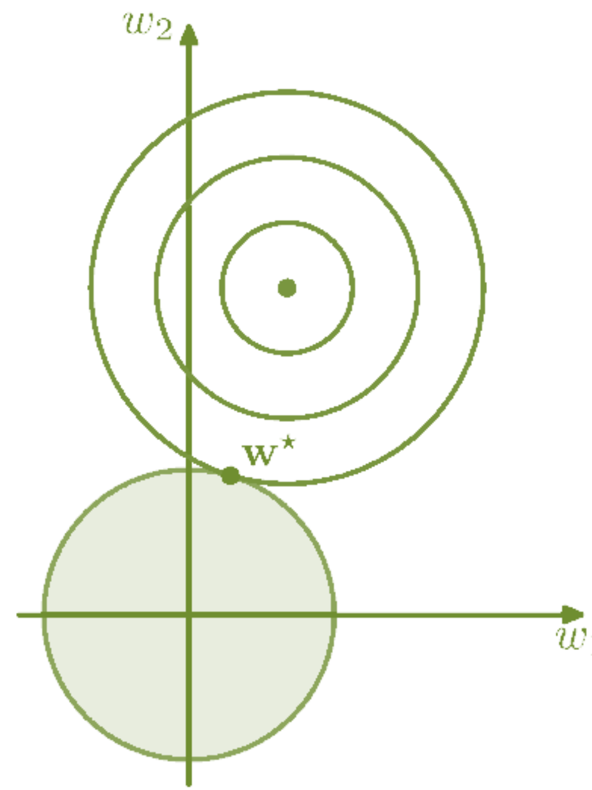




Why does L_1 Induce Sparsity?

- Understanding from constrained optimization

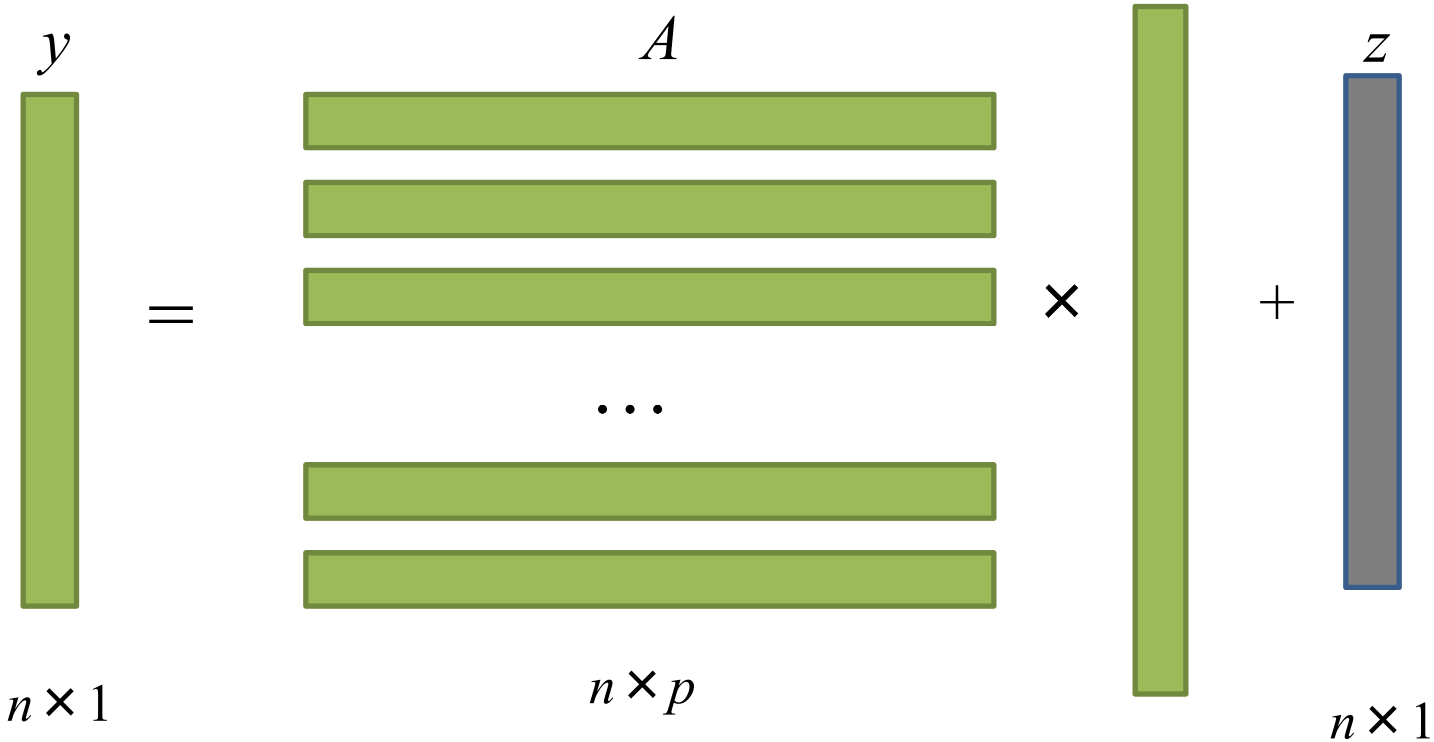




(Bishop, 2006, Hastie et al., 2009)

Lasso (Tibshirani, 1996)

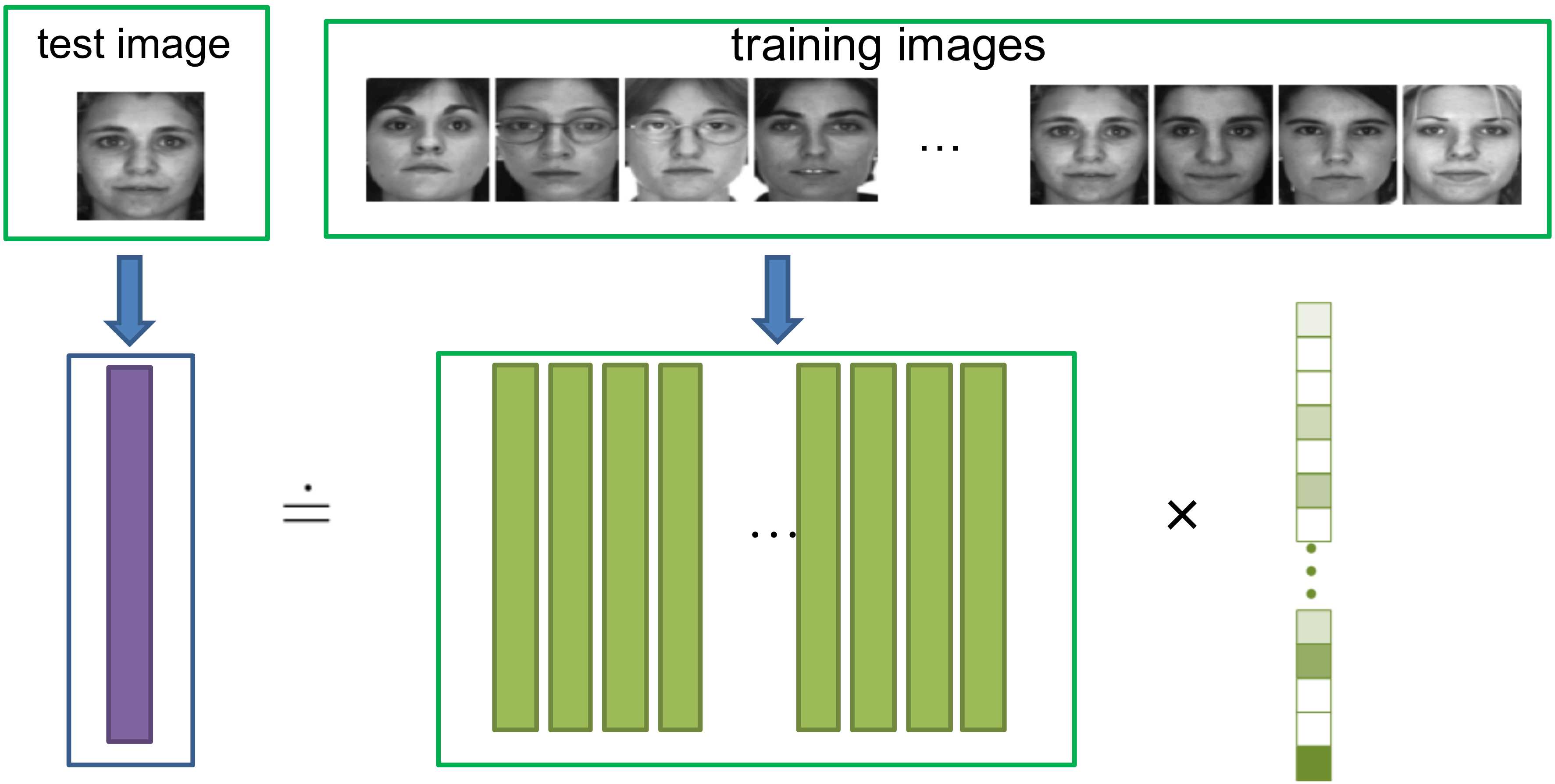
$$\frac{1}{2}\|Ax - y\|_2^2 + \lambda\|x\|_1$$



Simultaneous feature selection and regression

Application: Face Recognition

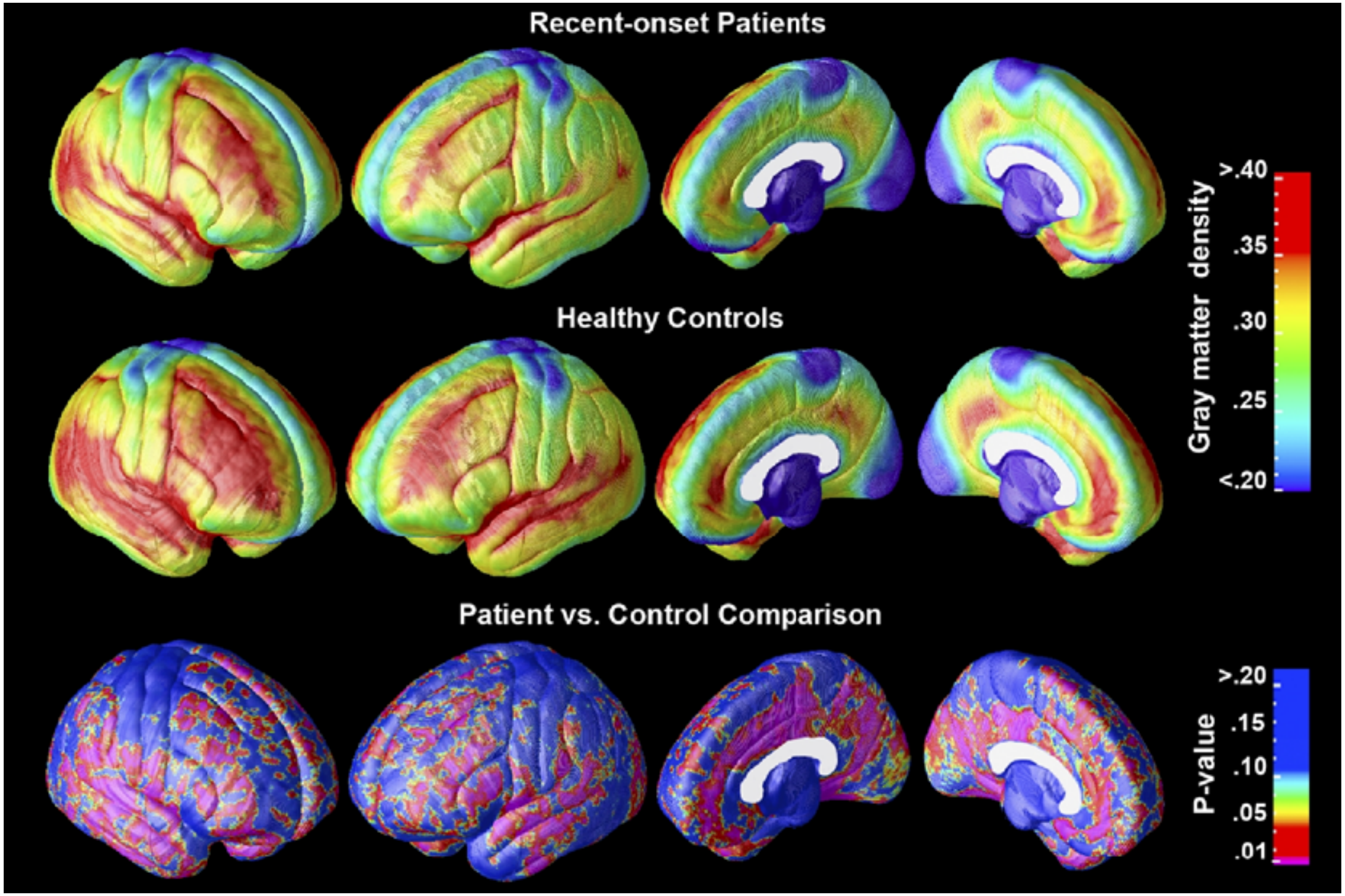
(Wright et al. 2009)



Use the computed sparse coefficients for classification

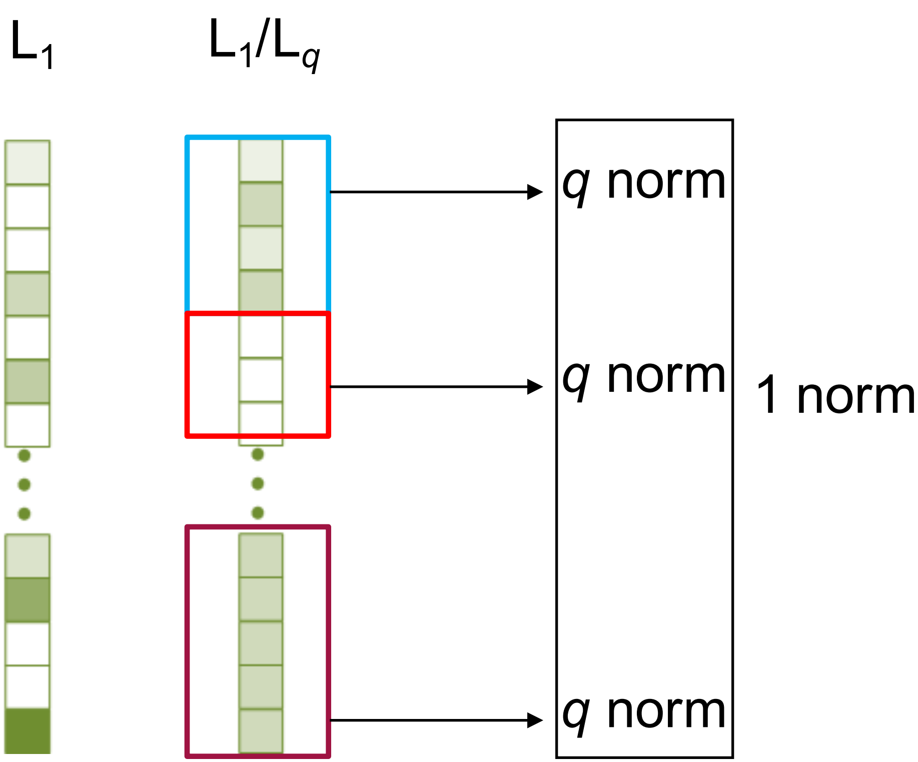
Application: Biomedical Informatics

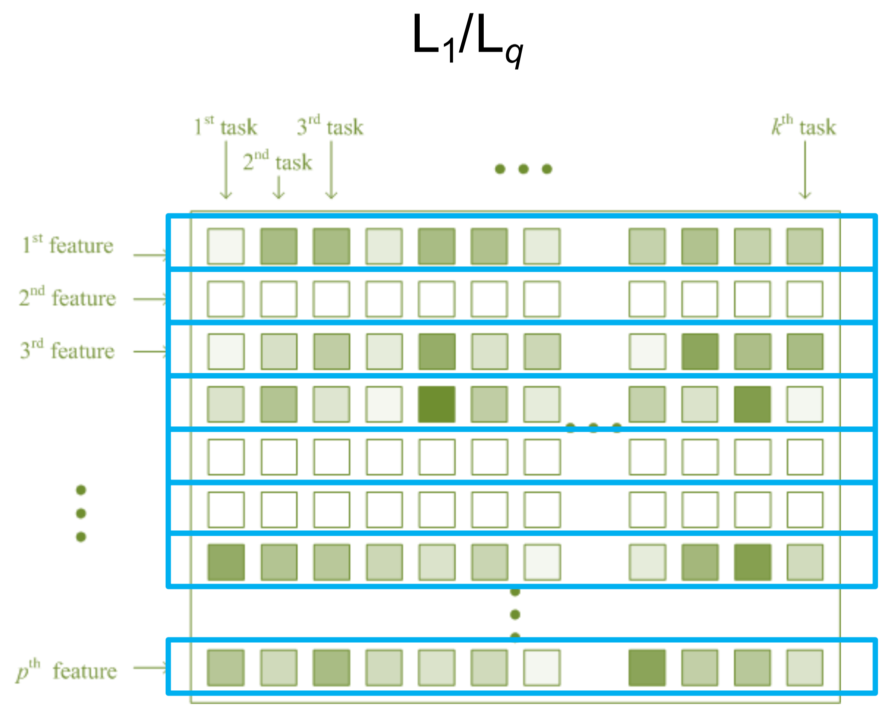
(Sun et al. 2009)



Elucidate a Magnetic Resonance Imaging-Based Neuroanatomic Biomarker for Psychosis

From L_1 to L_1/L_q (q>1)?



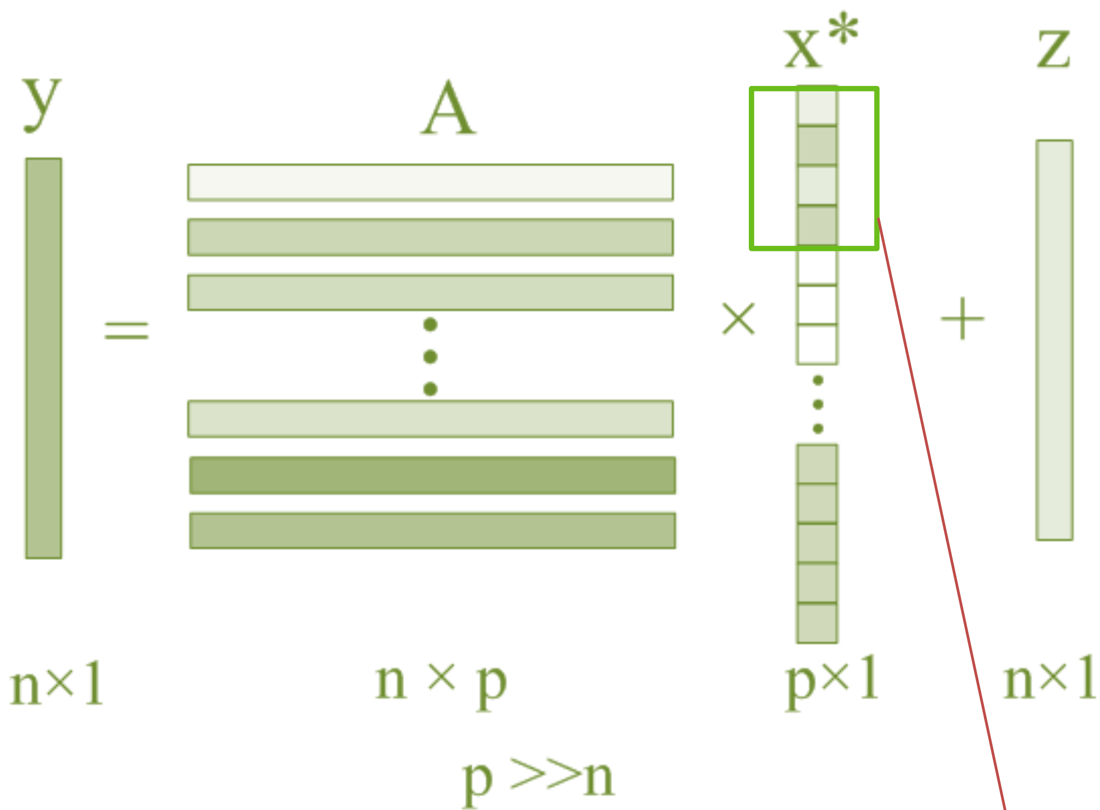


$$\|X\|_{q,1} = \sum_i \|X_{G_i}\|_q$$

Most existing work focus on q=2, ∞

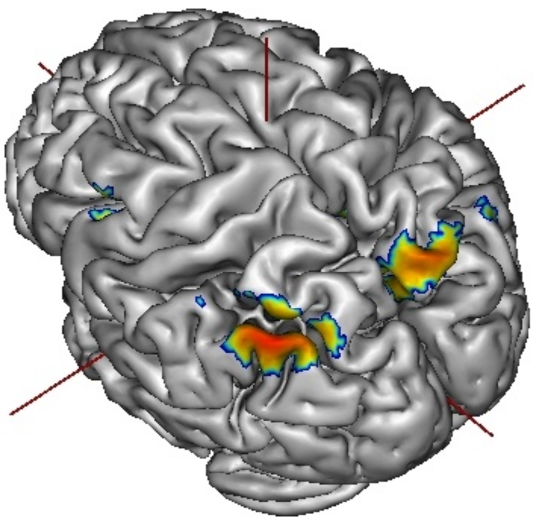
MICHIGAN STATE UNIVERSITY

Group Lasso (Yuan and Lin, 2006)

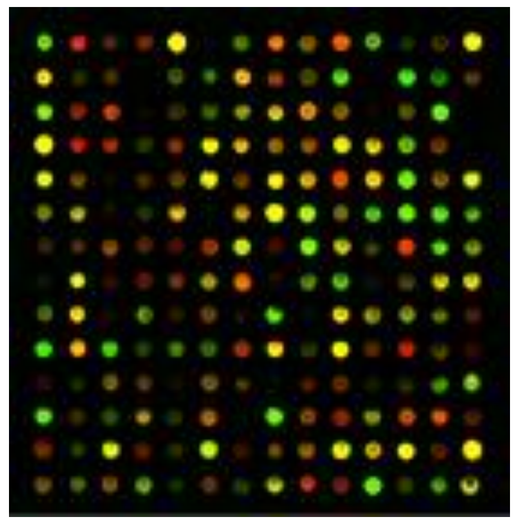


$$\min \frac{1}{2}\|Ax - y\|_2^2 + \lambda \sum_{i=1}^{g} \boxed{d_i \|x_{G_i}\|_2}$$

Group Feature Selection



brain region



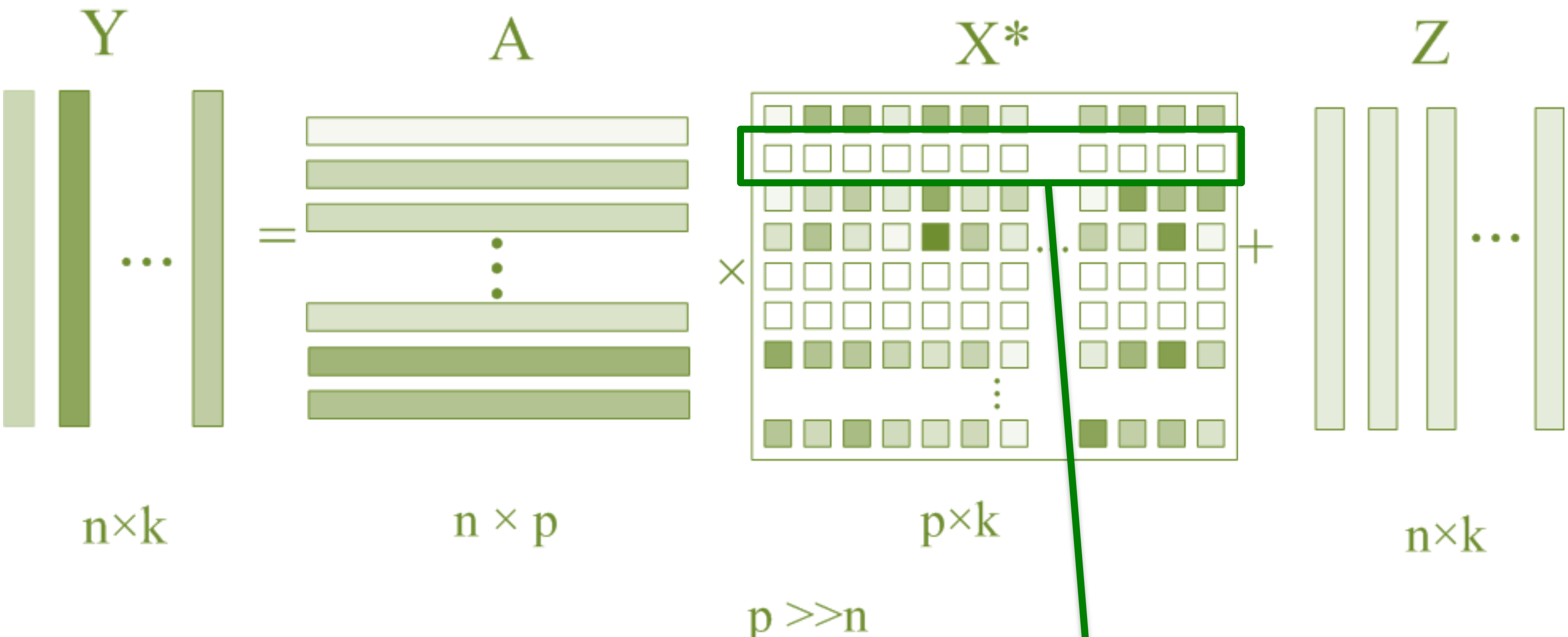
functional group

group

A	**1**	**0**	**0**	**0**
T	0	1	0	0
C	0	0	1	0
G	0	0	0	1

categorical variable

Multi-Task/Class Learning via L_1/L_q



$$\min_X \frac{1}{2}\|AX - Y\|_F^2 + \lambda \sum_{i=1}^{p} \boxed{\|\mathbf{x}_i\|_q}$$

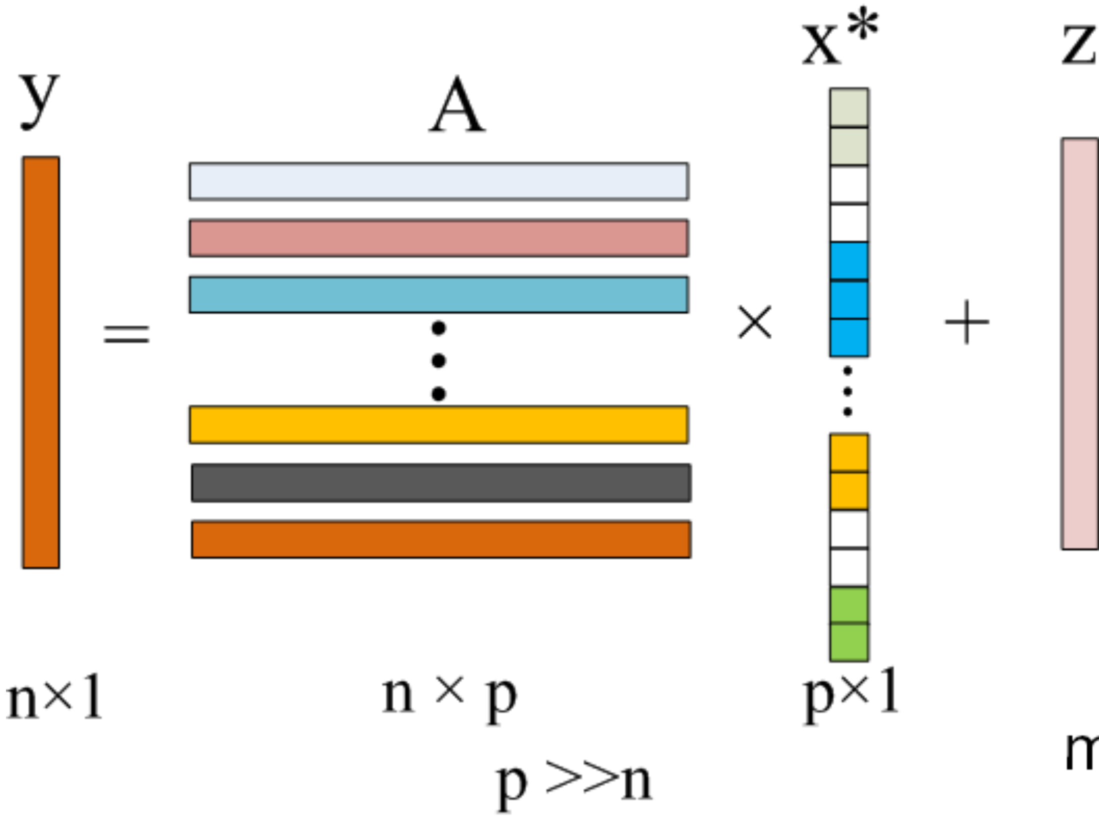
Writer-specific Character Recognition

(Obozinski, Taskar, and Jordan, 2006)

- Letter data set:
  - The letters are from more than 180 different writers
  - It has 8 tasks for discriminating letter c/e, g/y, g/s, m/n, a/g, i,/j, a/o. f/t, and h/n

The letter 'a' written by 40 different people

Fused Lasso



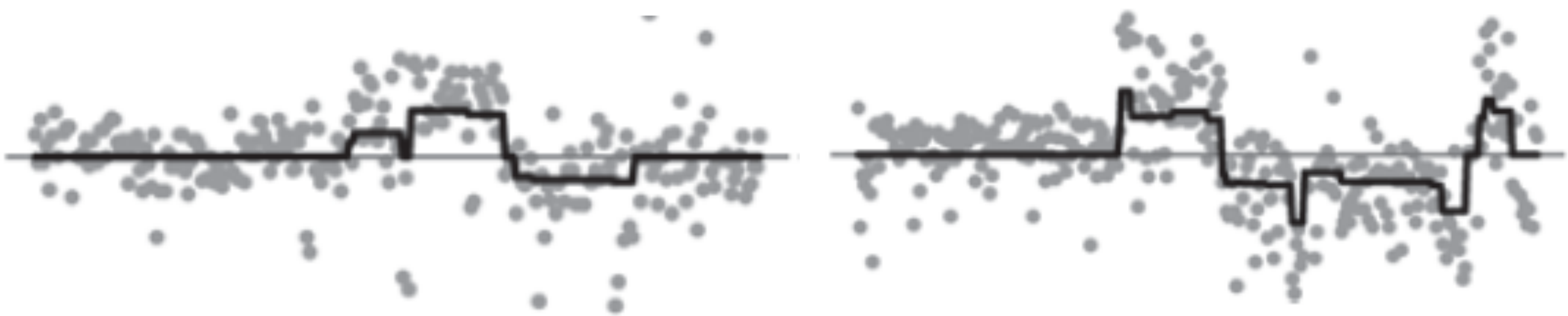
$$\min_x \frac{1}{2}\|Ax - y\|_2^2 + \text{fl}(x)$$

$$\text{fl}(x) = \lambda_1 \sum_{i=1}^{p} |x_i| + \lambda_2 \sum_{i=1}^{p-1} |x_i - x_{i+1}|$$

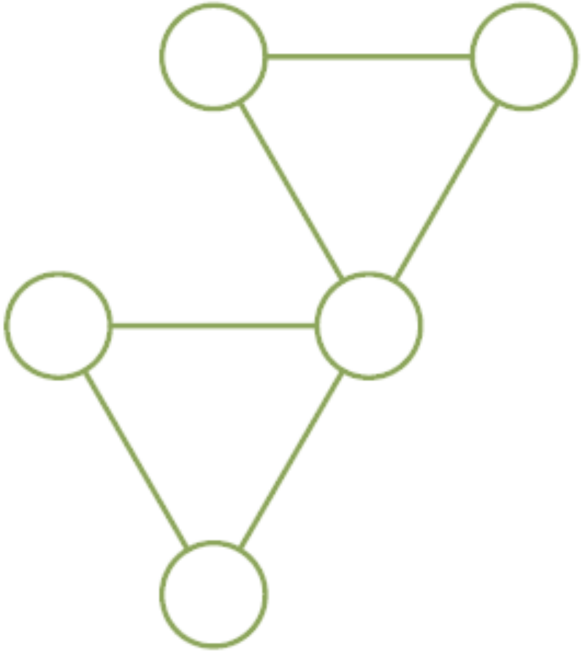
Application: Arracy CGH Data Analysis

(Tibshirani and Wang, 2008)

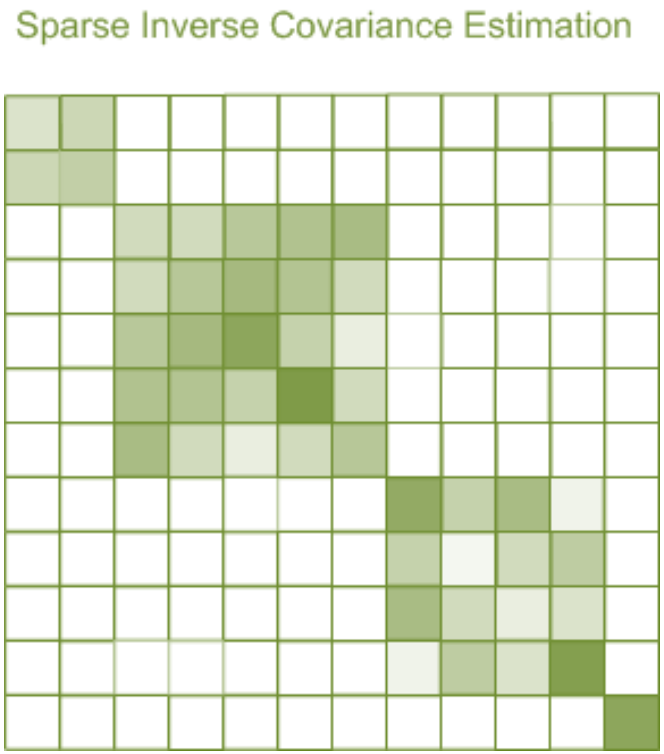
- Comparative genomic hybridization (CGH)
  - Measuring DNA copy numbers of selected genes on the genome
  - In cells with cancer, mutations can cause a gene to be either deleted or amplified
- Array CGH profile of two chromosomes of breast cancer cell line MDA157.



Sparse Inverse Covariance Estimation



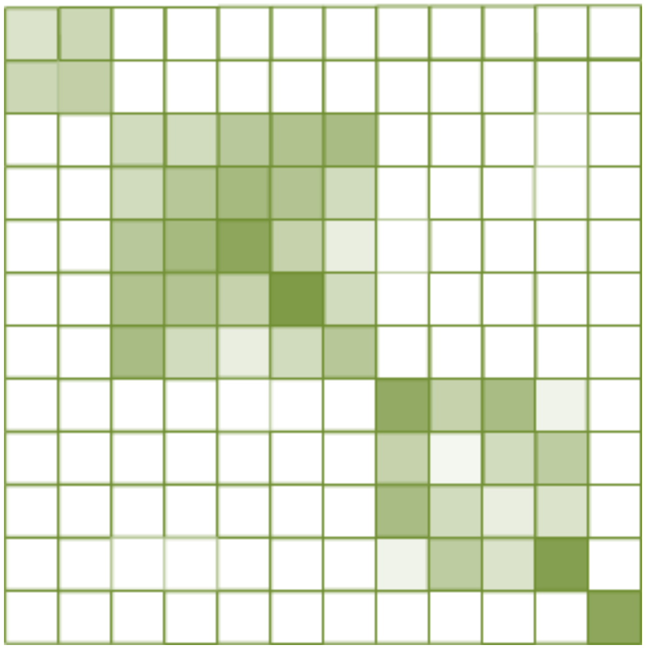
Undirected graphical model
(Markov Random Field)



The pattern of zero entries in the inverse covariance matrix of a multivariate normal distribution corresponds to conditional independence restrictions between variables.

The SICE Model

Sparse Inverse Covariance Estimation



Sparsity

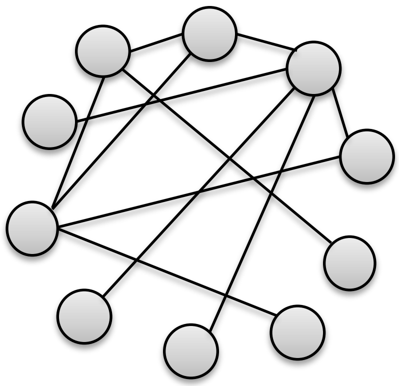
$$\arg\max_{X \succ 0} \log\det X - \underline{\mathrm{trace}(SX)} - \boxed{\lambda\|X\|_1}$$

Log-likelihood

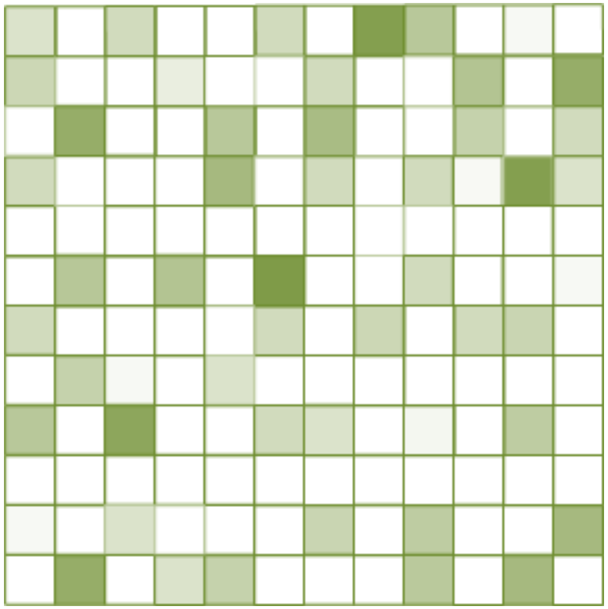
When S is invertible, directly maximizing the likelihood gives

$$X = S^{-1}$$

Network Construction



- Biological network
- Social network
- Brain network



Equivalent matrix representation

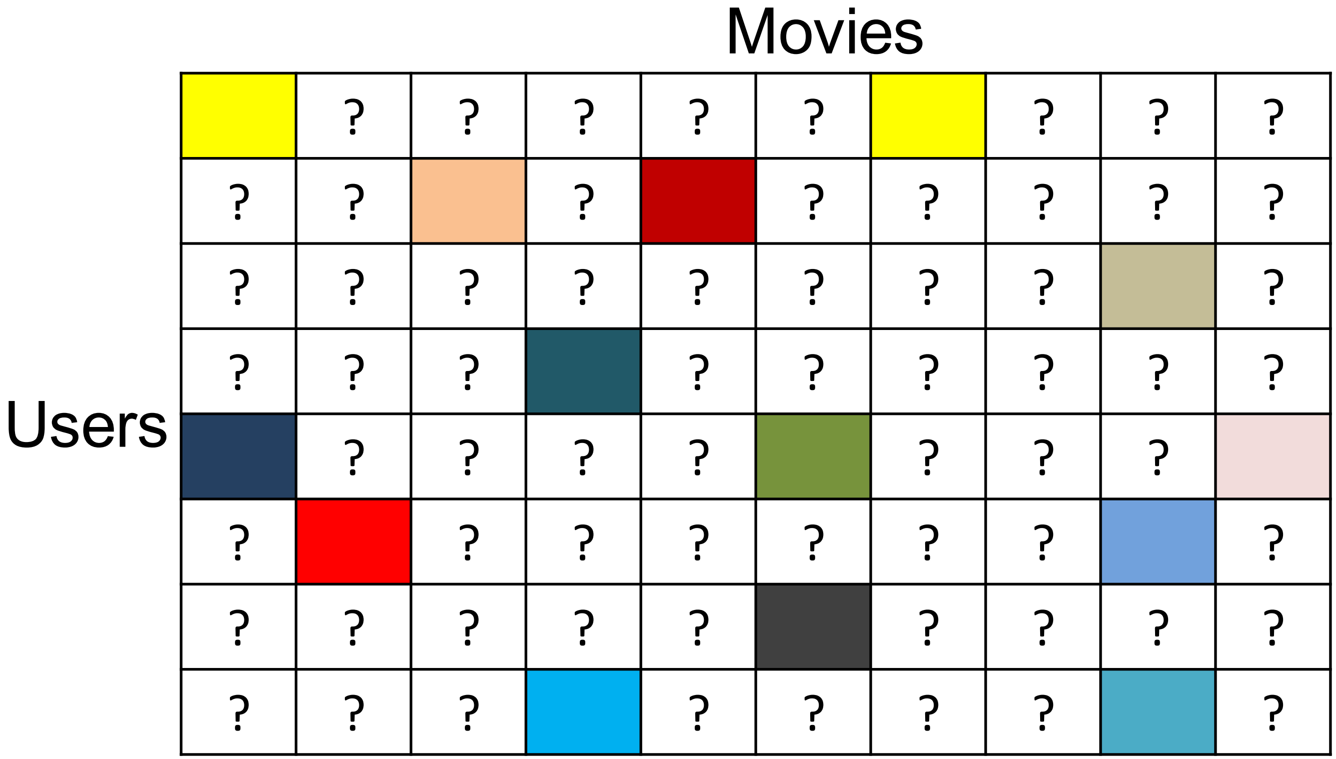
Sparsity: Each node is linked to a small number of neighbors in the network.

Matrix Completion

	?	?	?	?	?		?	?	?
?	?		?		?	?	?	?	?
?	?	?	?	?	?	?	?		?
?	?	?		?	?	?	?	?	?
	?	?	?	?		?	?	?	
?		?	?	?	?	?	?		?
?	?	?	?	?		?	?	?	?
?	?	?		?	?	?	?		?

- Predict the missing values

The Netflix Problem

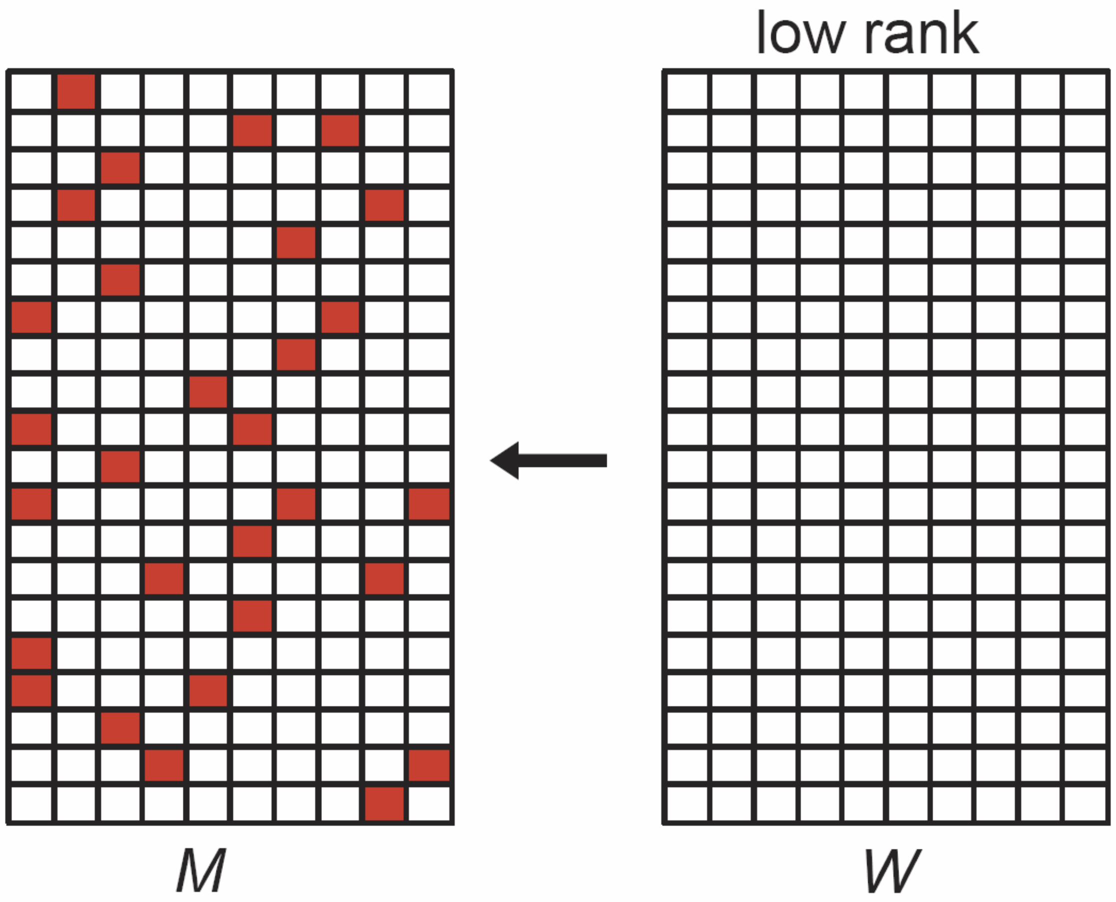


- About a million users and 25,000 movies
- Known ratings are sparsely distributed
- Predict unknown ratings

Preferences of users are determined by a small number of factors → low rank

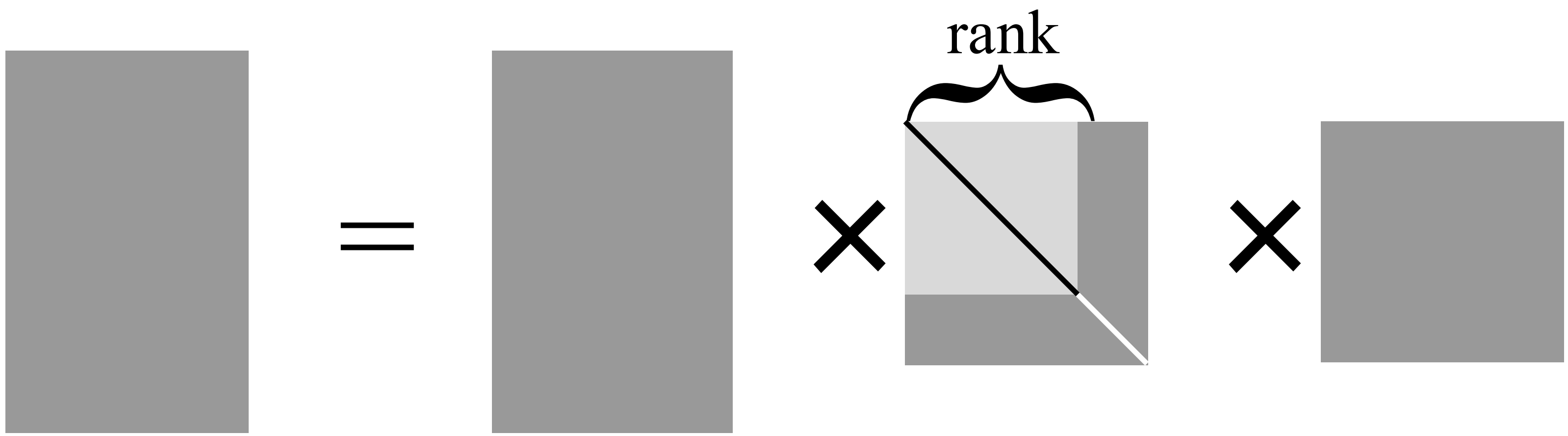
Low Rank Matrix Completion

$$\min_{W} \sum_{i,j \in \text{observed}} \ell(M_{ij}, W_{ij}) + \lambda * \text{rank}(W)$$



Matrix Rank

- The number of independent rows or columns
- The singular value decomposition (SVD):



Optimization

$$\min_{\mathbf{w}\in\mathbb{R}^d} \left\{ f(\mathbf{w}) = l(\mathbf{w}) + r(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} l_i(\mathbf{w}) + r(\mathbf{w}) \right\},$$

Name	Loss function $l_i(\mathbf{w})$
Least Squares	$\frac{1}{2}(y_i - \mathbf{x}_i^T\mathbf{w})^2$
Logistic Regression	$\log(1+\exp(-y_i\mathbf{x}_i^T\mathbf{w}))$
Squared Hinge Loss	$\max(0, 1 - y_i\mathbf{x}_i^T\mathbf{w})^2$

Name	regularizer (penalty) $r(\mathbf{w})$
Lasso [49]	$\lambda\sum_{j=1}^{d}\lvert w_j\rvert$
Fused Lasso [50]	$\lambda_1\sum_{j=1}^{d}\lvert w_j\rvert + \lambda_2\sum_{j=1}^{d-1}\lvert w_j - w_{j+1}\rvert$
Graph Fused Lasso [8]	$\lambda_1\sum_{j=1}^{d}\lvert w_j\rvert + \lambda_2\sum_{(j,k)\in\mathcal{E}}\lvert w_j - w_k\rvert$
Group Lasso [65]	$\lambda\sum_{k=1}^{K}\lVert\mathbf{w}_{\mathcal{G}_k}\rVert$
Sparse Group Lasso [13, 44]	$\lambda_1\sum_{j=1}^{d}\lvert w_j\rvert + \lambda_2\sum_{k=1}^{K}\lVert\mathbf{w}_{\mathcal{G}_k}\rVert$
Tree Lasso [34, 24]	$\sum_{j=1}^{J}\sum_{k=1}^{K_j}\lambda_k^j\lVert\mathbf{w}_{\mathcal{G}_k^j}\rVert$

Gradient Descent for the Composite Model

(Nesterov, 2007; Beck and Teboulle, 2009)

min $f(x) = \text{loss}(x) + \lambda \times \text{penalty}(x)$

Model

$$\mathcal{M}(x_i, \gamma_i) = \underbrace{[\text{loss}(x_i) + \langle \text{loss}'(x_i), x - x_i \rangle]}_{\text{1st order Taylor expansion}} + \frac{1}{2\gamma_i} \underbrace{\|x - x_i\|_2^2}_{\text{Regularization}} + \lambda \times \underbrace{\text{penalty}(x)}_{\text{Nonsmooth part}}$$

Repeat

$$x_{i+1} = \arg\min \mathcal{M}(x_i, \gamma_i)$$

Until "convergence"

Convergence rate $O(1/N)$

First Order Optimization

$$\begin{aligned}\mathbf{w}^{k+1} &= \arg\min_{\mathbf{w}\in\mathbb{R}^d}\left\{l(\mathbf{s}^k) + \nabla l(\mathbf{s}^k)^T(\mathbf{w}-\mathbf{s}^k) + \frac{1}{2\alpha_k}\|\mathbf{w}-\mathbf{s}^k\|^2 + r(\mathbf{w})\right\}\\ &= \arg\min_{\mathbf{w}\in\mathbb{R}^d}\left\{\frac{1}{2}\left\|\mathbf{w}-(\mathbf{s}^k-\alpha_k\nabla l(\mathbf{s}^k))\right\|^2 + \alpha_k r(\mathbf{w})\right\}\\ &= \mathrm{Prox}^r_{\alpha_k}\left(\mathbf{s}^k - \alpha_k\nabla l(\mathbf{s}^k)\right),\end{aligned}$$

- FISTA, SpaRSA
- How to efficiently solve the proximal operator problem?
- Closed-form solution for L1, L1/L2, analytical form for trace norm

Second Order Optimization

– Compute the descent direction:

$$\Delta \mathbf{w}^k = \arg\min_{\Delta \mathbf{w} \in \mathbb{R}^d} \left\{ l(\mathbf{w}^k) + \nabla l(\mathbf{w}^k)^T \Delta \mathbf{w} + \frac{1}{2} \Delta \mathbf{w}^T H^k \Delta \mathbf{w} + r(\mathbf{w}^k + \Delta \mathbf{w}) - r(\mathbf{w}^k) \right\},$$

where H^k is the (approximated) Hessian matrix of $l(\mathbf{w})$ at $\mathbf{w} = \mathbf{w}^k$.

– Iterate along the descent direction:

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha_k \Delta \mathbf{w}^k.$$

- How to efficiently solve the above subproblem?
  - Coordinate Descent, FISTA, SpaRSA

Stochastic Optimization

- Randomly pick a sample $i \in \{1, \cdots, n\}$.
- Evaluate the gradient on the i-th sample and generate a sequence $\{\mathbf{w}^k\}$ via

$$
\begin{aligned}
\mathbf{w}^{k+1} &= \arg\min_{\mathbf{w} \in \mathbb{R}^d} \left\{ l(\mathbf{w}^k) + \nabla l_i(\mathbf{w}^k)^T (\mathbf{w} - \mathbf{w}^k) + \frac{1}{2\alpha_k} \|\mathbf{w} - \mathbf{w}^k\|^2 + r(\mathbf{w}) \right\} \\
&= \arg\min_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{2} \left\| \mathbf{w} - (\mathbf{w}^k - \alpha_k \nabla l_i(\mathbf{w}^k)) \right\|^2 + \alpha_k r(\mathbf{w}) \right\} \\
&= \mathrm{Prox}_{\alpha_k}^r \left(\mathbf{w}^k - \alpha_k \nabla l_i(\mathbf{w}^k) \right).
\end{aligned}
$$

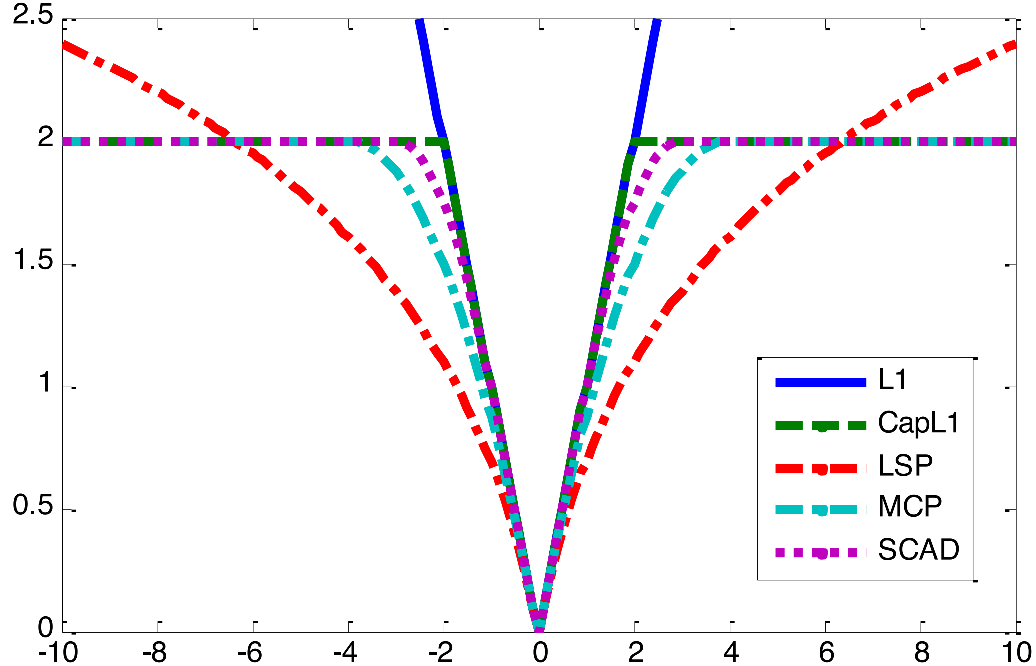
Road Map

- Introduction to Sparsity
- Convex Approaches
- **Non-Convex Approaches**
- Topic: Matrix Completion
- Topic: Multi-task Learning

Non-convex Sparse Models

$$\min_{\mathbf{w}\in\mathbb{R}^d} \{f(\mathbf{w}) = l(\mathbf{w}) + \lambda r(\mathbf{w})\}$$

$l(\mathbf{w})$ and $r(\mathbf{w})$ *may not be convex*



Ref. J. Fan (2001, 2012), H. Zou (2008), X. Shen (2012)
T. Zhang (2010,2012), C.H. Zhang (2010)

Different Non-convex Penalties

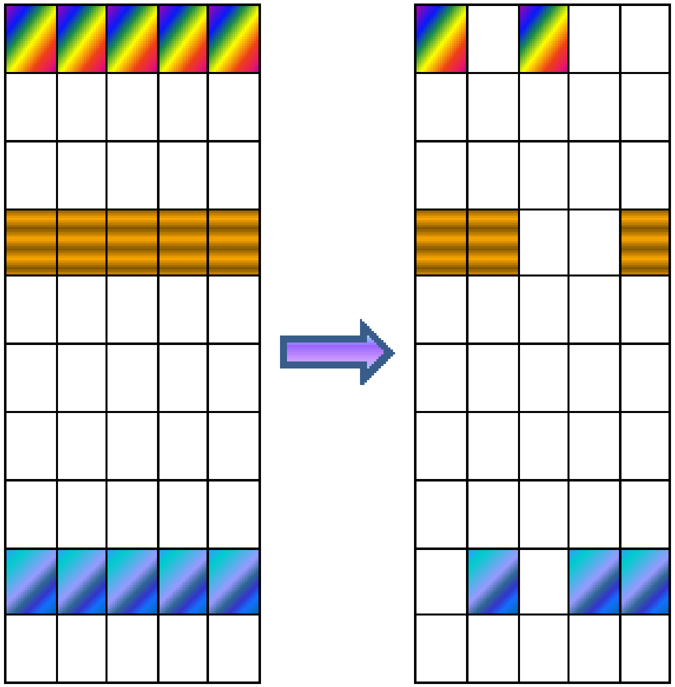
ℓ_1-norm	$\lambda\lvert w_i\rvert$
LSP	$\lambda \log(1+\lvert w_i\rvert/\theta)\ (\theta>0)$
SCAD	$\lambda\int_0^{\lvert w_i\rvert}\min\left(1,\frac{[\theta\lambda-x]_+}{(\theta-1)\lambda}\right)dx\ (\theta>2)$ $=\begin{cases}\lambda\lvert w_i\rvert, & \text{if } \lvert w_i\rvert\le\lambda,\\ \frac{-w_i^2+2\theta\lambda\lvert w_i\rvert-\lambda^2}{2(\theta-1)}, & \text{if } \lambda<\lvert w_i\rvert\le\theta\lambda,\\ (\theta+1)\lambda^2/2, & \text{if } \lvert w_i\rvert>\theta\lambda.\end{cases}$
MCP	$\lambda\int_0^{\lvert w_i\rvert}\left[1-\frac{x}{\theta\lambda}\right]_+dx\ (\theta>0)$ $=\begin{cases}\lambda\lvert w_i\rvert-w_i^2/(2\theta), & \text{if } \lvert w_i\rvert\le\theta\lambda,\\ \theta\lambda^2/2, & \text{if } \lvert w_i\rvert>\theta\lambda.\end{cases}$
Capped ℓ_1	$\lambda\min(\lvert w_i\rvert,\theta)\ (\theta>0)$

Non-convex Models: Advantages

- Better approximation of L_0-norm: reduce over-penalization
- Theoretical advantages of non-convex sparse learning models over the convex ones
  - Unbiased feature selection
  - Weak conditions to achieve oracle properties
  - Sharp parameter estimation bound
- Computational Challenges

Ref. J. Fan (2001, 2012), H. Zou (2008), X. Shen (2012)
T. Zhang (2010,2012), C.H. Zhang (2010)

Example: Non-convex MTL Model



$$\min_{W \in \mathbb{R}^{d \times m}} \{ l(W) + r(W) \}$$

$$r(W) = \lambda \sum_{j=1}^{d} \min \left(\|\mathbf{w}^j\|_1, \theta \right)$$ Non-convex

I Joint feature selection

II Shared features + Task specific Features

Pinghua Gong, Jieping Ye, Changshui Zhang. Multi-Stage Multi-Task Feature Learning. NIPS 2012.

Optimization Algorithm

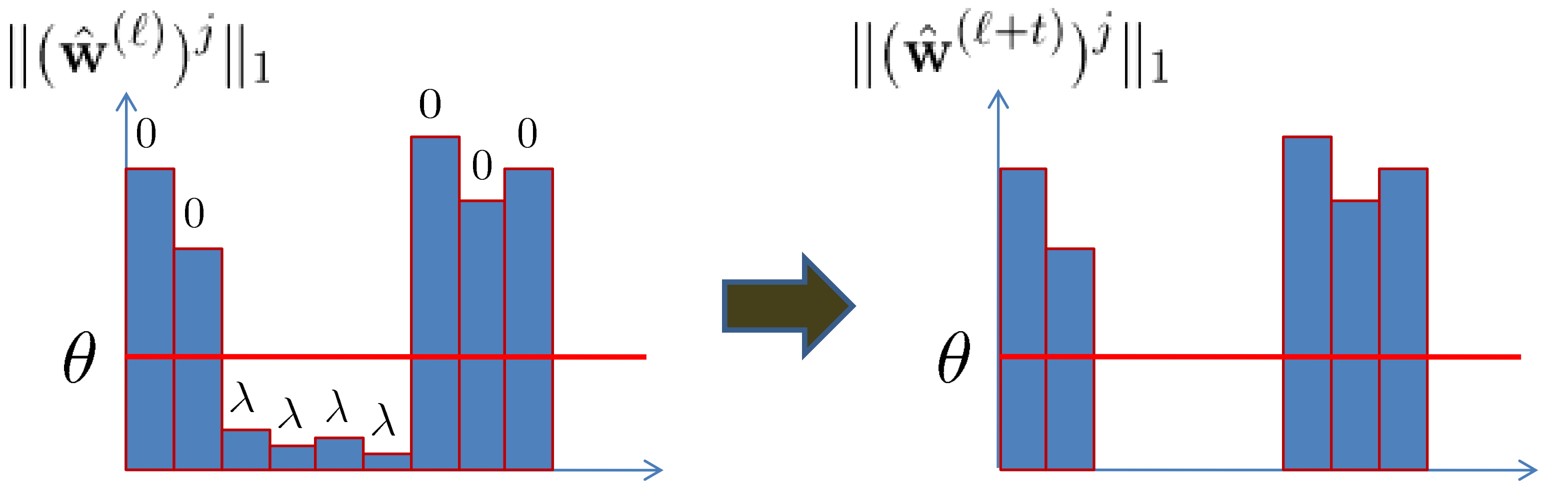
MSMTFL: Multi-Stage Multi-Task Feature Learning

1. Initialize $\lambda_j^{(0)} = \lambda$

repeat

2. $\hat{W}^{(\ell)} = \arg\min_{W \in \mathbb{R}^{d \times m}} \left\{ l(W) + \sum_{j=1}^{d} \lambda_j^{(\ell-1)} \|\mathbf{w}^j\|_1 \right\}$ reweighted Lasso

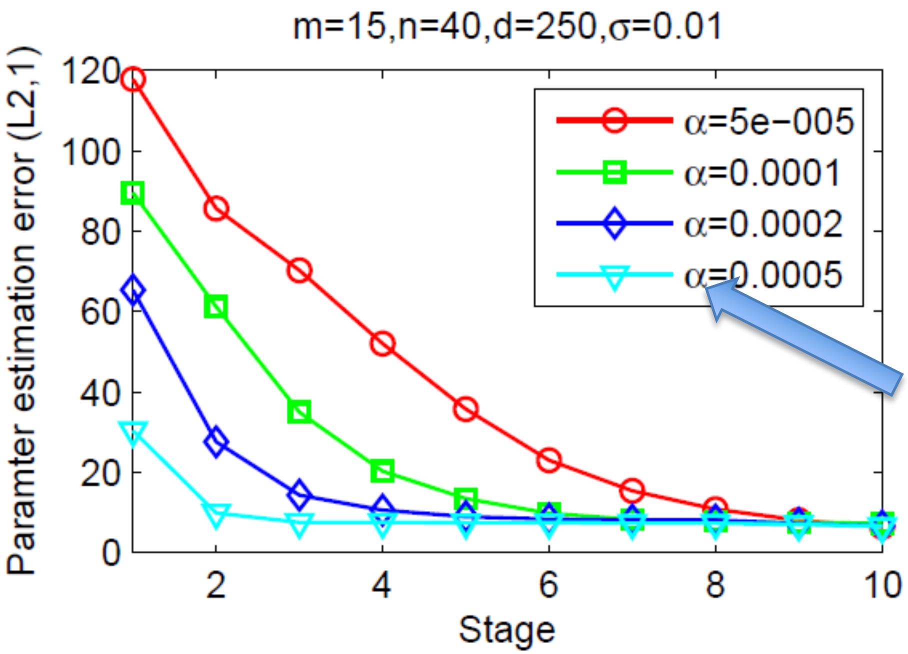
3. $\lambda_j^{(\ell)} = \lambda I(\|(\hat{\mathbf{w}}^{(\ell)})^j\|_1 < \theta) \; (j = 1, \cdots, d)$ penalize small rows



Parameter Estimation Error Bound

$$\|\hat{W}^{(\ell)} - \bar{W}\|_{2,1} = \boxed{0.8^{\ell/2} O\left(m\sqrt{\bar{r}\ln(dm/\eta)/n}\right)} + O\left(m\sqrt{\bar{r}/n + \ln(1/\eta)/n}\right)$$

Exponential shrinkage & stage-wise Improvement



$$\lambda = \alpha\sqrt{\ln(dm)/n},$$

Lasso: $$\|\hat{W}^{Lasso} - \bar{W}\|_{2,1} = O\left(m\sqrt{\bar{r}\ln(dm/\eta)/n}\right)$$

MSMTFL: $$\|\hat{W}^{(\ell)} - \bar{W}\|_{2,1} = O\left(m\sqrt{\bar{r}/n + \ln(1/\eta)/n}\right)$$

A General Solver

- Difference of Convex Programming

$$\min_{\mathbf{w}\in\mathbb{R}^d} \{f(\mathbf{w}) = f_1(\mathbf{w}) - f_2(\mathbf{w})\} \quad f_1(\mathbf{w}), f_2(\mathbf{w}) \text{ are convex}$$

Convex Sub-problem

$$\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}\in\mathbb{R}^d} f_1(\mathbf{w}) - f_2(\mathbf{w}^{(k)}) - \langle \mathbf{s}_2(\mathbf{w}^{(k)}), \mathbf{w} - \mathbf{w}^{(k)} \rangle$$

$\mathbf{s}_2(\mathbf{w}^{(k)})$: sub-gradient of $f_2(\mathbf{w})$ at $\mathbf{w} = \mathbf{w}^{(k)}$

Multiple times of solving convex sub-problems!!

The convex sub-problem usually doesn't have a closed-form solution!!

GIST: General Iterative Shrinkage and Thresholding for Non-convex Problems

$$\min_{\mathbf{w}\in\mathbb{R}^d} \{f(\mathbf{w}) = l(\mathbf{w}) + \lambda r(\mathbf{w})\}$$

$$\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}} \; l(\mathbf{w}^{(k)}) + \langle \nabla l(\mathbf{w}^{(k)}), \mathbf{w} - \mathbf{w}^{(k)} \rangle + \frac{t^{(k)}}{2}\|\mathbf{w} - \mathbf{w}^{(k)}\|^2 + \lambda r(\mathbf{w})$$

$$\mathbf{u}^{(k)} = \mathbf{w}^{(k)} - \nabla l(\mathbf{w}^{(k)})/t^{(k)}$$

Proximal Operator

$$\boxed{\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}} \; \frac{1}{2}\|\mathbf{w} - \mathbf{u}^{(k)}\|^2 + \frac{\lambda}{t^{(k)}} r(\mathbf{w})}$$

Closed-form solution: Capped L1, LSP, SCAD, MCP

Non-convex

Pinghua Gong, Jieping Ye, Changshui Zhang. A General Iterative and Shrinkage Thresholding Algorithm for Non-convex Regularized Problems. ICML 2013.

Step Size Selection

- Initialization: Barzilai-Borwein (BB) rule

$$\mathbf{x}^{(k)} = \mathbf{w}^{(k)} - \mathbf{w}^{(k-1)} \qquad \mathbf{y}^{(k)} = \nabla l(\mathbf{w}^{(k)}) - \nabla l(\mathbf{w}^{(k-1)})$$

$$t^{(k)} = \arg\min_t \|t\mathbf{x}^{(k)} - \mathbf{y}^{(k)}\|^2 = \frac{\langle \mathbf{x}^{(k)}, \mathbf{y}^{(k)} \rangle}{\langle \mathbf{x}^{(k)}, \mathbf{x}^{(k)} \rangle}$$

- Line Search: Monotone & Non-monotone

$$f(\mathbf{w}^{(k+1)}) \leq \max_{i=\max(0,k-m+1),\cdots,k} f(\mathbf{w}^{(i)}) - \frac{\sigma}{2} t^{(k)} \|\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}\|^2$$

Where $\sigma \in (0, 1)$ is a constant

m=1: Monotone*;* m>1: Non-monotone

Assumptions

$$\min_{\mathbf{w}\in\mathbb{R}^d} \{f(\mathbf{w}) = l(\mathbf{w}) + \lambda r(\mathbf{w})\}$$

- A1: $l(\mathbf{w})$ is continuously differentiable with Lipschitz continuous gradient

- A2: $r(\mathbf{w})$ is a continuous function with difference of two convex functions:

$$r(\mathbf{w}) = r_1(\mathbf{w}) - r_2(\mathbf{w})$$

- A3: $f(\mathrm{w})$ is bounded from below

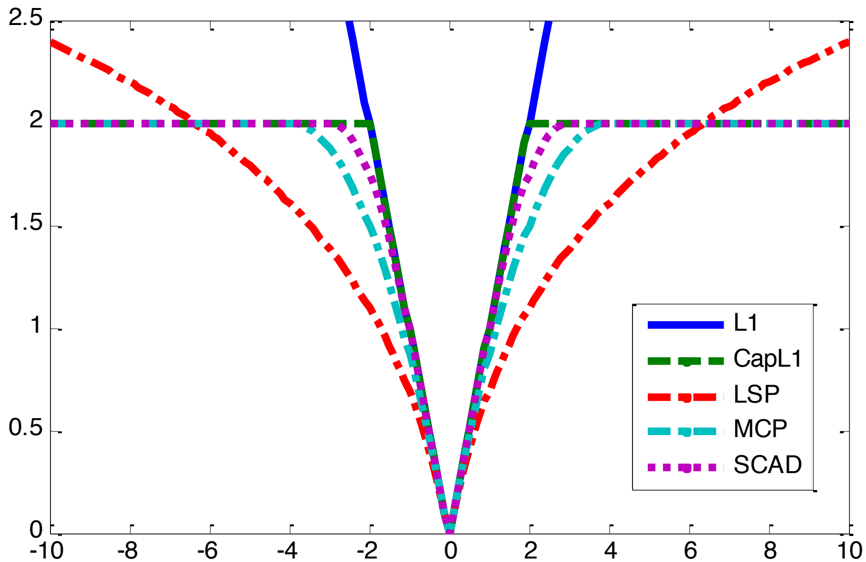
Example

Least Squares: $$l(\mathbf{w}) = \frac{1}{2n}\|X\mathbf{w} - \mathbf{y}\|^2$$

Logistic Regression: $$l(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + \exp(-y_i \mathbf{x}_i^T \mathbf{w})\right)$$

Squared Hinge Loss: $$l(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n} \max\left(0, 1 - y_i \mathbf{x}_i^T \mathbf{w}\right)^2$$

Non-convex Regularizer



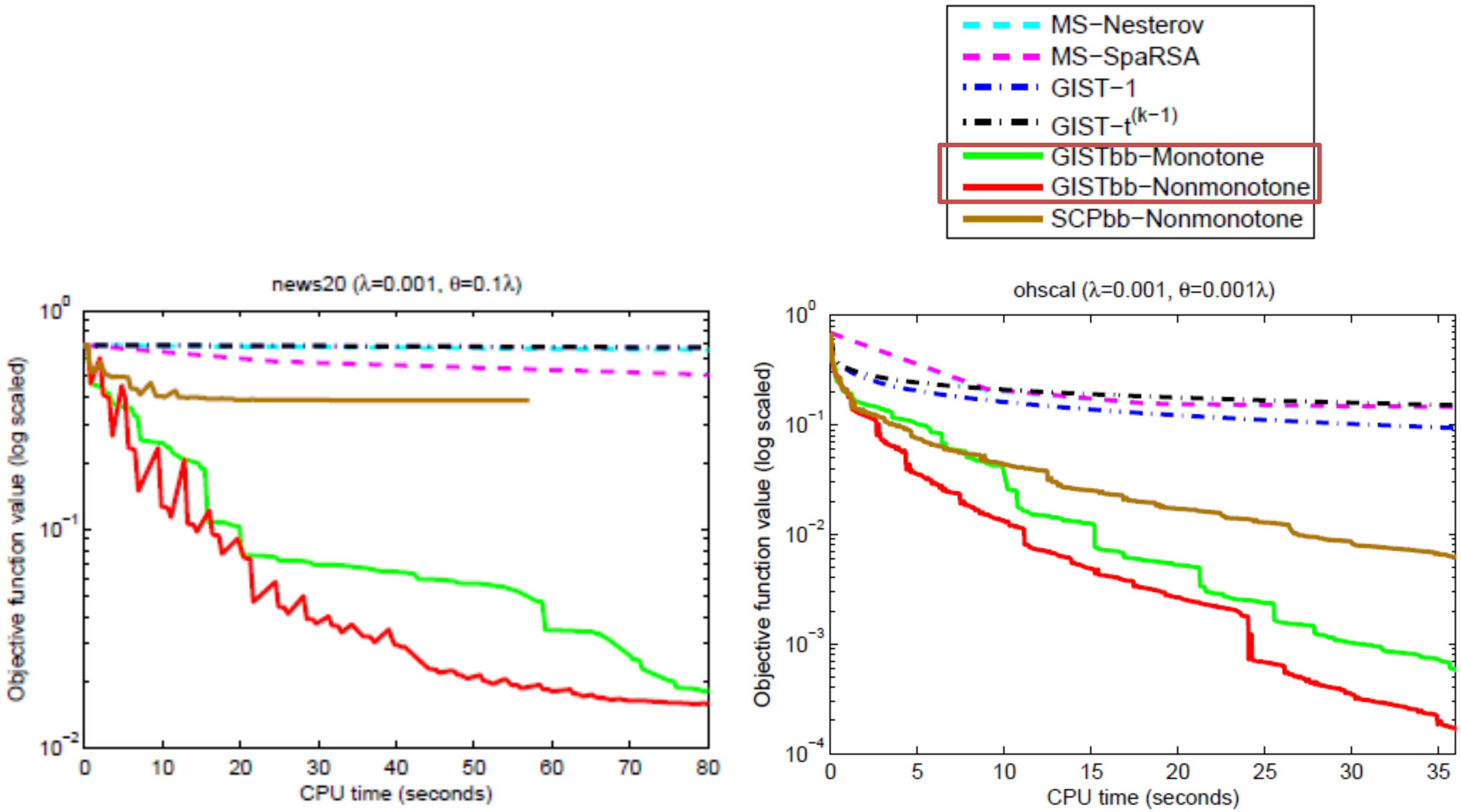
Convergence Analysis

Theorem 1: *Let the assumptions A1-A3 hold and the monotone/Non-monotone line search criterion in be satisfied. Then all limit points of the sequence* $\left\{\mathbf{w}^{(k)}\right\}$ *generated by GIST are* ***critical points***.

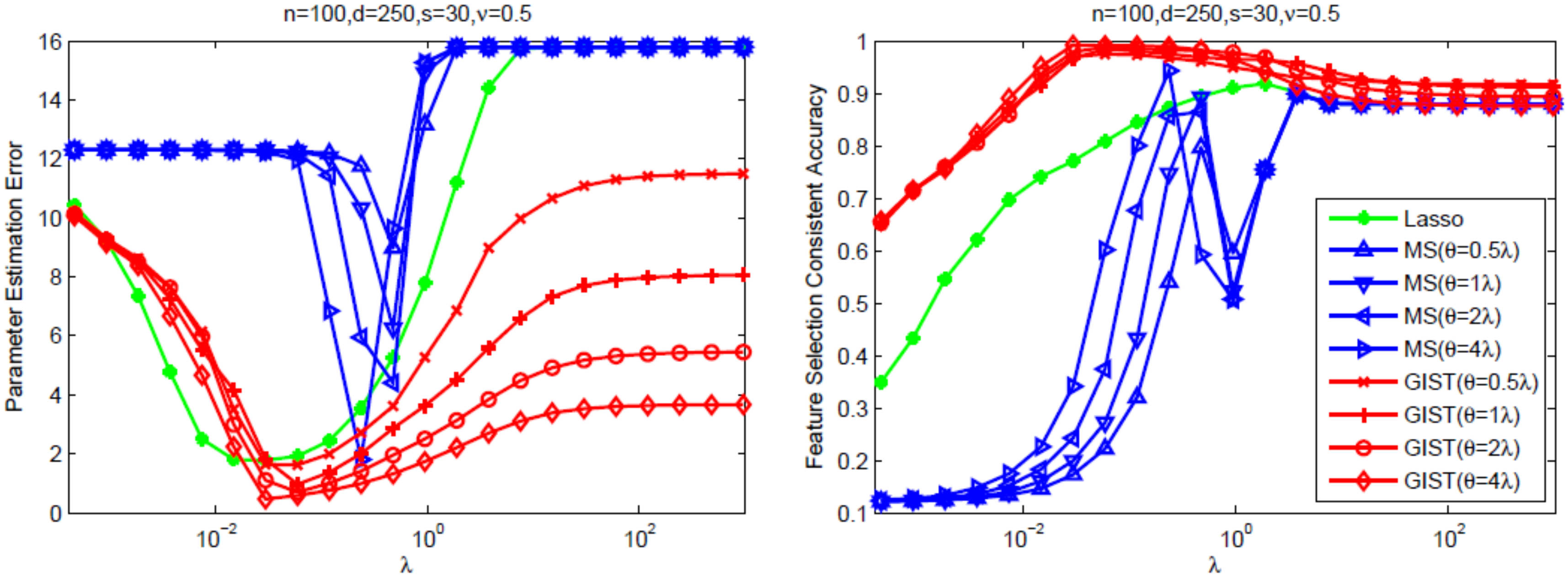
Theorem 2: *Let the assumptions A1-A4 hold and the monotone/non-monotone line search criterion be satisfied. Then the sequence* $\left\{\mathbf{w}^{(k)}\right\}$ *generated by GIST has* ***at least one limit point***.

$$A4: f(\mathbf{w}) \rightarrow +\infty \text{ when } \|\mathbf{w}\| \rightarrow +\infty$$

Evaluation: Convergence



Evaluation: Recovery Performance



Software: GIST

GIST: A Non-Convex Sparse Learning Package

- Loss functions:
    - The least squares loss
    - The logistic loss
    - The squared hinge loss (L2 SVM loss)

- Non-convex Regularizers:
    - LSP
    - SCAD
    - MCP
    - Capped L1

Proximal Alternating Linearized Minimization (PALM) [Bolte et. al. 2013]

Let $\mathbf{w} = (\mathbf{u},\mathbf{v}), l(\mathbf{w}) = l(\mathbf{u},\mathbf{v}), r(\mathbf{w}) = r_1(\mathbf{u}) + r_2(\mathbf{v})$

$\min_{\mathbf{w}} \{l(\mathbf{w}) + r(\mathbf{w})\} \Leftrightarrow \min_{\mathbf{u},\mathbf{v}} \{f(\mathbf{u},\mathbf{v}) = l(\mathbf{u},\mathbf{v}) + r_1(\mathbf{u}) + r_2(\mathbf{v})\}$

- Fix $\mathbf{u} = \mathbf{u}^k$ and conduct a proximal gradient descent with respect to $\mathbf{v}$:

$$\begin{aligned}
\mathbf{v}^{k+1} &= \arg\min_{\mathbf{v}} \left\{ l(\mathbf{u}^k, \mathbf{v}^k) + \nabla_{\mathbf{v}} l(\mathbf{u}^k, \mathbf{v}^k)^T (\mathbf{v} - \mathbf{v}^k) + \frac{1}{2\alpha_k} \|\mathbf{v} - \mathbf{v}^k\|^2 + r_2(\mathbf{v}) \right\} \\
&= \arg\min_{\mathbf{v}} \left\{ \frac{1}{2} \left\| \mathbf{v} - (\mathbf{v}^k - \alpha_k \nabla_{\mathbf{v}} l(\mathbf{u}^k, \mathbf{v}^k)) \right\|^2 + \alpha_k r_2(\mathbf{v}) \right\} \\
&= \mathrm{Prox}_{\alpha_k}^{r_2} \left(\mathbf{v}^k - \alpha_k \nabla_{\mathbf{v}} l(\mathbf{u}^k, \mathbf{v}^k) \right).
\end{aligned}$$

- Fix $\mathbf{v} = \mathbf{v}^{k+1}$ and conduct a proximal gradient descent with respect to $\mathbf{u}$:

$$\begin{aligned}
\mathbf{u}^{k+1} &= \arg\min_{\mathbf{u}} \left\{ l(\mathbf{u}^k, \mathbf{v}^{k+1}) + \nabla_{\mathbf{u}} l(\mathbf{u}^k, \mathbf{v}^{k+1})^T (\mathbf{u} - \mathbf{u}^k) + \frac{1}{2\beta_k} \|\mathbf{u} - \mathbf{u}^k\|^2 + r_1(\mathbf{u}) \right\} \\
&= \arg\min_{\mathbf{u}} \left\{ \frac{1}{2} \left\| \mathbf{u} - (\mathbf{u}^k - \beta_k \nabla_{\mathbf{u}} l(\mathbf{u}^k, \mathbf{v}^{k+1})) \right\|^2 + \beta_k r_1(\mathbf{u}) \right\} \\
&= \mathrm{Prox}_{\beta_k}^{r_1} \left(\mathbf{u}^k - \beta_k \nabla_{\mathbf{u}} l(\mathbf{u}^k, \mathbf{v}^{k+1}) \right).
\end{aligned}$$

Quasi-Newton Method [Rakotomamonjy et. al. 2015]

$$\min_{\mathbf{w}\in\mathbb{R}^n} \{f(\mathbf{w}) = l(\mathbf{w}) + r(\mathbf{w})\}$$

$$l(\mathbf{w}) = \hat{l}(\mathbf{w}) - \tilde{l}(\mathbf{w}) \text{ and } r(\mathbf{w}) = \hat{r}(\mathbf{w}) - \tilde{r}(\mathbf{w})$$

$\hat{l}(\mathbf{w}), \tilde{l}(\mathbf{w}), \hat{r}(\mathbf{w}), \tilde{r}(\mathbf{w})$ are convex functions ($\hat{l}(\mathbf{w})$ and $\tilde{l}(\mathbf{w})$ are differentiable but $\hat{r}(\mathbf{w})$ and $\tilde{r}(\mathbf{w})$ are typically not)

Approximate $\hat{l}(\mathbf{w})$ using the second-order information and approximate $\tilde{l}(\mathbf{w}), \hat{r}(\mathbf{w}), \tilde{r}(\mathbf{w})$ using the first-order information

Quasi-Newton Method [Rakotomamonjy et. al. 2015]

- Compute the descent direction:

$$\Delta \mathbf{w}^k = \arg\min_{\Delta \mathbf{w} \in \mathbb{R}^d} \Big\{ \hat{l}(\mathbf{w}^k) + \nabla \hat{l}(\mathbf{w}^k)^T \Delta \mathbf{w} + \frac{1}{2} \Delta \mathbf{w}^T H^k \Delta \mathbf{w} - \tilde{l}(\mathbf{w}^k) - \nabla \tilde{l}(\mathbf{w}^k)^T \Delta \mathbf{w} + \hat{r}(\mathbf{w}) - \tilde{r}(\mathbf{w}^k) - \tilde{\mathbf{g}}_r(\mathbf{w}^k)^T \Delta \mathbf{w} \Big\},$$

where $\tilde{\mathbf{g}}_r(\mathbf{w}^k)$ is a sub-gradient of $\tilde{r}(\mathbf{w})$ at $\mathbf{w} = \mathbf{w}^k$ and H^k is the (approximated) Hessian of $\hat{l}(\mathbf{w})$ at $\mathbf{w} = \mathbf{w}^k$.

- Iterate along the descent direction:

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \alpha_k \Delta \mathbf{w}^k.$$

- The cost of solving the regularized QP sub-problem is high!
- Avoid solving the QP sub-problem at each iteration (HONOR, 2015).

HONOR: Hybrid Optimization for Non-convex Regularized problems [Gong and Ye, NIPS 2015]

$$\min_{\mathbf{w}\in\mathbb{R}^n} \left\{ f(\mathbf{w}) = l(\mathbf{w}) + r(\mathbf{w}) \right\}$$

A1 : $l(\mathbf{w})$ is coercive, continuously differentiable and $\nabla l(\mathbf{w})$ is Lipschitz continuous with constant L. Moreover, $l(\mathbf{w}) > -\infty, \forall \mathbf{w} \in \mathbb{R}^n$.

A2 : $r(\mathbf{w}) = \sum_{i=1}^{n} \rho(|w_i|)$, where $\rho(t)$ is non-decreasing, continuously differentiable and concave with respect to t in $[0,\infty)$; $\rho(0) = 0$ and $\rho'(0) \neq 0$ with $\rho'(t) = \partial\rho(t)/\partial t$ denoting the derivative of $\rho(t)$ at the point t.

Examples: Non-convex Regularizers

$$\min_{\mathbf{w}\in\mathbb{R}^n} \left\{ f(\mathbf{w}) = l(\mathbf{w}) + r(\mathbf{w}) \right\}$$

$$\text{LSP: } \rho(|w_i|) = \lambda \log(1 + |w_i|/\theta)$$

$$\text{SCAD: } \rho(|w_i|) = \begin{cases} \lambda |w_i|, & \text{if } |w_i| \le \lambda, \\ \dfrac{-w_i^2 + 2\theta\lambda |w_i| - \lambda^2}{2(\theta - 1)}, & \text{if } \lambda < |w_i| \le \theta\lambda, \\ (\theta+1)\lambda^2/2, & \text{if } |w_i| > \theta\lambda. \end{cases}$$

$$\text{MCP: } \rho(|w_i|) = \begin{cases} \lambda |w_i| - w_i^2/(2\theta), & \text{if } |w_i| \le \theta\lambda, \\ \theta\lambda^2/2, & \text{if } |w_i| > \theta\lambda. \end{cases}$$

Mining Second-Order Information

- Obtain a direction using second-order information

$$\mathbf{d}^k = \arg\min_{\mathbf{d}\in\mathbb{R}^n} \left\{ f(\mathbf{w}^k) + \Diamond f(\mathbf{w}^k)^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T B^k \mathbf{d} \right\} = -H^k \Diamond f(\mathbf{w}^k)$$

$$H^k = (B^k)^{-1}, \quad \Diamond_i f(\mathbf{w}) = \begin{cases} \nabla_i l(\mathbf{w}) + \rho'(|w_i|), & \text{if } w_i > 0, \\ \nabla_i l(\mathbf{w}) - \rho'(|w_i|), & \text{if } w_i < 0, \\ \nabla_i l(\mathbf{w}) + \rho'(0), & \text{if } w_i = 0,\ \nabla_i l(\mathbf{w}) + \rho'(0) < 0, \\ \nabla_i l(\mathbf{w}) - \rho'(0), & \text{if } w_i = 0,\ \nabla_i l(\mathbf{w}) - \rho'(0) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

L-BFGS

$$\mathbf{p}^k = \pi(\mathbf{d}^k; \mathbf{v}^k), \text{ where } \mathbf{v}^k = -\Diamond f(\mathbf{w}^k)$$

: projection operation that keeps y and x in the same orthant

HONOR: Hybrid Strategy

- Hybrid Strategy: QN-step or GD-step

$$\mathcal{I}^k = \{i \in \{1,\cdots,n\} : 0 < |w_i^k| \leq \min(\|\mathbf{v}^k\|, \epsilon), w_i^k v_i^k < 0\}$$

Empty → QN-step

Non-empty → GD-step

- QN-step: $\mathbf{w}^k(\alpha) = \pi(\mathbf{w}^k + \alpha \mathbf{p}^k; \mathbf{w}^k)$

  Line search (QN): $f(\mathbf{w}^k(\alpha)) \leq f(\mathbf{w}^k) - \gamma\alpha(\mathbf{v}^k)^T \mathbf{d}^k$

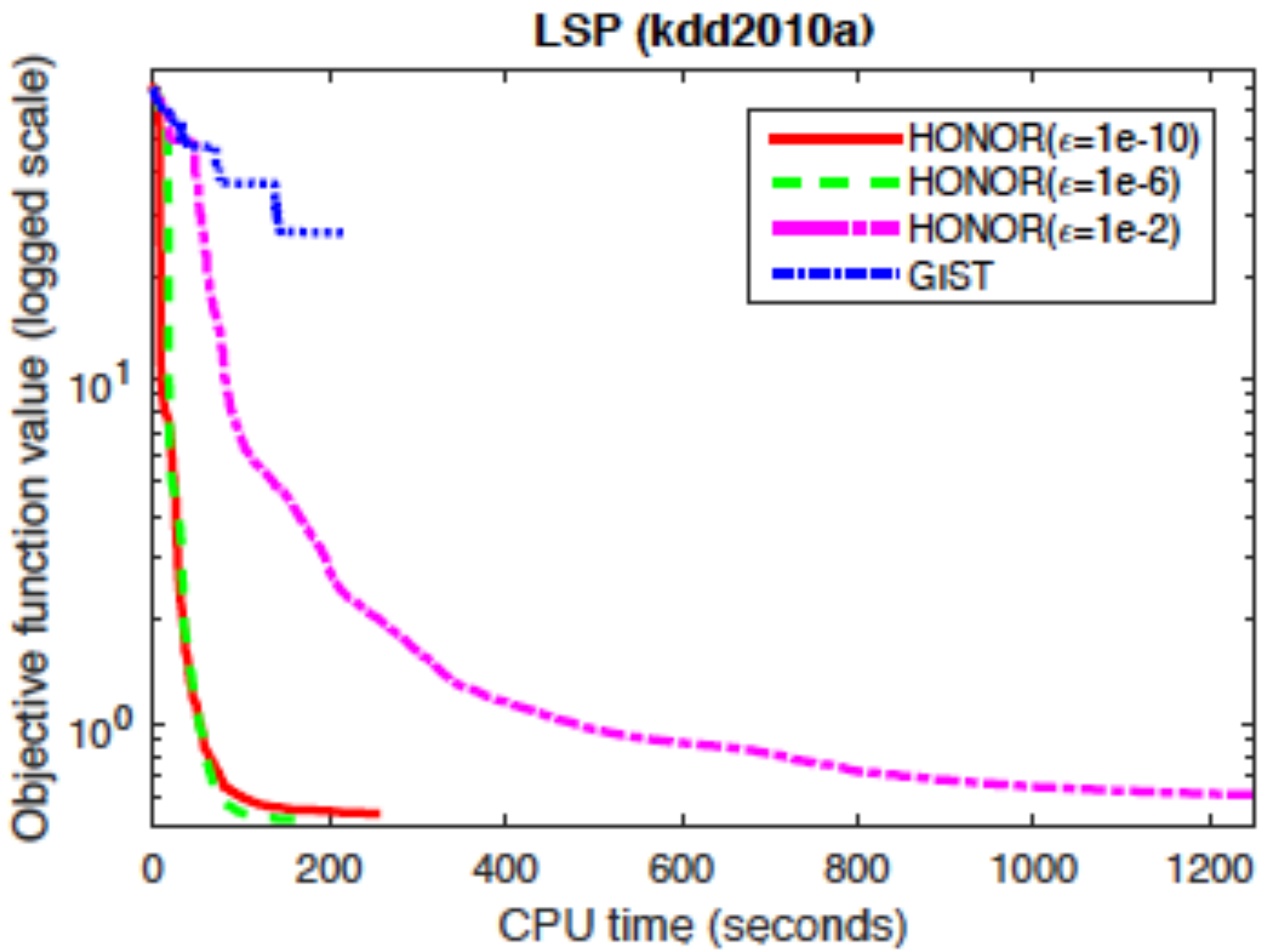
- GD-step: $\mathbf{w}^k(\alpha) \leftarrow \arg\min_{\mathbf{x}} \left\{ \nabla l(\mathbf{w}^k)^T(\mathbf{w} - \mathbf{w}^k) + \frac{1}{2\alpha}\|\mathbf{w} - \mathbf{w}^k\|^2 + \lambda\|\mathbf{w}\|_1 \right\}$

  Line search (GD): $f(\mathbf{w}^k(\alpha)) \leq f(\mathbf{w}^k) - \frac{\gamma}{2\alpha}\|\mathbf{w}^k(\alpha) - \mathbf{w}^k\|^2$

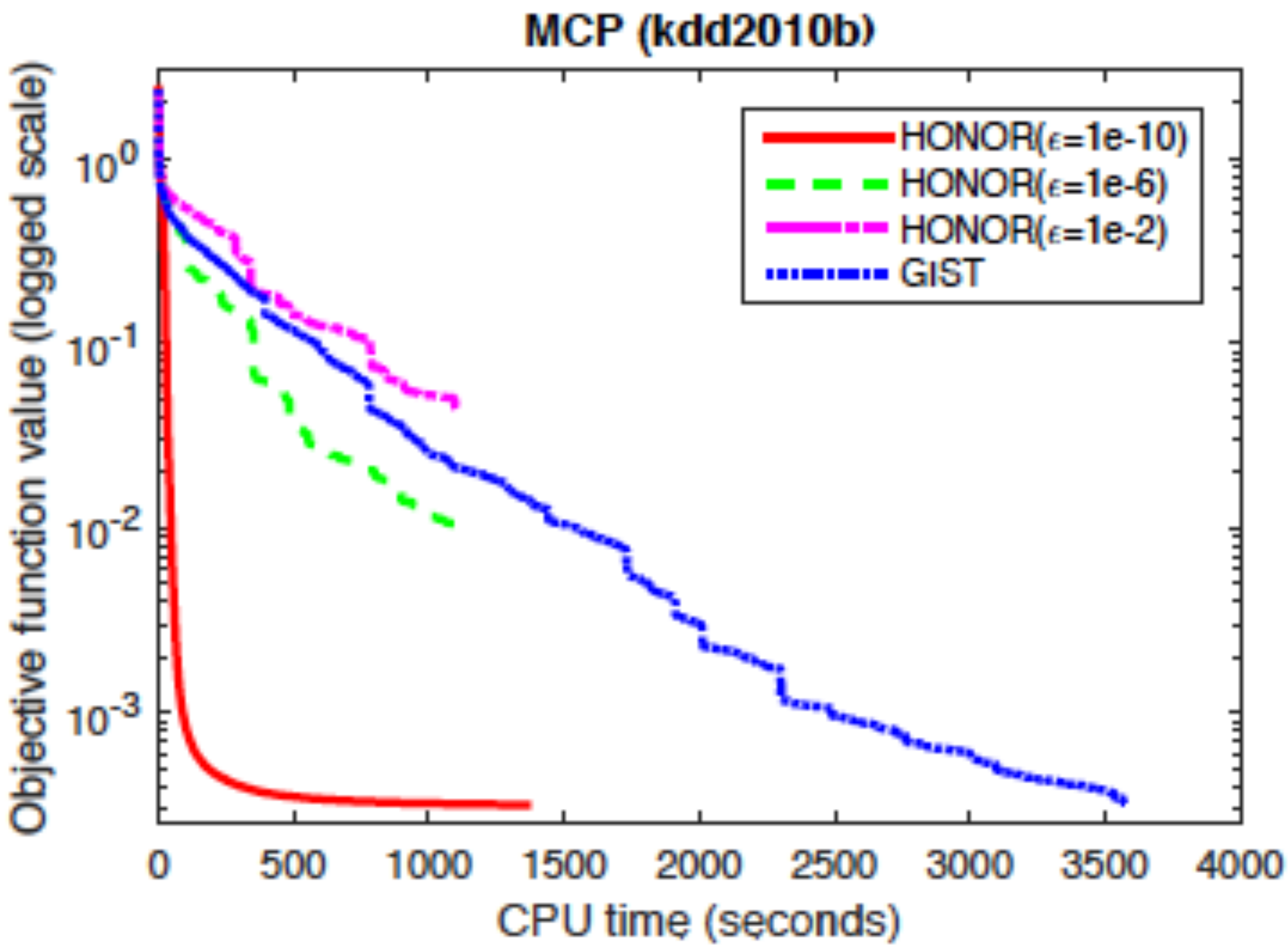
Why Hybrid Strategy

- The optimization problem is non-smooth
- The operation of projection a vector back to the previous orthant is not easy to handle
- The key difficulty: if there exists a subsequence $\mathcal{K}$ such that $\{x_i^k\}_{\mathcal{K}}$ converges to zero, it is possible that for a large enough $k \in \mathcal{K},\ |x_i^k|$ is arbitrarily small but is never equal to zero.

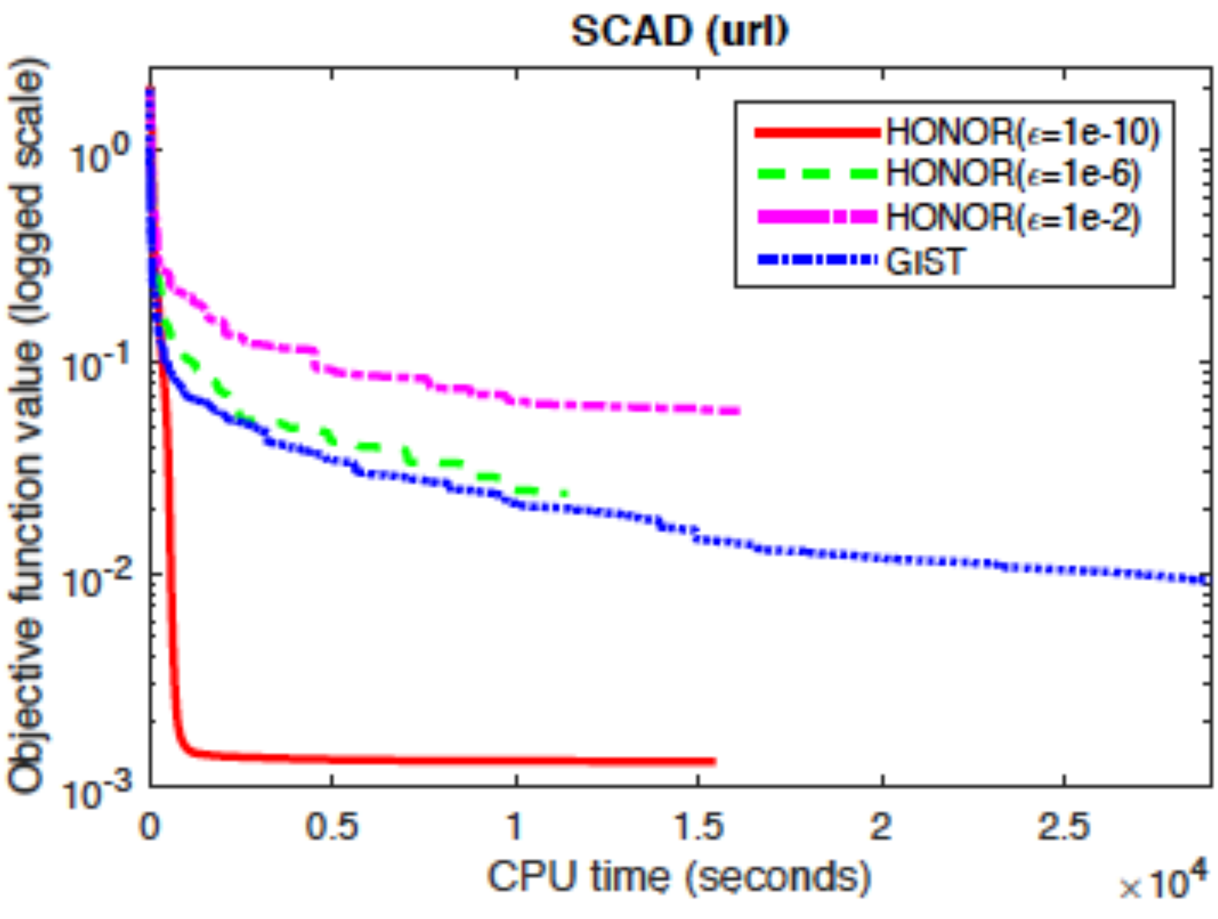
Experiments (LSP)



Experiments (MCP)



Experiments (SCAD)



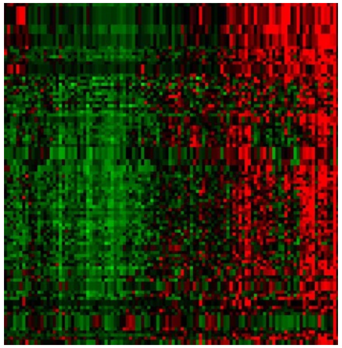
Road Map

- Introduction to Sparsity
- Convex Approaches
- Non-Convex Approaches
- **Topic: Matrix Completion**
- Topic: Multi-task Learning

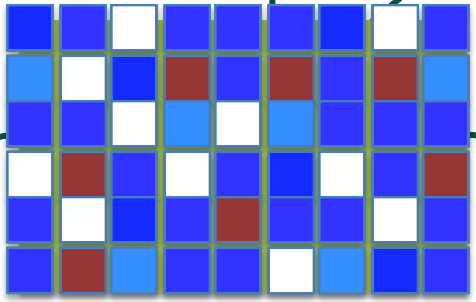
Matrix Completion

Image Inpainting

Microarray data imputation



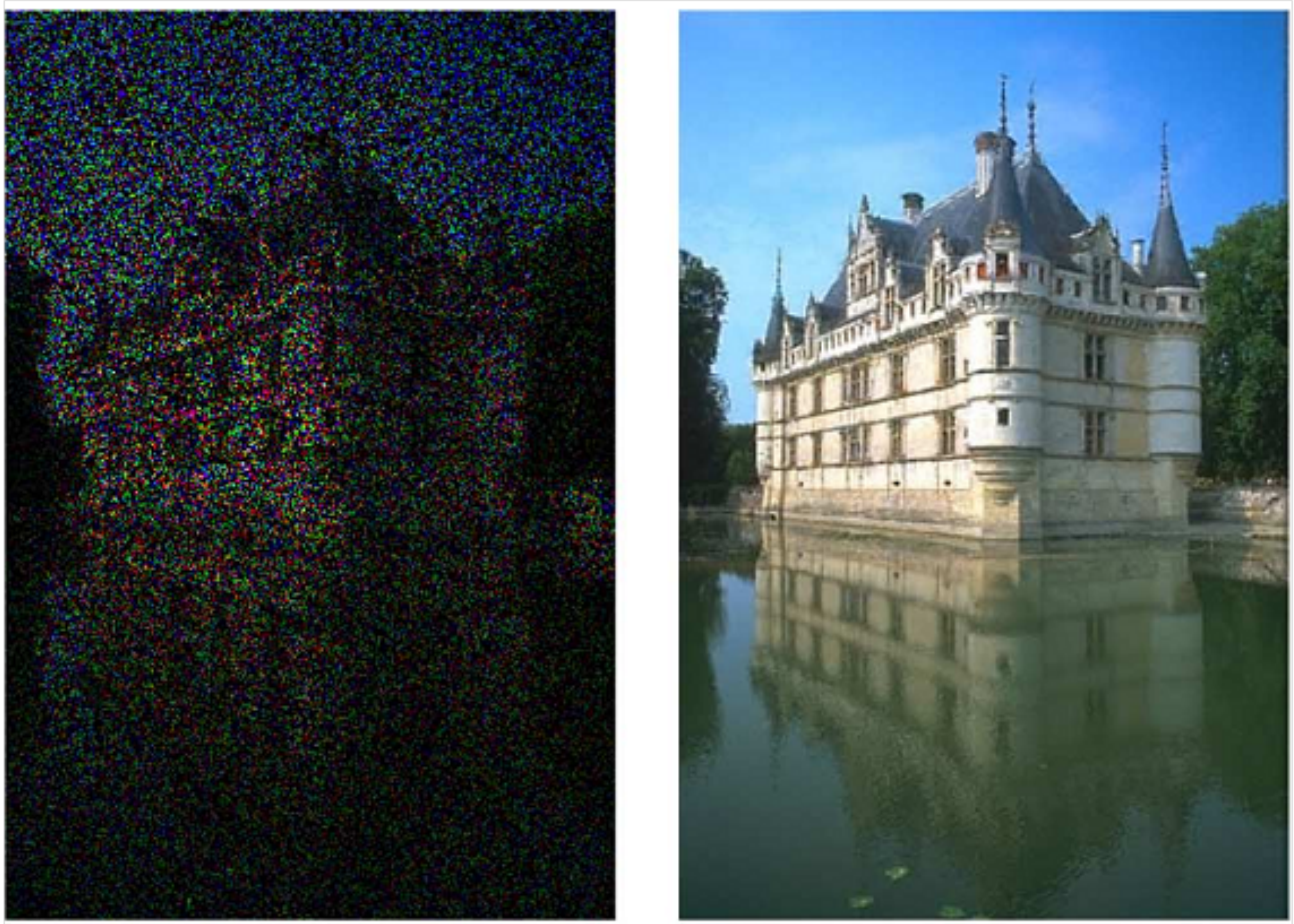
Video Recovery



Matrix Completion

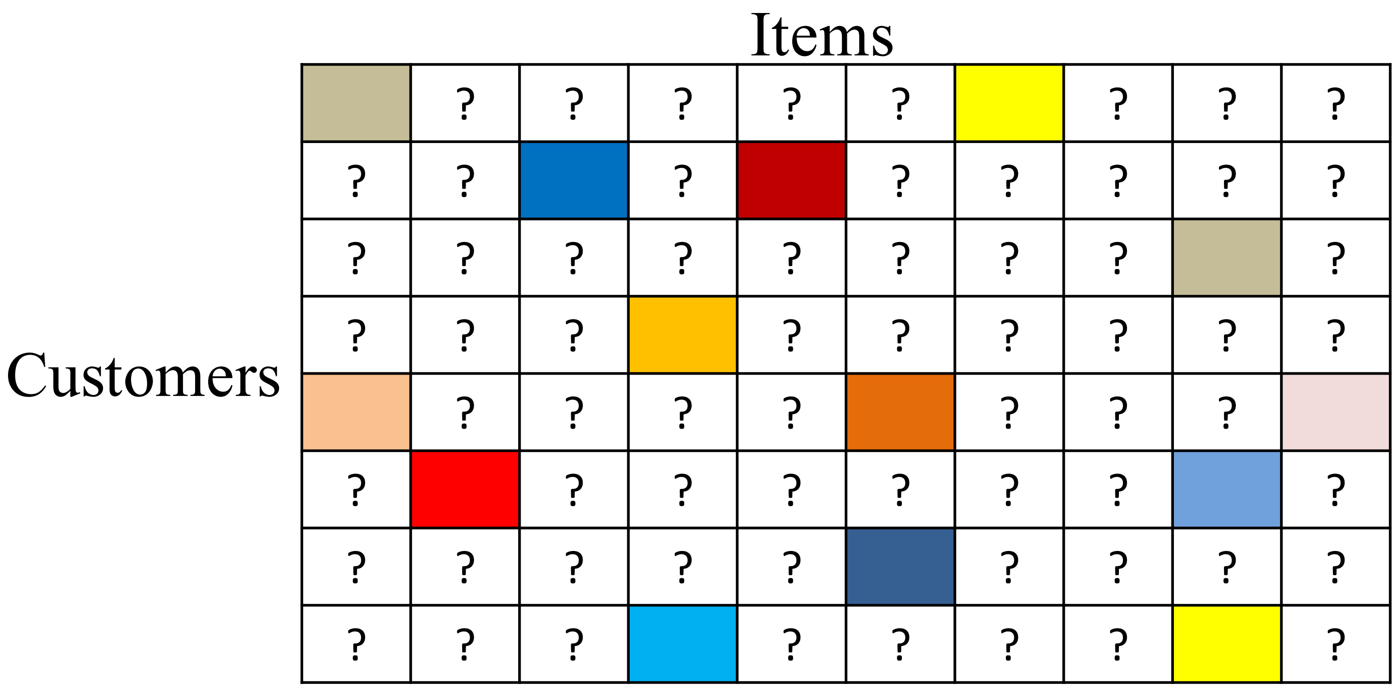
Collaborative filtering

Image Recovery



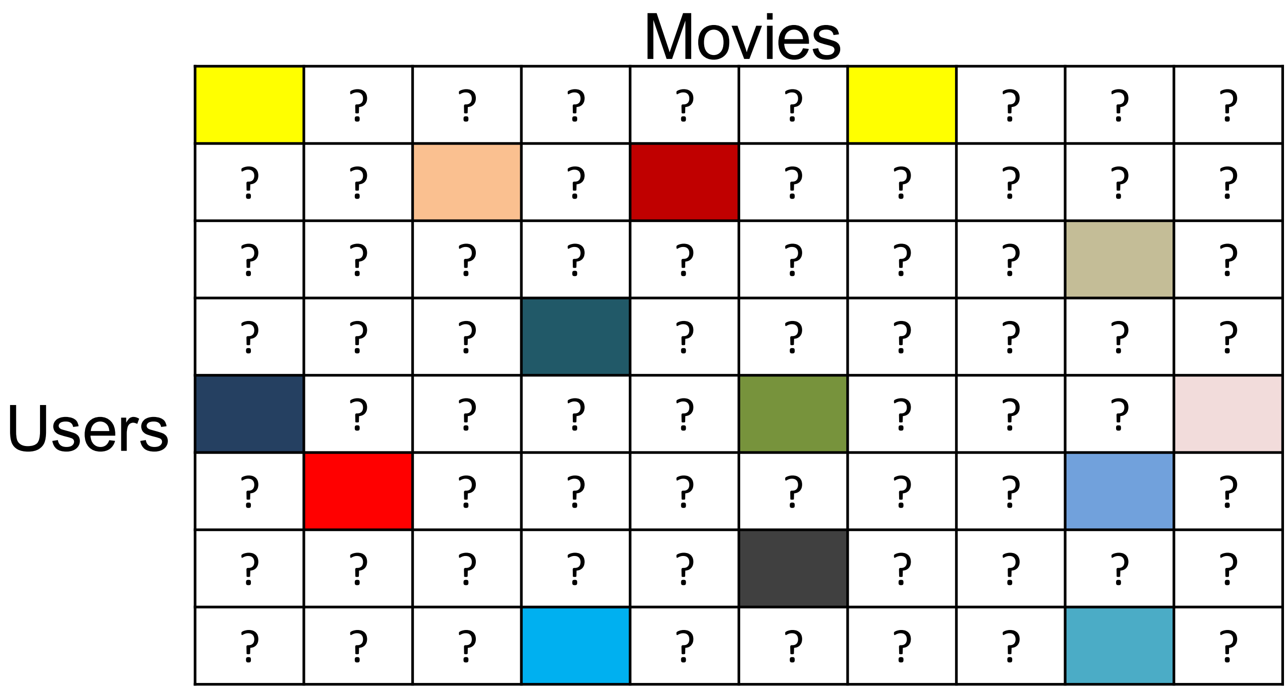
- Recover the original image with partial observation

Collaborative Filtering



- Customers are asked to rank items
- Not all customers ranked all items
- Predict the missing rankings (98.9% is missing)

The Netflix Problem

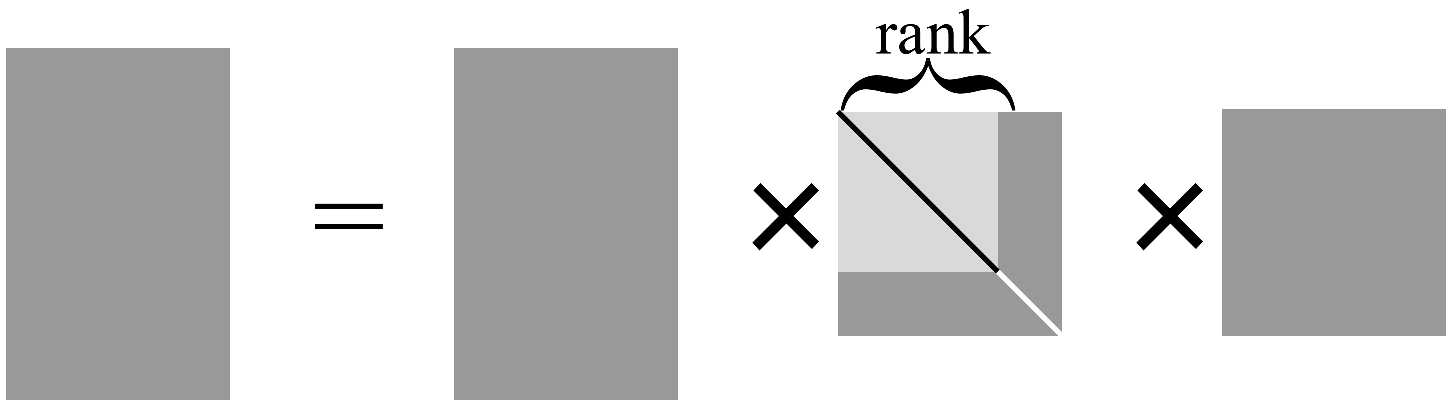


- About a million users and 25,000 movies
- Known ratings are sparsely distributed

Preferences of users are determined by a small number of factors → low rank

Matrix Rank

- The number of independent rows or columns
- The singular value decomposition (SVD):



Low Rank Matrix Completion

- Low rank matrix completion with incomplete observations can be formulated as:

$$\begin{aligned} \min_{\mathrm{X}} \quad & rank(\mathrm{X}) \\ s.t. \quad & P_{\Omega}(\mathrm{X}) = P_{\Omega}(\mathrm{Y}) \end{aligned}$$

with the projection operator defined as: $P_{\Omega}(\mathrm{X}) = \begin{cases} x_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega \end{cases}$

Other Low-Rank Problems

- Multi-Task/Class Learning
- Image compression
- Foreground-background separation problem in computer vision
- Low rank metric learning in machine learning
- Other settings:
  - System identification in control theory
  - low-degree statistical model for a random process
  - a low-order realization of a linear system
  - a low-order controller for a plant
  - a low-dimensional embedding of data in Euclidean space

Two Formulations for Rank Minimization

$$\min \quad \text{loss}(X) + \lambda^*\text{rank}(X)$$

$$\text{loss}(X) = \frac{1}{2}\left\| P_\Omega(\mathrm{X}) - P_\Omega(\mathrm{Y}) \right\|_F^2$$

$$\begin{aligned} &\min && \text{rank}(X) \\ &\text{subject to} && \text{loss}(X) \le \varepsilon \end{aligned}$$

Rank minimization is NP-hard

Trace Norm (Nuclear Norm)

Trace norm of a matrix is the sum of its singular values:

$$X = U \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_k \end{pmatrix} V^T$$

$$\| X \|_* = \sum_{i=1}^{k} \sigma_i$$

- **trace norm $\Leftrightarrow$ 1-norm of the vector of singular values**
- **trace norm is the convex envelope of the rank function over the unit ball of spectral norm $\Rightarrow$ a convex relaxation**

Two Convex Formulations

$$\min \quad \text{loss}(X) + \lambda \times \|X\|_*$$

$$\begin{aligned} &\min && \|X\|_* \\ &\text{subject to} && \text{loss}(X) \leq \varepsilon \end{aligned}$$

Trace norm minimization is **convex**

- Can be solved by semi-definite programming
  - Computationally expensive
- Recent more efficient solvers:
  - Singular value thresholding (Cai et al, 2008)
  - Fixed point method (Ma et al, 2009)
  - Accelerated gradient descent (Toh & Yun, 2009, Ji & Ye, 2009)

Trace Norm Minimization

- Trace norm convex relaxation

$$\min_{X} \|X\|_* \quad s.t. \quad P_\Omega(X) = P_\Omega(Y) \xrightarrow{\text{noisy case}} \min_{X} \frac{1}{2}\|P_\Omega(X) - P_\Omega(Y)\|_F^2 + \lambda\|X\|_*$$

Can be solved by
- sub-gradient method
- the proximal gradient method
- the conditional gradient method

Convergence speed: sub-linear

Iteration: truncated SVD or top-SVD (Frank-Wolfe)

Ref: 1. Candes, E. J. and Recht, B. Exact matrix completion via convex optimization. Foundations of Computational Mathematics, 9(6):717–772, 2009.
2. Jaggi, M. and Sulovsky, M. A simple algorithm for nuclear norm regularized problems. In ICML, 2010.

Gradient Descent for the Composite Model

(Nesterov, 2007; Beck and Teboulle, 2009)

min $f(x) = \text{loss}(x) + \lambda \times \text{penalty}(x)$

Model

$$\mathcal{M}(x_i, \gamma_i) = \underbrace{[\text{loss}(x_i) + \langle \text{loss}'(x_i), x - x_i \rangle]}_{\text{1st order Taylor expansion}} + \underbrace{\frac{1}{2\gamma_i} \|x - x_i\|_2^2}_{\text{Regularization}} + \underbrace{\lambda \times \text{penalty}(x)}_{\text{Nonsmooth part}}$$

Repeat

$$x_{i+1} = \arg\min \mathcal{M}(x_i, \gamma_i)$$

Until "convergence"

Convergence rate $O(1/N)$

Proximal Operator Associated with Trace Norm

Optimization problem

$$\min_X f(X) = \text{loss}(X) + \lambda \|X\|_*$$

Associated proximal operator

$$X^* = \pi_{tr}(V) = \arg\min_X \frac{1}{2}\|X - V\|_2^2 + \lambda \times \|X\|_*$$

Closed form solution: $X^* = P\text{diag}(\tilde{\sigma})Q^{\mathrm{T}},$

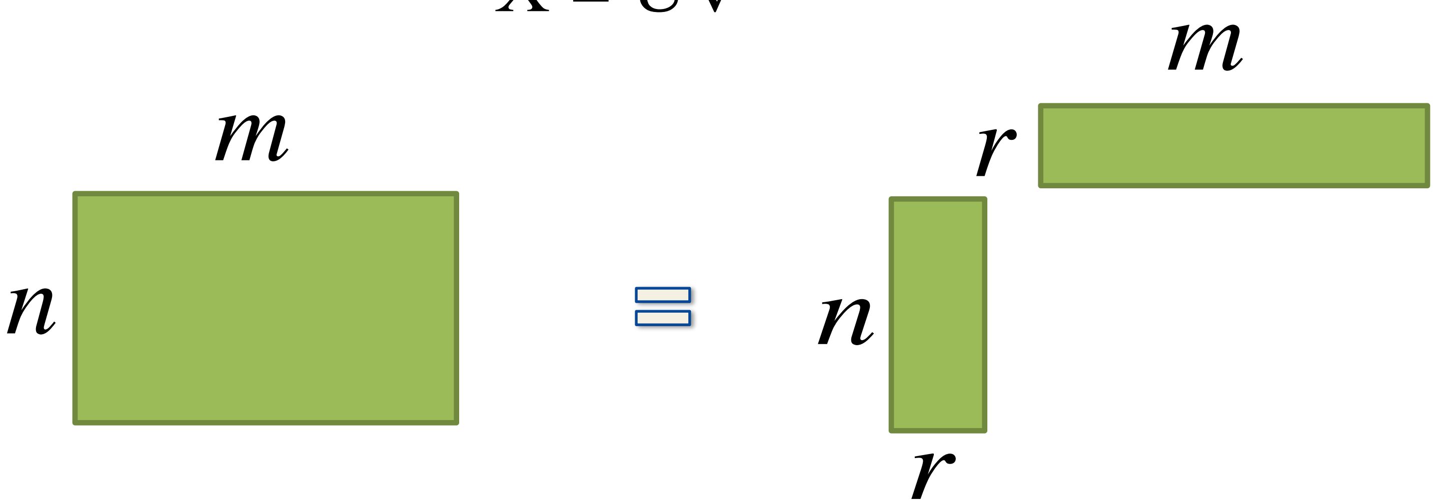
where $V = P\text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_k)Q^{\mathrm{T}}$ is the SVD of $V \in \mathbb{R}^{m\times n}$, $k = \min(m, n)$, $P \in \mathbb{R}^{m\times k}$, $Q \in \mathbb{R}^{n\times k}$, and

$$\tilde{\sigma}_i = \begin{cases} v_i - \lambda & \sigma_i > \lambda \\ 0 & \sigma_i \leq \lambda \end{cases}$$

A Non-convex Formulation via Matrix Factorization

- Rank-*r* matrix X can be written as a product of two smaller matrices U and V

$$X = UV^T$$



$$\|X\|_* = \min_{X=UV^T} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right)$$

Alternating Optimization

$$\boxed{\min_{\mathrm{U,V}} \quad \left\| P_{\Omega}(\mathrm{UV}^T) - P_{\Omega}(\mathrm{Y}) \right\|_F^2 + \frac{1}{2}\left(\|\mathrm{U}\|_F^2 + \|\mathrm{V}\|_F^2 \right)}$$

Non-convex

- Can be solved via
  - Alternating minimization (Jain et al, 2012)

Theoretical Result

$$\boxed{\min_{\mathrm{U}\in R^{m\times k},\mathrm{V}\in R^{n\times k}} \left\| P_\Omega(\mathrm{U}\mathrm{V}^T) - P_\Omega(\mathrm{Y}) \right\|_F^2}$$

$$\mathrm{V}_{t+1} = \operatorname*{argmin}_{\mathrm{V}\in R^{n\times k}} \left\| P_{\Omega_{t+1}}(\mathrm{U}_t \mathrm{V}^T - \mathrm{Y}) \right\|_F^2$$

$$\mathrm{U}_{t+1} = \operatorname*{argmin}_{\mathrm{U}\in R^{m\times k}} \left\| P_{\Omega_{T+t+1}}(\mathrm{U}\mathrm{V}_{t+1}^T - \mathrm{Y}) \right\|_F^2$$

- Under certain condition with proper initialization, alternating optimization algorithm guarantee geometric convergence.

Practical Algorithm

$$\begin{aligned} \min_{\mathrm{U,V,Z}} \quad & \left\| \mathrm{UV}^T - \mathrm{Z} \right\|_F^2 \\ s.t. \quad & P_\Omega(\mathrm{Z}) = P_\Omega(\mathrm{Y}) \end{aligned}$$

$$L = \left\| \mathrm{UV}^T - \mathrm{Z} \right\|_F^2 - \Lambda \bullet P_\Omega(\mathrm{Z} - \mathrm{Y})$$

- The Lagrangian function can be solved by alternating optimization method.
- Weak convergence guarantee

Robust Matrix Completion

$$\begin{array}{ll} \min\limits_{U,V,Z} & \|P_\Omega(Z-Y)\|_1 \\ s.t. & UV^T - Z \end{array}$$

$$L = \|P_\Omega(Z-Y)\|_1 + \langle \Lambda, UV^T - Z \rangle + \frac{\beta}{2}\|UV^T - Z\|_F^2$$

- The robust matrix completion problem can be solved by augmented Lagrangian alternating direction method.
- Weak convergence guarantee

Summary of Two Approaches

- Trace norm convex relaxation

$$\min_{X} \|X\|_* \quad s.t. \quad P_\Omega(X) = P_\Omega(Y)$$

noisy case

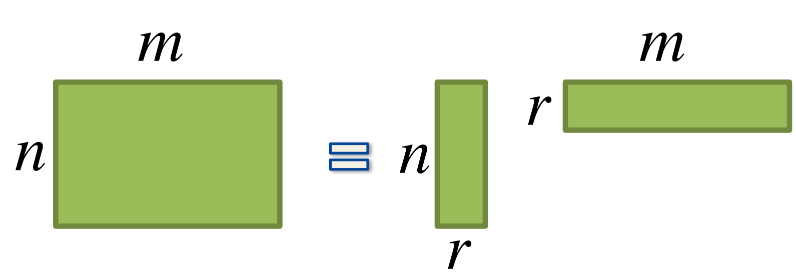
$$\min_{X} \|P_\Omega(X) - P_\Omega(Y)\|_F^2 + \lambda \|X\|_*$$

Projection operator: $P_\Omega(X) = \begin{cases} x_{ij} & (i,j) \in \Omega \\ 0 & (i,j) \notin \Omega \end{cases}$

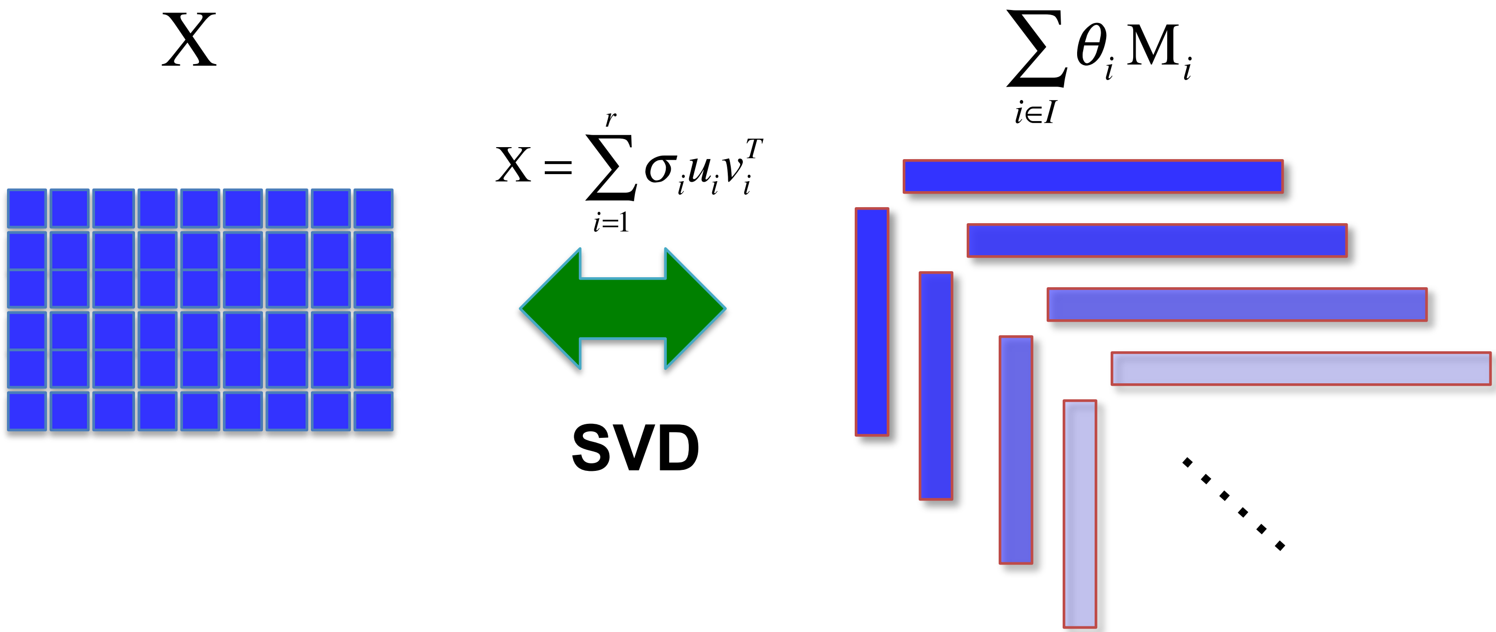
- Bilinear non-convex relaxation

$$X = UV^T$$

$$\min_{U,V} \|P_\Omega(UV^T) - P_\Omega(Y)\|_F^2$$



Rank-One Matrix Space



Rank-one matrices with unit norm as *Atoms*

$$\mathrm{M} \in \Re^{n \times m} \quad \text{for} \quad \mathrm{M} = uv^T \quad u \in \Re^n \quad v \in \Re^m$$

Matrix Completion in Rank-One Matrix Space

- Matrix completion in rank-one matrix space

$$\begin{aligned} \min_{\boldsymbol{\theta}\in\Re^{I},\{M_i\}} \quad & \|\boldsymbol{\theta}\|_0 \\ s.t. \quad & P_\Omega(\mathrm{X}(\theta)) = P_\Omega(\mathrm{Y}) \end{aligned}$$

with the estimated matrix in the rank-one matrix space as $\mathrm{X}(\boldsymbol{\theta}) = \sum_{i\in I} \theta_i \mathrm{M}_i$

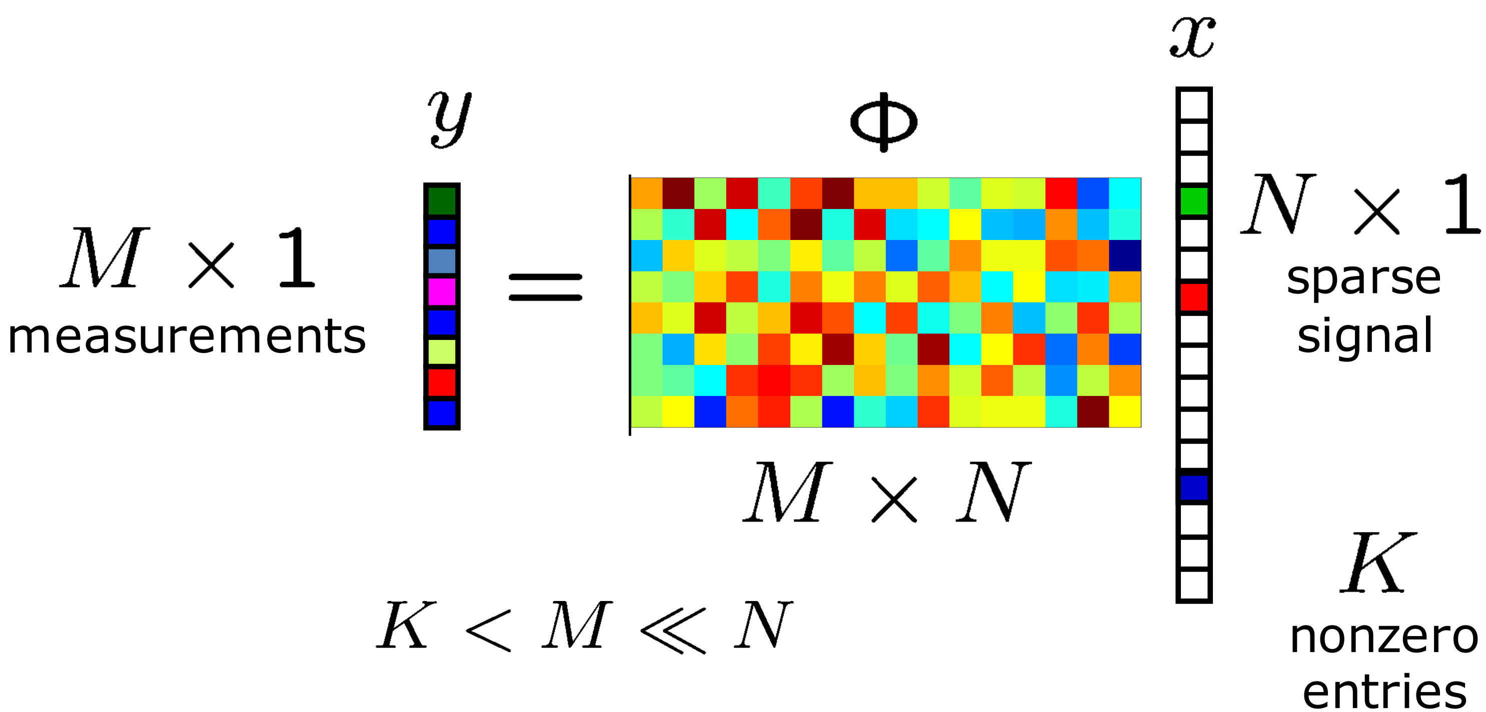
- Reformulation in the noisy case

$$\begin{aligned} \min_{\mathrm{X}(\boldsymbol{\theta})} \quad & \|P_\Omega(\mathrm{X}(\theta)) - P_\Omega(\mathrm{Y})\|_F^2 \\ s.t. \quad & \|\boldsymbol{\theta}\|_0 \le r \end{aligned}$$

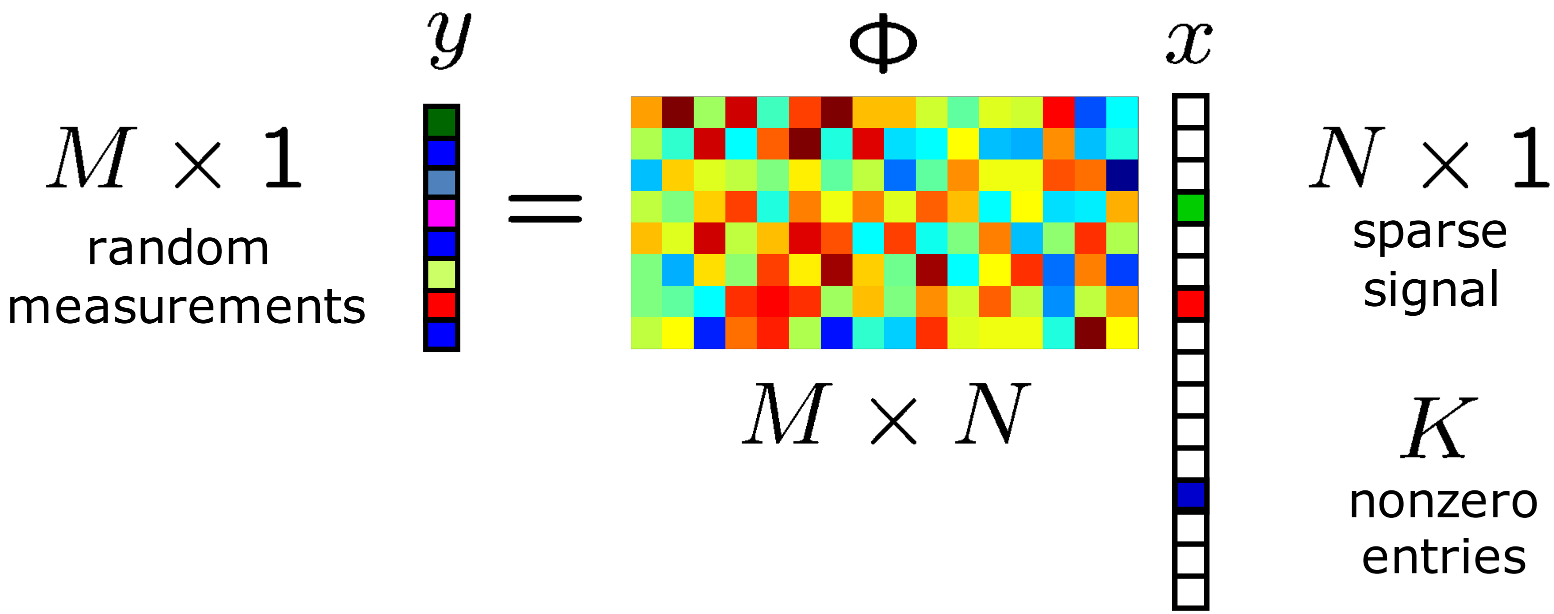
We solve this problem using an orthogonal matching pursuit type greedy algorithm. The candidate set is an infinite set composed by all rank-one matrices $\mathrm{M} \in \Re^{n\times m}$

Vector Case: Compressive Sensing

- When data is sparse/compressible, can directly acquire a ***condensed representation*** $y = \Phi x$



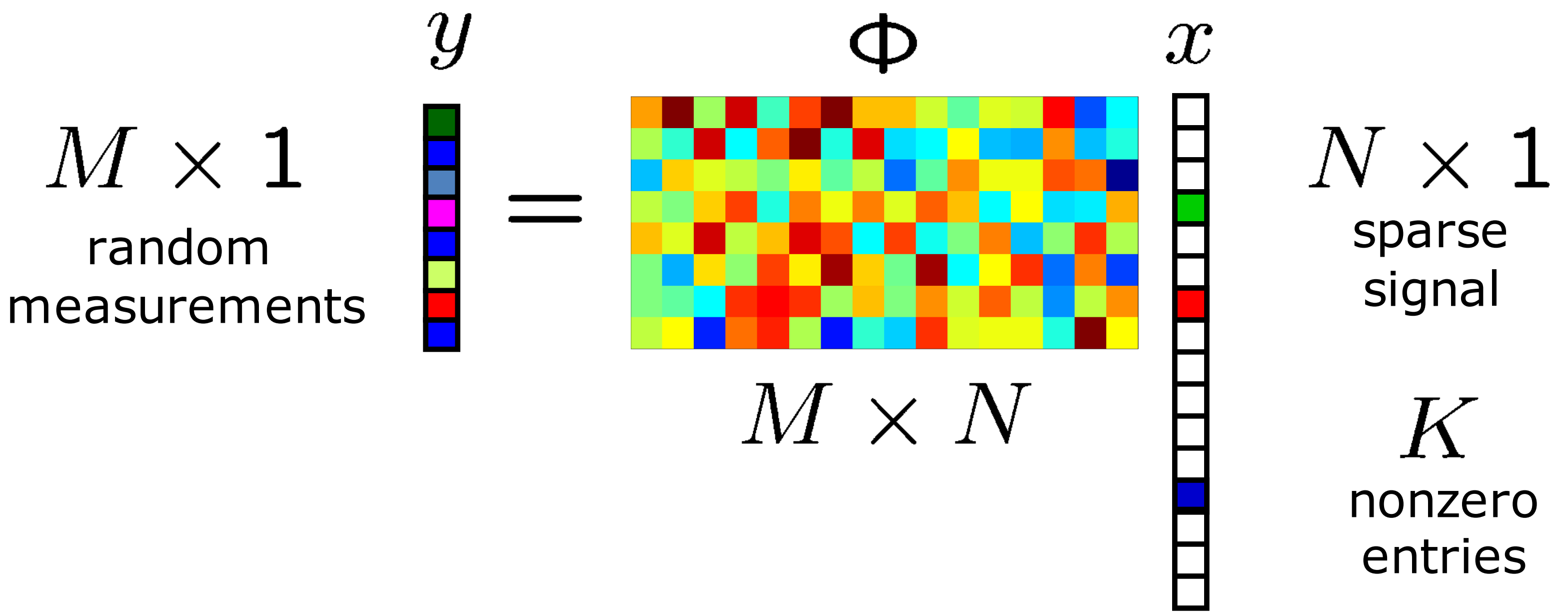
Convex Formulation for Recovery



- Signal **recovery** via ℓ_1 optimization
  [Candes, Romberg, Tao; Donoho]

$$\widehat{x} = \arg\min_{y=\Phi x} \|x\|_1$$

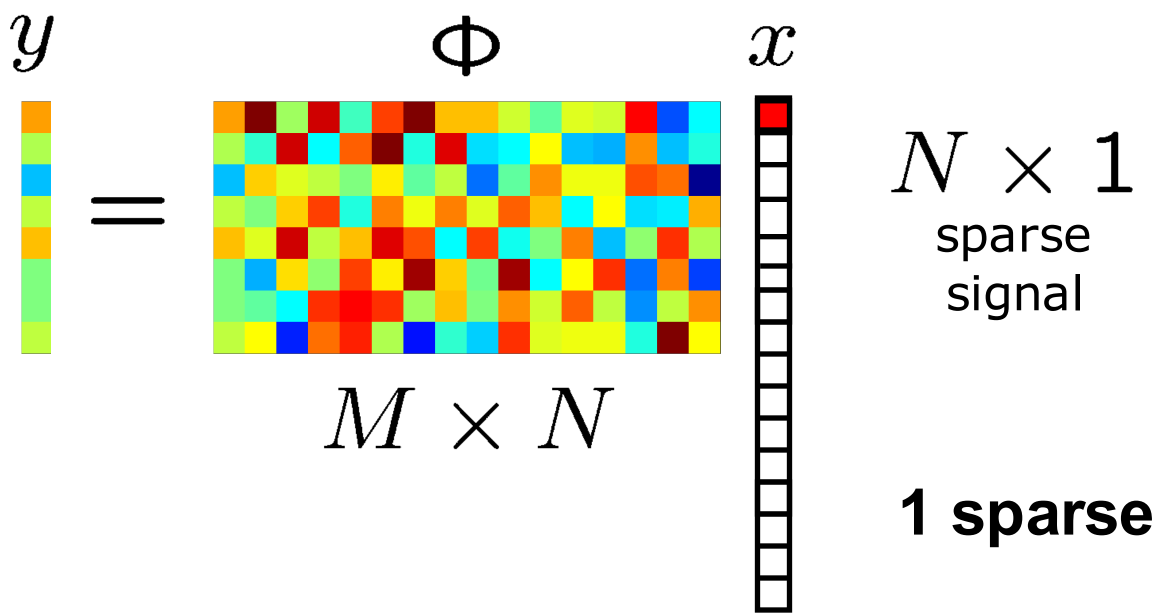
Greedy Algorithms



- Signal **recovery** via iterative greedy algorithms
  - (orthogonal) matching pursuit [Gilbert, Tropp]
  - iterated thresholding [Nowak, Figueiredo; Kingsbury, Reeves; Daubechies, Defrise, De Mol; Blumensath, Davies; …]
  - CoSaMP [Needell and Tropp]

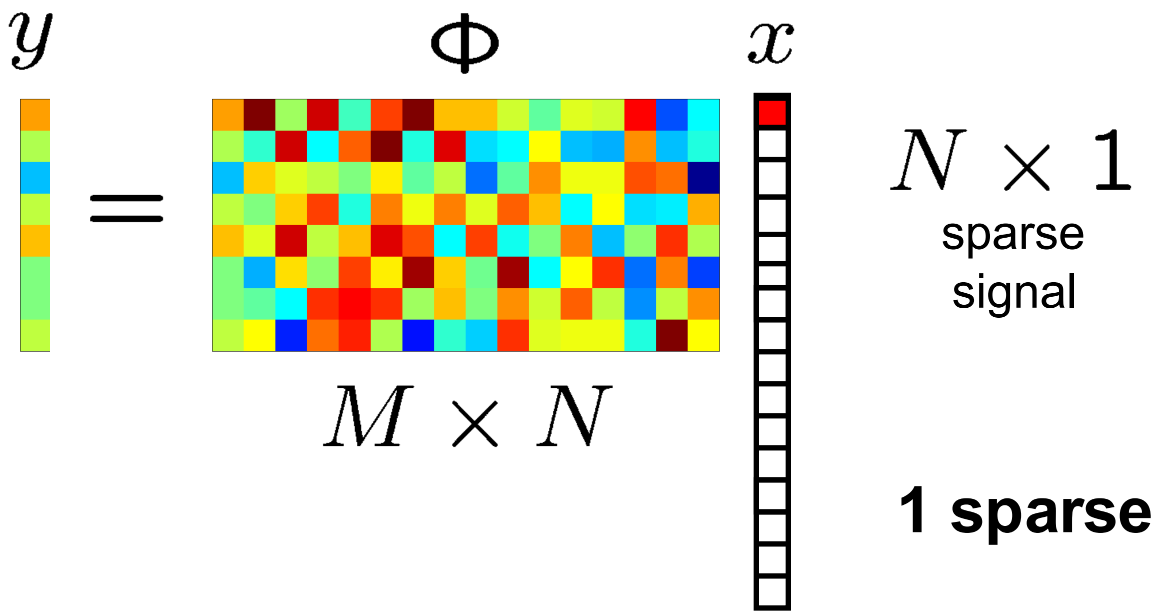
Greedy Recovery Algorithm (1)

- Consider the following problem



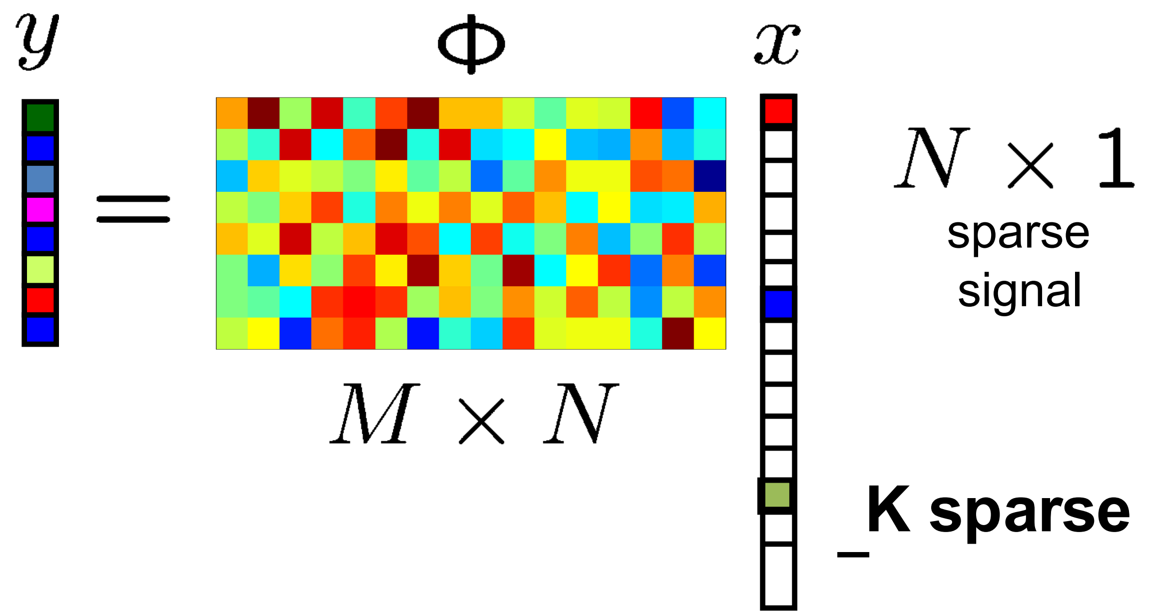
- Can we recover the **support**?
  - 1-Sparse (only one support)
  - K-Sparse

Greedy Recovery Algorithm (2)



- If $\Phi = [\phi_1, \phi_2, \ldots, \phi_N]$
  then $\arg\max |\langle \phi_i, y \rangle|$ gives the support of *x*
- How to extend to *K*-sparse signals?

Greedy Recovery Algorithm (3)



Residue: $r = y - \Phi \widehat{x}_{k-1}$

Find atom: $k = \arg\max \left| \langle \phi_i , r \rangle \right|$

Add atom to support: $S = S \bigcup \{k\}$

Signal estimate $x_k = (\Phi_S)^{\dagger} y$

Orthogonal Matching Pursuit

goal:
given $y = \Phi x$, recover a sparse x
columns of Φ are unit-norm

initialize: $\widehat{x}_0 = 0, r = y, \Lambda = \{\}, i = 0$

iteration:

- $\circ\ i = i + 1$
- $\circ\ b = \Phi^T r$
- $\circ\ \boxed{k = \arg\max\{|b(1)|, |b(2)|, \ldots, |b(N)|\}}$ **Find atom with largest support**
- $\circ\ \Lambda = \Lambda \bigcup k$
- $\circ\ \boxed{(\widehat{x}_i)_{|\Lambda} = (\Phi_{|\Lambda})^\dagger y,\ (\widehat{x}_i)_{|\Lambda^c} = 0}$ **Update signal estimate**
- $\circ\ \boxed{r = y - \Phi \widehat{x}_i}$ **Update residual**

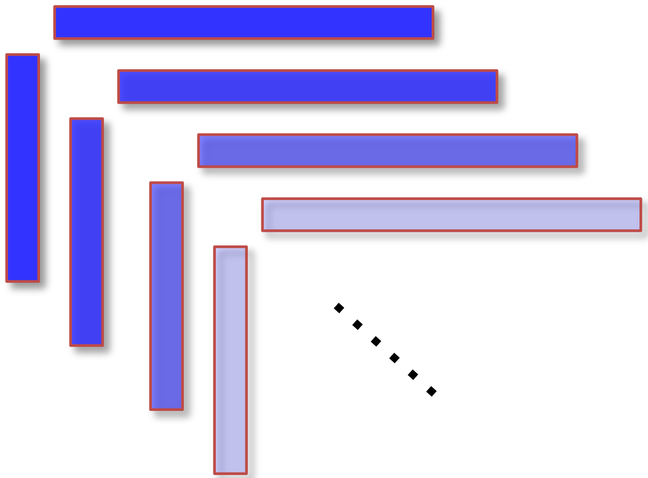
Baraniuk et al., 2012

Orthogonal Rank-One Matrix Pursuit for Matrix Completion

- Matrix completion in rank-one matrix space

$$\min_{X(\boldsymbol{\theta})} \quad \left\| P_\Omega(X(\boldsymbol{\theta})) - P_\Omega(Y) \right\|_F^2$$
$$s.t. \quad \left\| \boldsymbol{\theta} \right\|_0 \le r$$

$$X(\boldsymbol{\theta}) = \sum_{i \in I} \theta_i M_i$$



We solve this problem using an orthogonal matching pursuit type greedy algorithm. The candidate set is an infinite set composed by all rank-one matrices.

Rank-One Matrix Basis

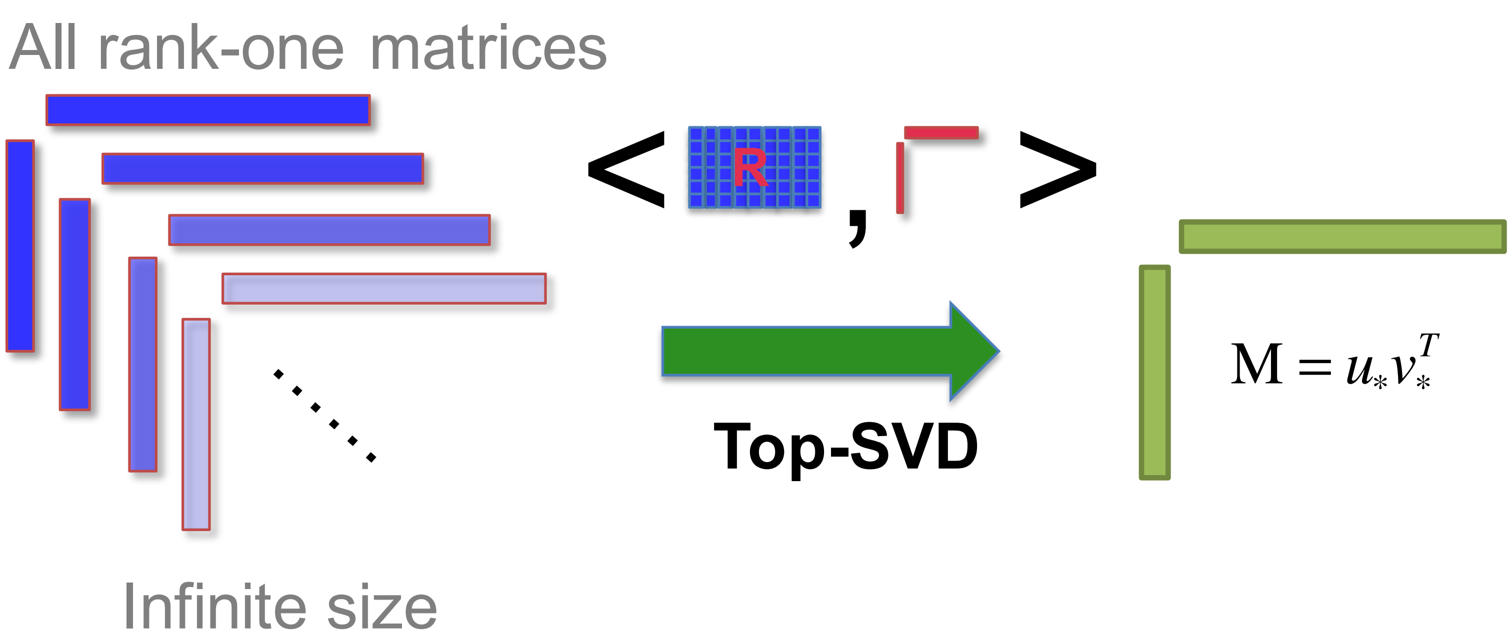
Step 1: basis construction

$$[u_*, v_*] = \underset{\|u\|=1, \|v\|=1}{\operatorname{argmax}} \left\langle \mathrm{R}, uv^T \right\rangle = u^T \mathrm{R}\, v$$

with residual matrix

$$\mathrm{R} = \mathrm{Y}_\Omega - \mathrm{X}_\Omega$$

$\mathrm{M} = u_* v_*^T$ is selected from all rank-one matrices with unit norm.



Rank-One Matrix Pursuit Algorithm

Step 1: construct the optimal rank-one matrix basis

$$[u_*, v_*] = \underset{u,v}{\arg\max} \left\langle (\mathrm{Y} - \mathrm{X}_k)_\Omega, uv^T \right\rangle \qquad \mathrm{M}_{k+1} = u_* v_*^T$$

This is the top singular vector pair, which can be solved efficiently by power method.

This generalizes OMP with *infinite* dictionary set of all rank-one matrices $\mathrm{M} \in \Re^{n \times m}$

Step 2: calculate the optimal weights for current bases

$$\theta^k = \underset{\theta \in \Re^k}{\arg\min} \left\| \sum_i \theta_i \mathrm{M}_i - \mathrm{Y} \right\|_\Omega^2$$

This is a least squares problem, which can be solved incrementally.

Linear Convergence

- Linear upper bound for the algorithm to converge

Theorem 3.1. *The rank-one matrix pursuit algorithm satisfies*

$$||\mathbf{R}_k|| \ \leq \ \gamma^{k-1}\|\mathbf{Y}\|_\Omega, \quad \forall k \geq 1.$$

γ is a constant in $[0,1)$.

This is significantly different from the standard MP/OMP algorithm with a finite dictionary, which are known to have a sub-linear convergence speed at the worst case.

At each iteration, we guarantee a significant reduction of the residual, which depends on the top singular vector pair pursuit step.

Z. Wang et al. **ICML'14; SIAM J. Scientific Computing 2015**

Efficiency and Scalability

- An efficient and scalable algorithm for matrix completion: Rank-One Matrix Pursuit
  - **Scalability**: top-SVD
  - **Convergence**: linear convergence

Z. Wang et al. **ICML'14**; **SIAM J. Scientific Computing 2015**

Related Work

Atomic decomposition $\mathrm{X} = \sum_{i \in I} \theta_i \mathrm{M}_i$

can be solved by matching pursuit type algorithms.

- Vs. Frank-Wolfe algorithm (FW)

  Similarity: top-SVD

  Difference: linear convergence Vs. sub-linear convergence

- Vs. existing greedy approach (ADMiRA)

  Similarity: linear convergence

  Difference: 1. top-SVD Vs. truncated SVD
  2. no extra condition for linear convergence

Ref: Lee, K. and Bresler, Y. Admira: atomic decomposition for minimum rank approximation. IEEE Trans. on Information Theory, 56(9):4402–4416, 2010.

Time and Storage Complexity

- Time complexity

	R1MP	ADMiRA & AltMin	FW	Proximal	SVT
Each Iter.	$O(\|\Omega\|)$	$O(r\|\Omega\|)$	$O(\|\Omega\|)$	$O(r\|\Omega\|)$	$O(r\|\Omega\|)$
Iterations	$O(\log(1/\varepsilon))$	$O(\log(1/\varepsilon))$	$O(1/\varepsilon)$	$O(1/\sqrt{\varepsilon})$	$O(1/\varepsilon)$
Total	$O(\|\Omega\|\log(1/\varepsilon))$	$O(r\|\Omega\|\log(1/\varepsilon))$	$O(\|\Omega\|/\varepsilon)$	$O(r\|\Omega\|/\sqrt{\varepsilon})$	$O(r\|\Omega\|/\varepsilon)$

minimum iteration cost
+ linear convergence

Storage complexity

Economic Rank-One Matrix Pursuit

- **Step 1**: find the optimal rank-one matrix basis

$$[u_*, v_*] = \underset{u,v}{\operatorname{argmax}} \left\langle (\mathrm{Y} - \mathrm{X}_k)_\Omega, uv^T \right\rangle \qquad \mathbf{M}_{k+1} = u_* v_*^T$$

- **Step 2: calculate the weights for two matrices**

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha} \in \Re^2}{\arg\min} \left\| \alpha_1 \boxed{\mathrm{X}_k} + \alpha_2 \boxed{\mathrm{M}_{k+1}} - \mathrm{Y} \right\|_\Omega^2$$

$$\theta_i^{k-1} = \theta_i^{k-1} \alpha_1 \qquad \theta_i^k = \alpha_2$$

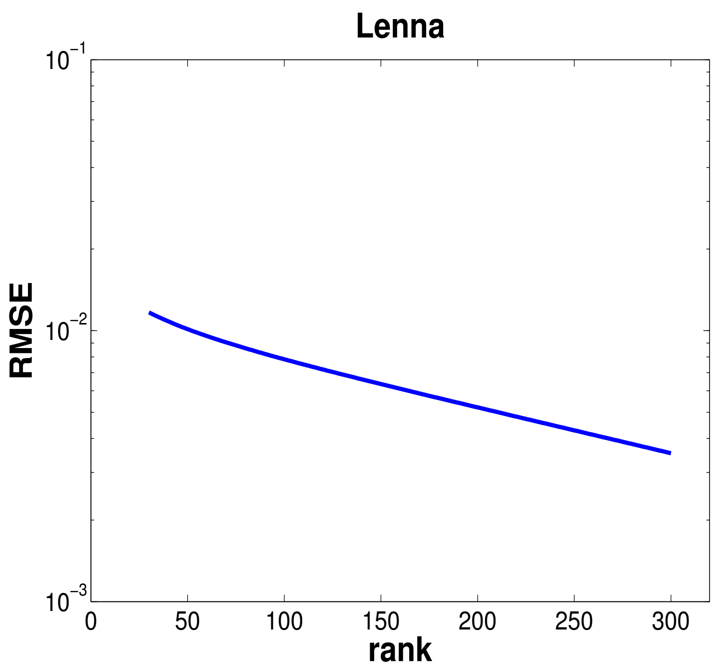
- It retains the linear convergence

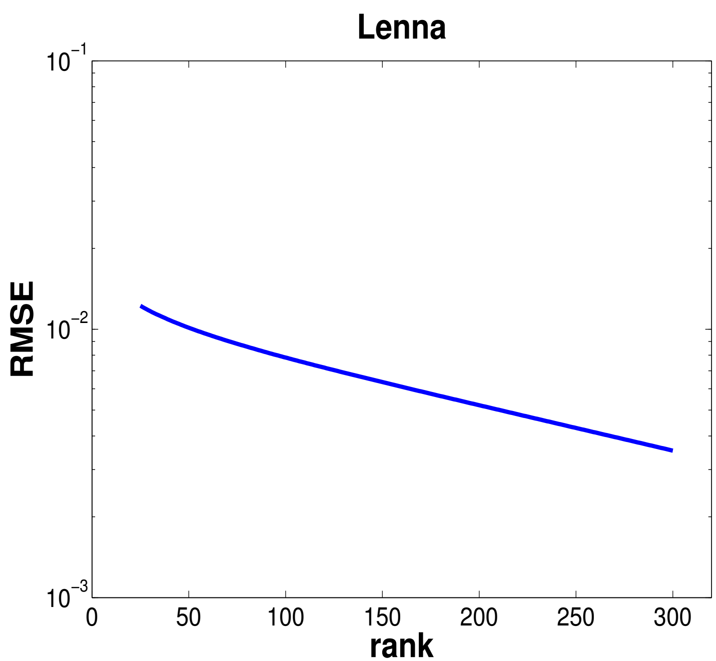
Theorem 4.1. *The economic rank-one matrix pursuit algorithm satisfies*

$$||\mathbf{R}_k|| \ \leq \ \tilde{\gamma}^{k-1} \|\mathbf{Y}\|_\Omega, \quad \forall k \geq 1.$$

$\tilde{\gamma}$ is a constant in $[0, 1)$.

Convergence





Residual curves of the Lena image for R1MP and ER1MP in log-scale

Experiments

- Experiments
  - Collaborative filtering
  - Image recovery
  - Convergence property
- Competing algorithms
  - singular value projection (SVP)
  - spectral regularization algorithm (SoftImpute)
  - low rank matrix fitting (LMaFit)
  - alternating minimization (AltMin)
  - boosting type accelerated matrix-norm penalized solver (Boost)
  - Jaggi's fast algorithm for trace norm constraint (JS)
  - greedy efficient component optimization (GECO)
  - Rank-one matrix pursuit (R1MP)
  - Economic rank-one matrix pursuit (ER1MP)

trace norm minimization

alternating optimization

atomic decomposition

Collaborative Filtering

Running time for different algorithms

Dataset	SVP	SoftImpute	LMaFit	AltMin	Boost	JS	GECO	R1MP	ER1MP
Jester1	18.35	161.49	3.68	11.14	93.91	29.68	$> 10^4$	1.83	0.99
Jester2	16.85	152.96	2.42	10.47	261.70	28.52	$> 10^4$	1.68	0.91
Jester3	16.58	10.55	8.45	12.23	245.79	12.94	$> 10^3$	0.93	0.34
MovieLens100K	1.32	128.07	2.76	3.23	2.87	2.86	10.83	0.04	0.04
MovieLens1M	18.90	59.56	30.55	68.77	93.91	13.10	$> 10^4$	0.87	0.54
MovieLens10M	$> 10^3$	$> 10^3$	154.38	310.82	–	130.13	$> 10^5$	23.05	13.79

Prediction accuracy in terms of RMSE

Dataset	SVP	SoftImpute	LMaFit	AltMin	Boost	JS	GECO	R1MP	ER1MP
Jester1	4.7311	5.1113	4.7623	4.8572	5.1746	4.4713	4.3680	4.3418	4.3384
Jester2	4.7608	5.1646	4.7500	4.8616	5.2319	4.5102	4.3967	4.3649	4.3546
Jester3	8.6958	5.4348	9.4275	9.7482	5.3982	4.6866	5.1790	4.9783	5.0145
MovieLens100K	0.9683	1.0354	1.2308	1.0042	1.1244	1.0146	1.0243	1.0168	1.0261
MovieLens1M	0.9085	0.8989	0.9232	0.9382	1.0850	1.0439	0.9290	0.9595	0.9462
MovieLens10M	0.8611	0.8534	0.8625	0.9007	–	0.8728	0.8668	0.8621	0.8692

Summary

- Matrix completion background
- Trace norm convex formulation
- Matrix factorization: non-convex formulation
- Orthogonal rank-one matrix pursuit
  - Efficient update: top SVD
  - Fact convergence rate: linear

Road Map

- Introduction to Sparsity
- Convex Approaches
- Non-Convex Approaches
- Topic: Matrix Completion
- **Topic: Multi-task Learning**

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	...
72981	1985	95	...	83%	...

Exam score
?

$\underbrace{\text{Birth year} \quad \text{Previous score} \quad \ldots}_{\text{student-dependent}}$ $\underbrace{\text{School ranking} \quad \ldots}_{\text{school-dependent}}$

⋮

School 138 - **Jefferson Intermediate School**

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

Exam score
?

student-dependent school-dependent

School 139 - **Rosemead High School**

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

Exam score
?

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

⋮

School 138 - **Jefferson Intermediate School**

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

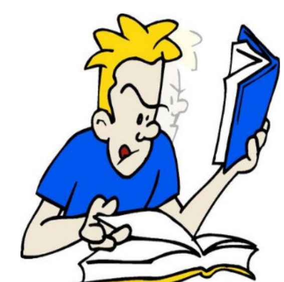
⇦ task 138th

Excellent

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

School 1 - Alverno High School

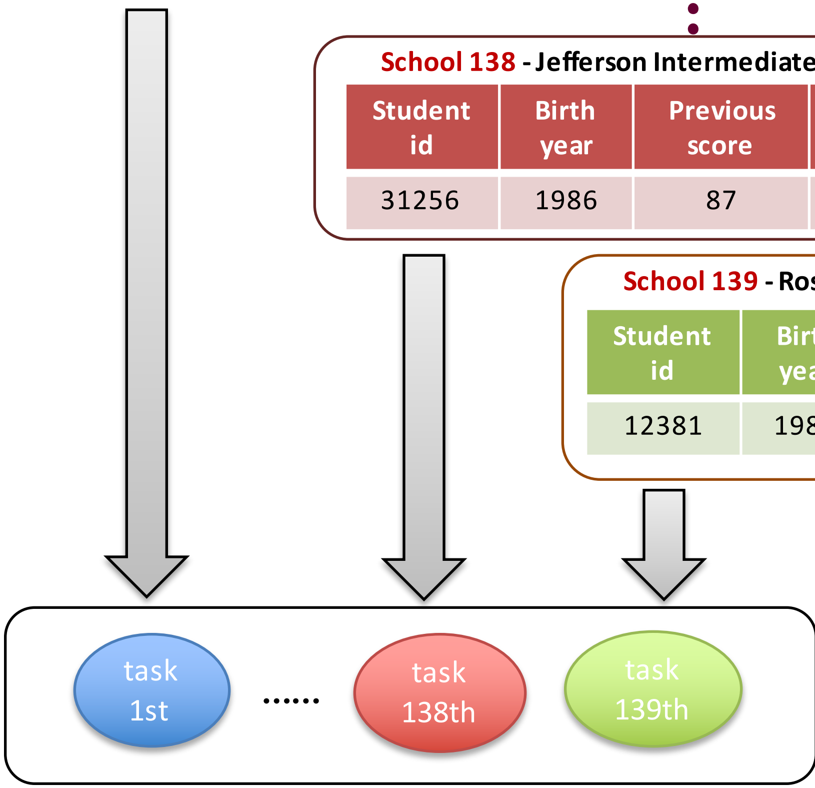
Student id	Birth year	Previous score	School ranking	…	→	Exam Score
72981	1985	95	83%	…		?

School 138 - Jefferson Intermediate School

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

School 139 - Rosemead High School

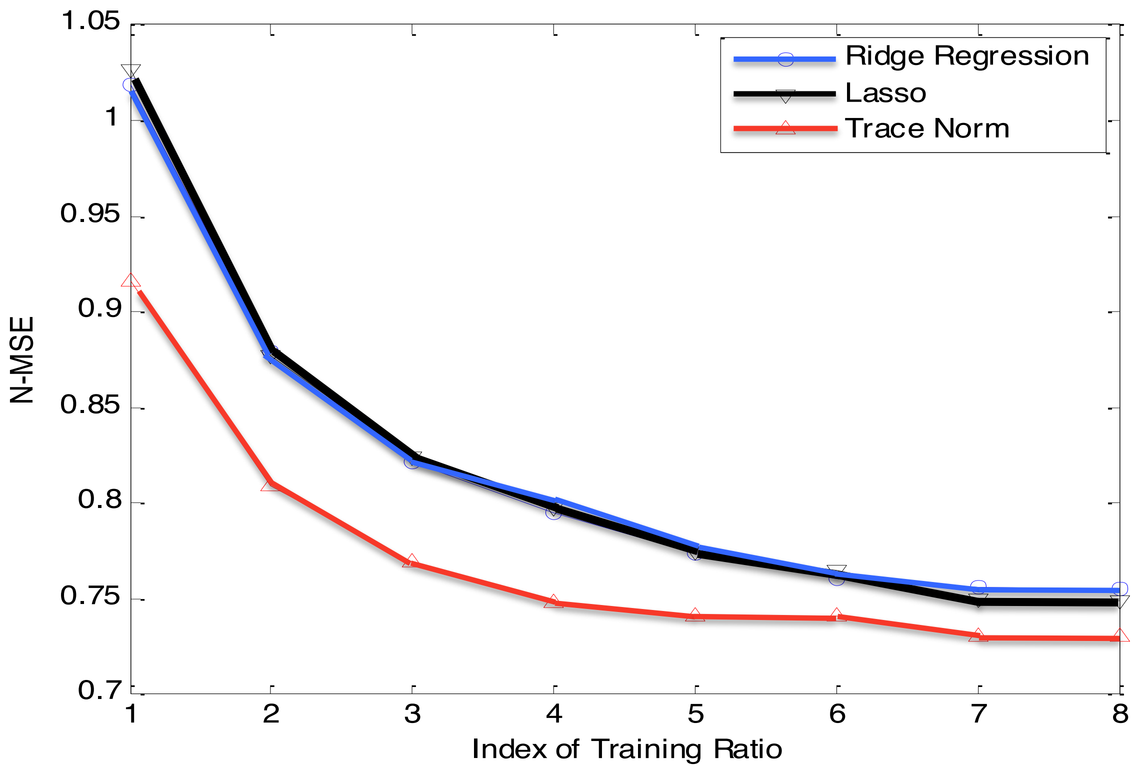
Student id	Birth year	Previous score	School ranking	…	→	Exam Score
12381	1986	83	77%	…		?



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

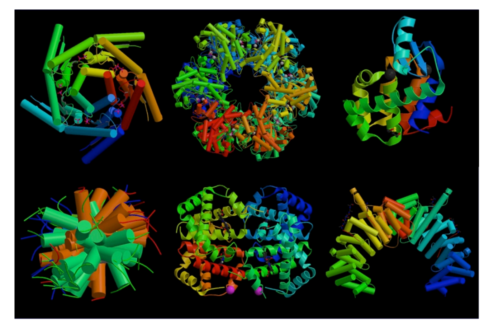
HIV Therapy Screening *[Bickel, ICML'08]*

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]

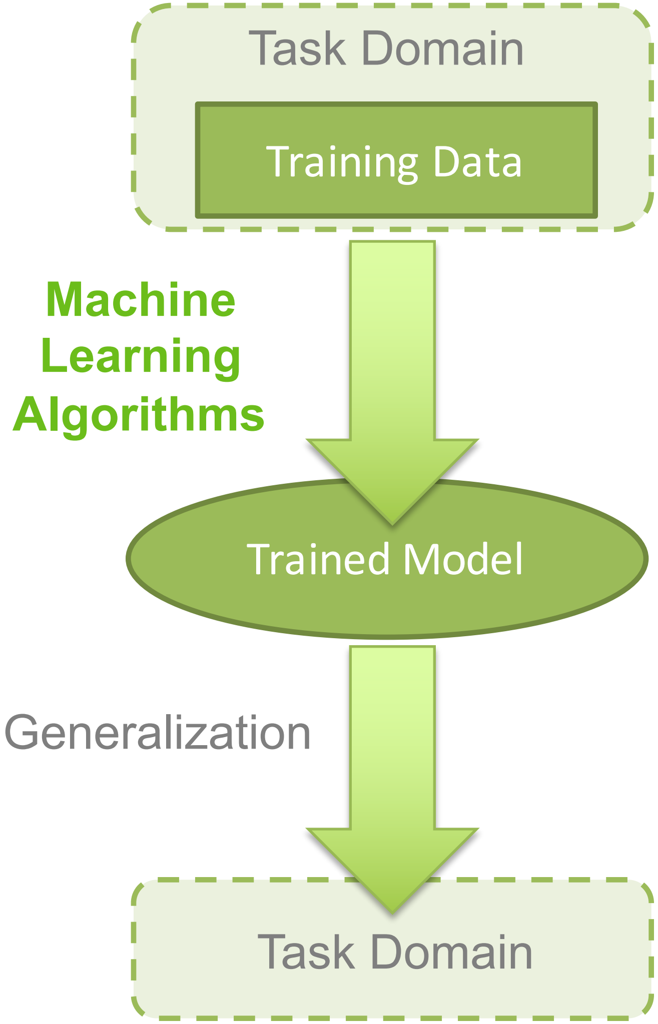
Disease prediction
[Zhang et. al. NeuroImage 12]



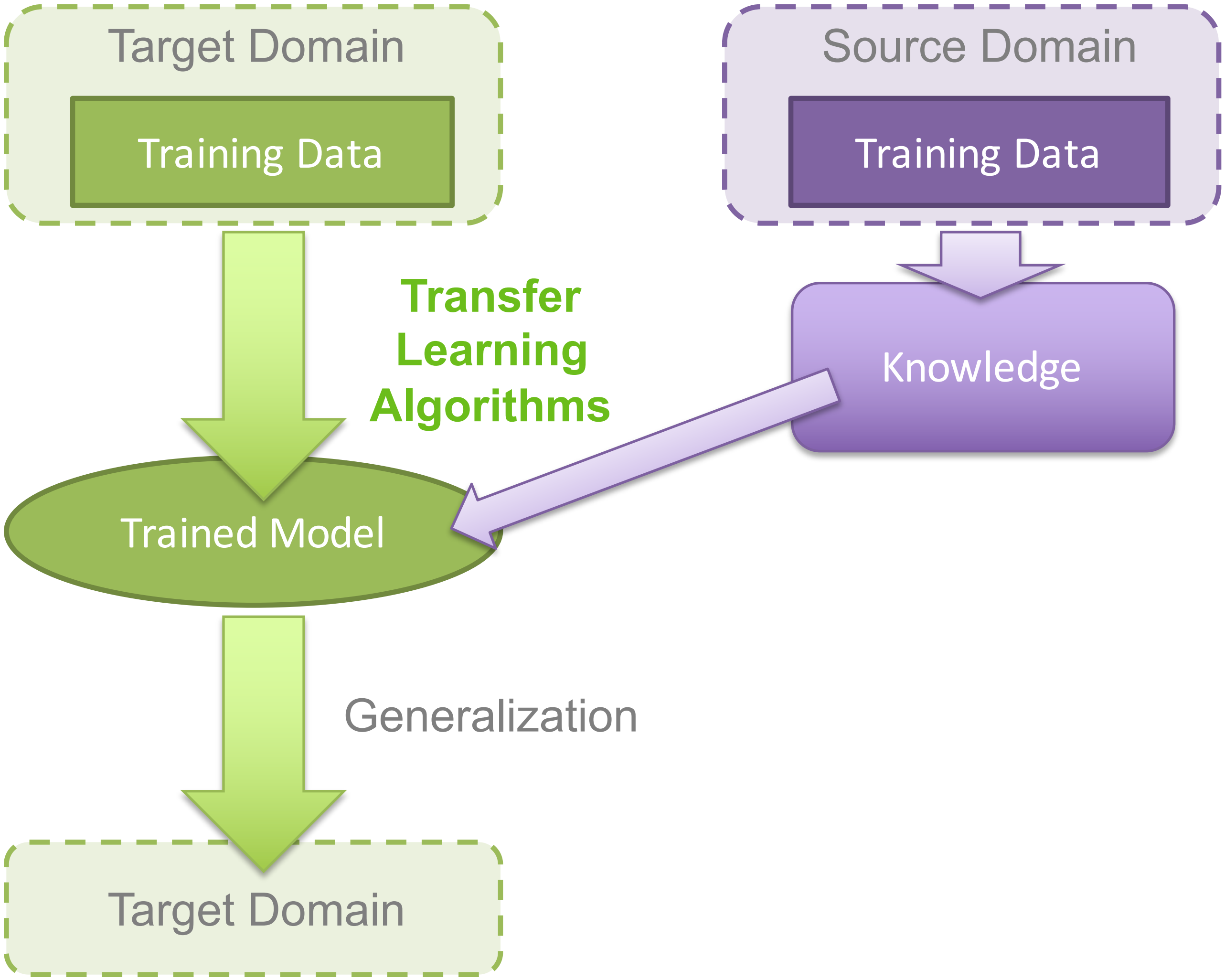
Protein classification
[Charuvaka et. al. ICDM'12]

Traditional Machine Learning

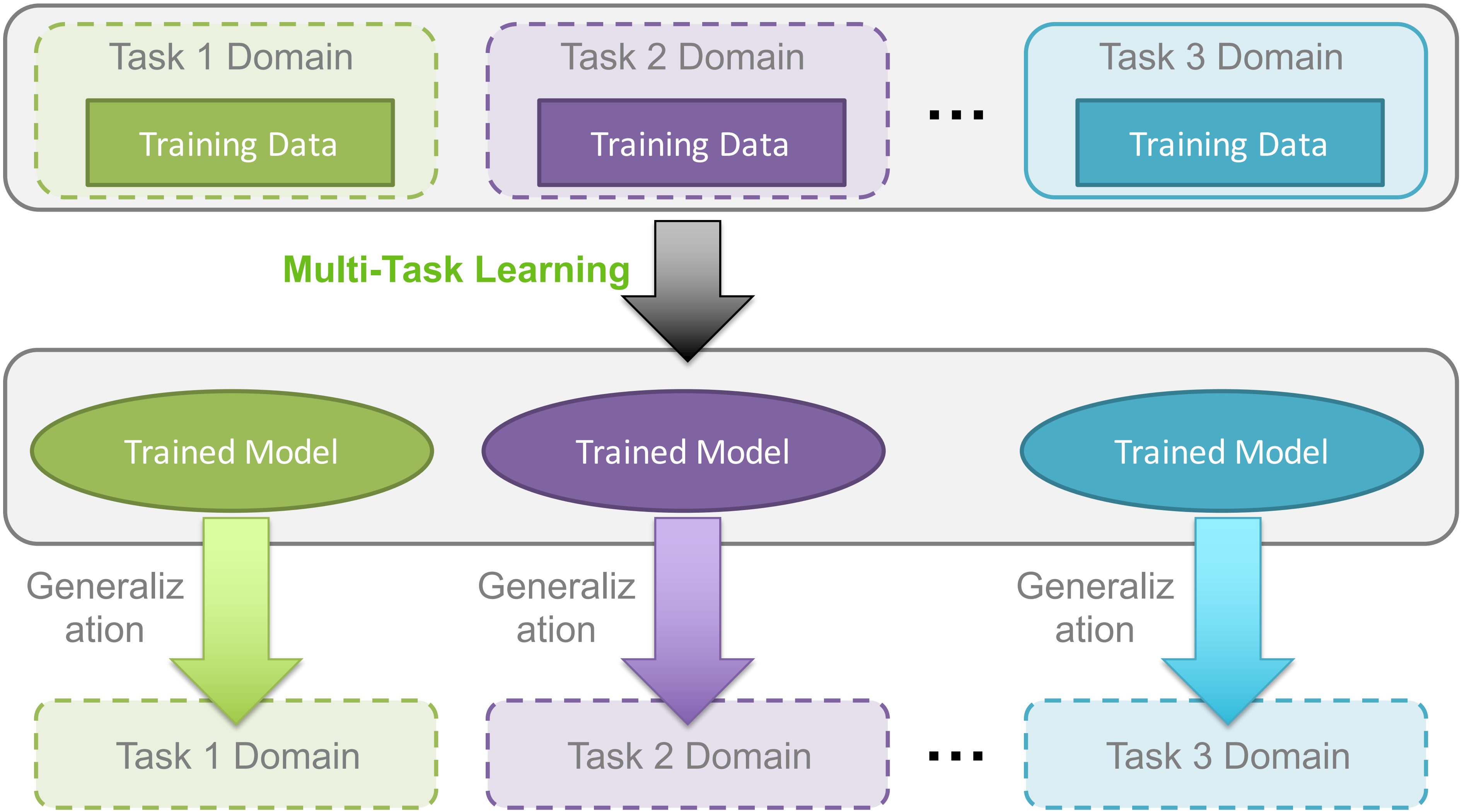
- Elements of machine learning on single task
  - The problem (task/domain)
  - Training data
  - Learning algorithms
  - Trained model
  - Applying model on unseen data (generalization)



Transfer Learning

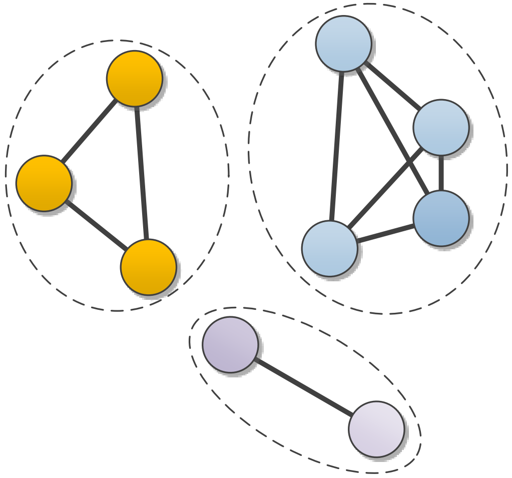


Multi-Task Learning



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

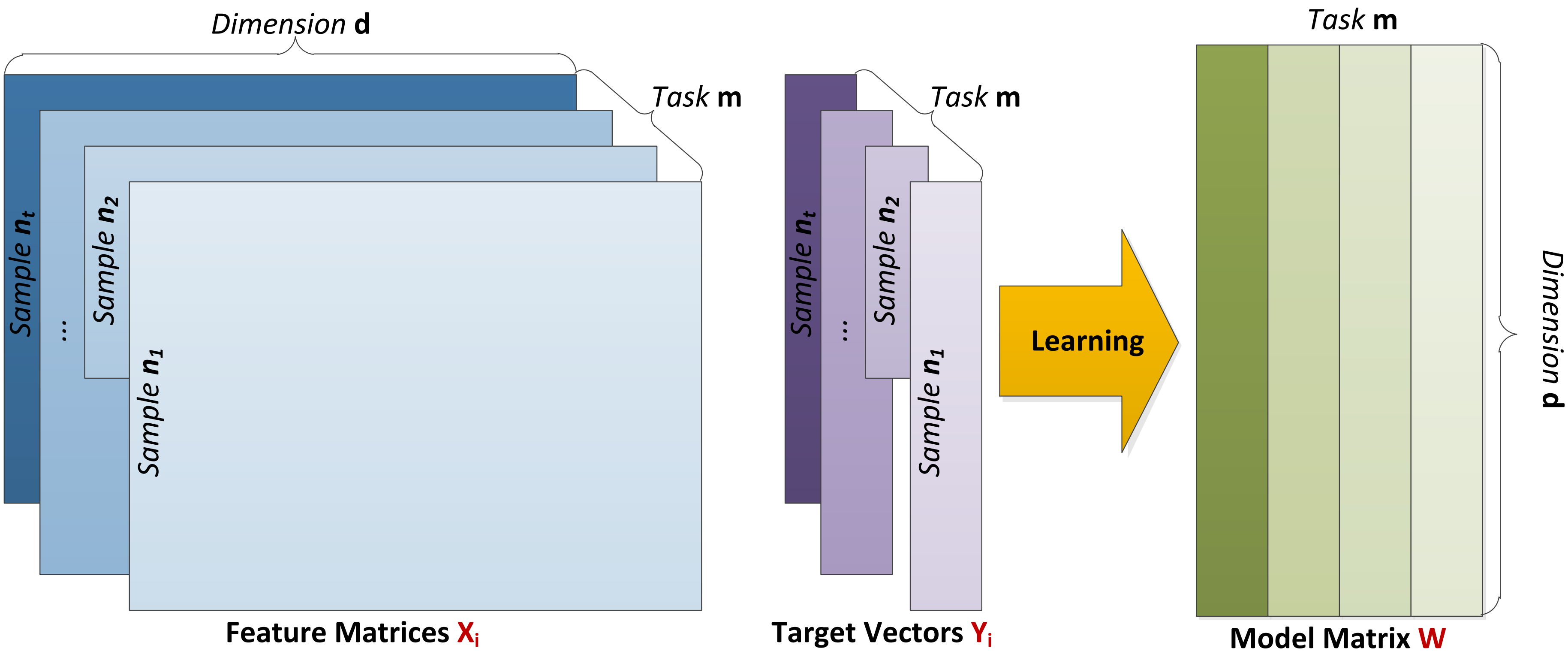
Achieve Multi-Task Learning

- 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

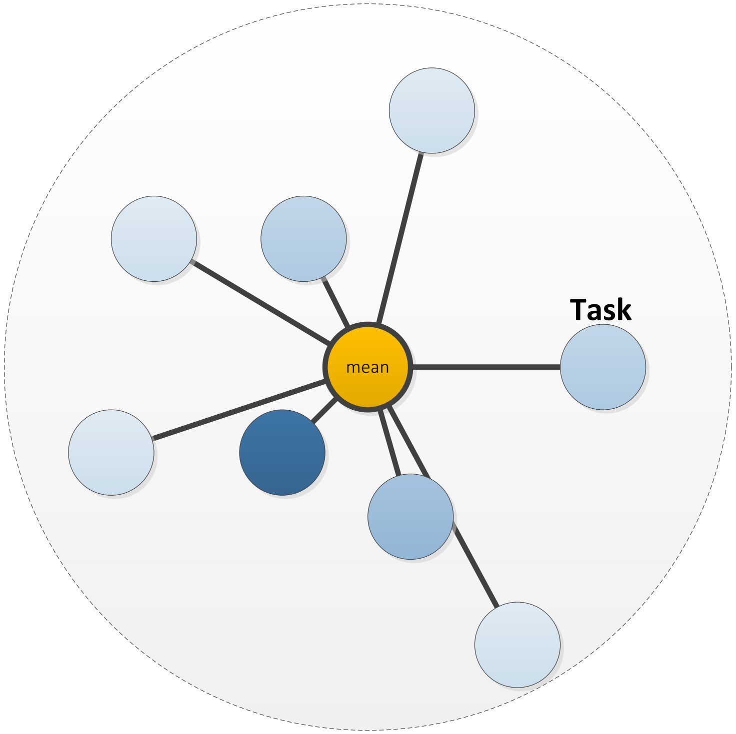
Evgeniou & Pontil, 2004 KDD

- 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

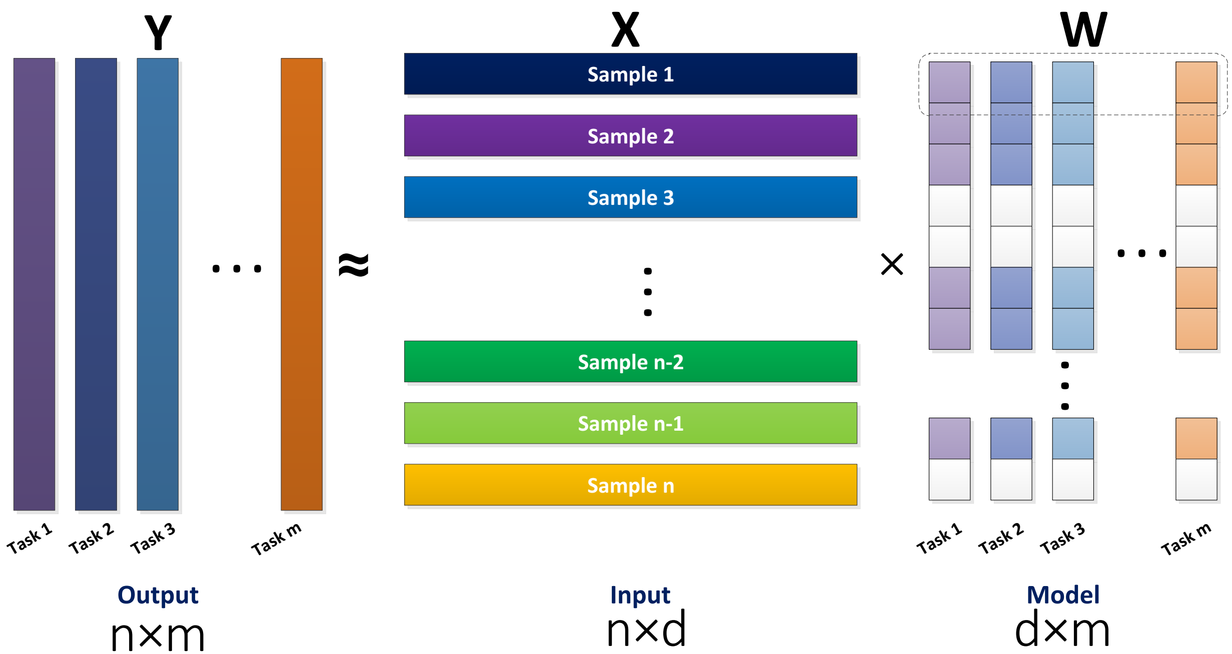
$$\min_{W} \frac{1}{2}\|XW - Y\|_F^2 + \lambda \sum_{i=1}^{m} \left\| W_i - \frac{1}{m}\sum_{s=1}^{m} W_s \right\|_2^2$$



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.



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

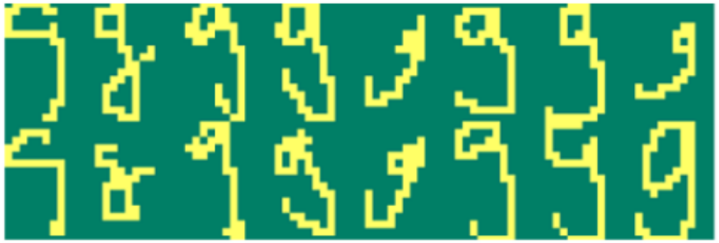
Regularization

Encourages group sparsity

Writer-Specific Character Recognition

Obozinski, Taskar, and Jordan, 2006

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

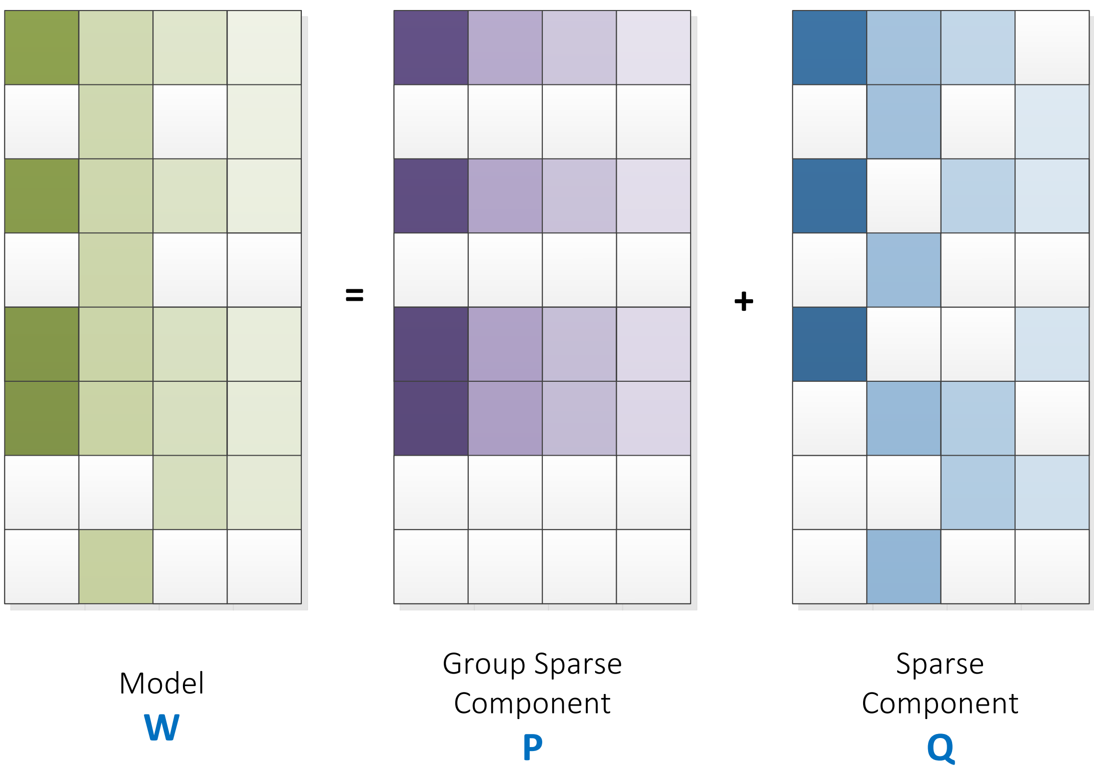


	pixels: error (%)			
Task	ℓ_1/ℓ_2	ℓ_1/ℓ_1	id.ℓ_1	pool
c/e	**4.0**	8.5	9.0	4.5
g/y	**11.4**	16.1	17.2	18.6
g/s	**4.4**	10.0	10.3	6.9
m/n	**2.5**	6.3	6.9	4.1
a/g	**1.3**	3.6	4.1	3.6
i/j	12.0	14.0	14.0	**11.3**
a/o	**2.8**	4.8	5.2	4.2
f/t	**5.0**	6.7	6.1	8.2
h/n	**3.2**	14.3	18.6	5.0

Dirty Model for Multi-Task Learning

Jalali *et. al.* 2010 NIPS

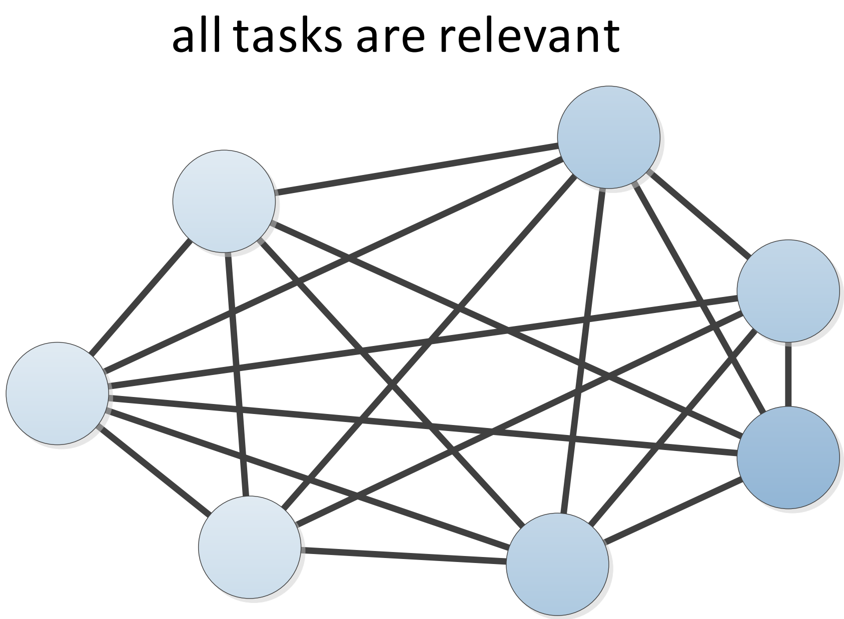
- In practical applications, it is too restrictive to constrain all tasks to share a single shared structure.



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

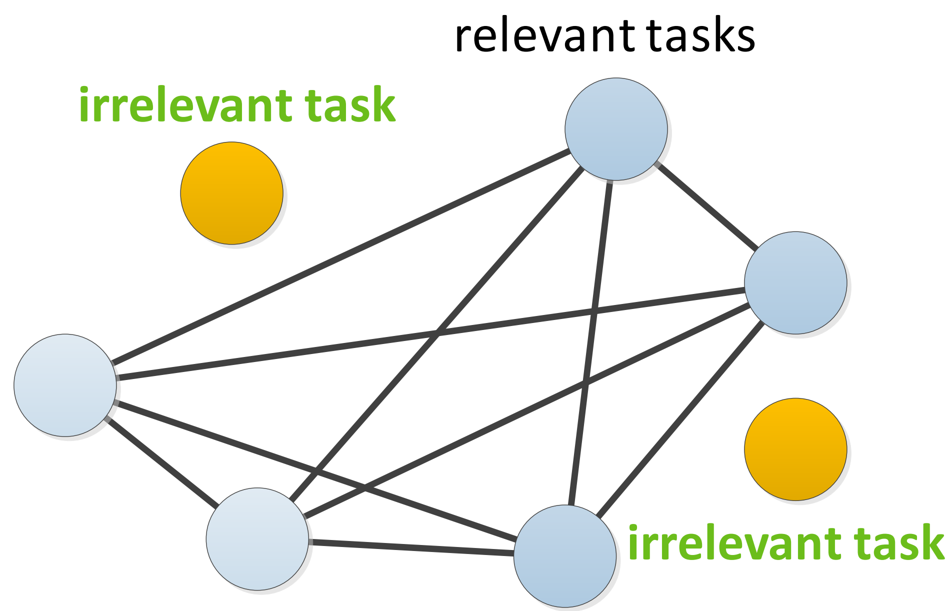
Robust Multi-Task Learning

- Most Existing MTL Approaches



Assumption:
All tasks are related

- Robust MTL Approaches

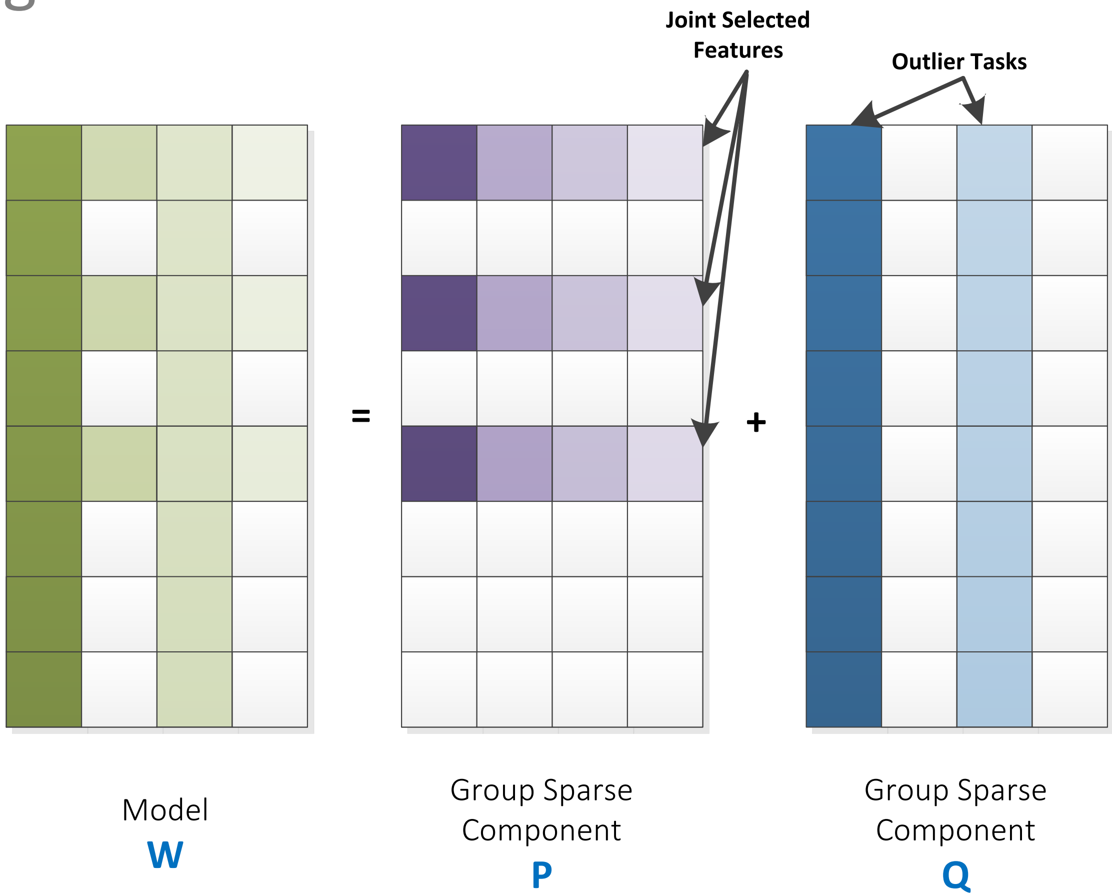


Assumption:
There are outlier tasks

Robust Multi-Task Feature Learning

Gong *et. al.* 2012 KDD

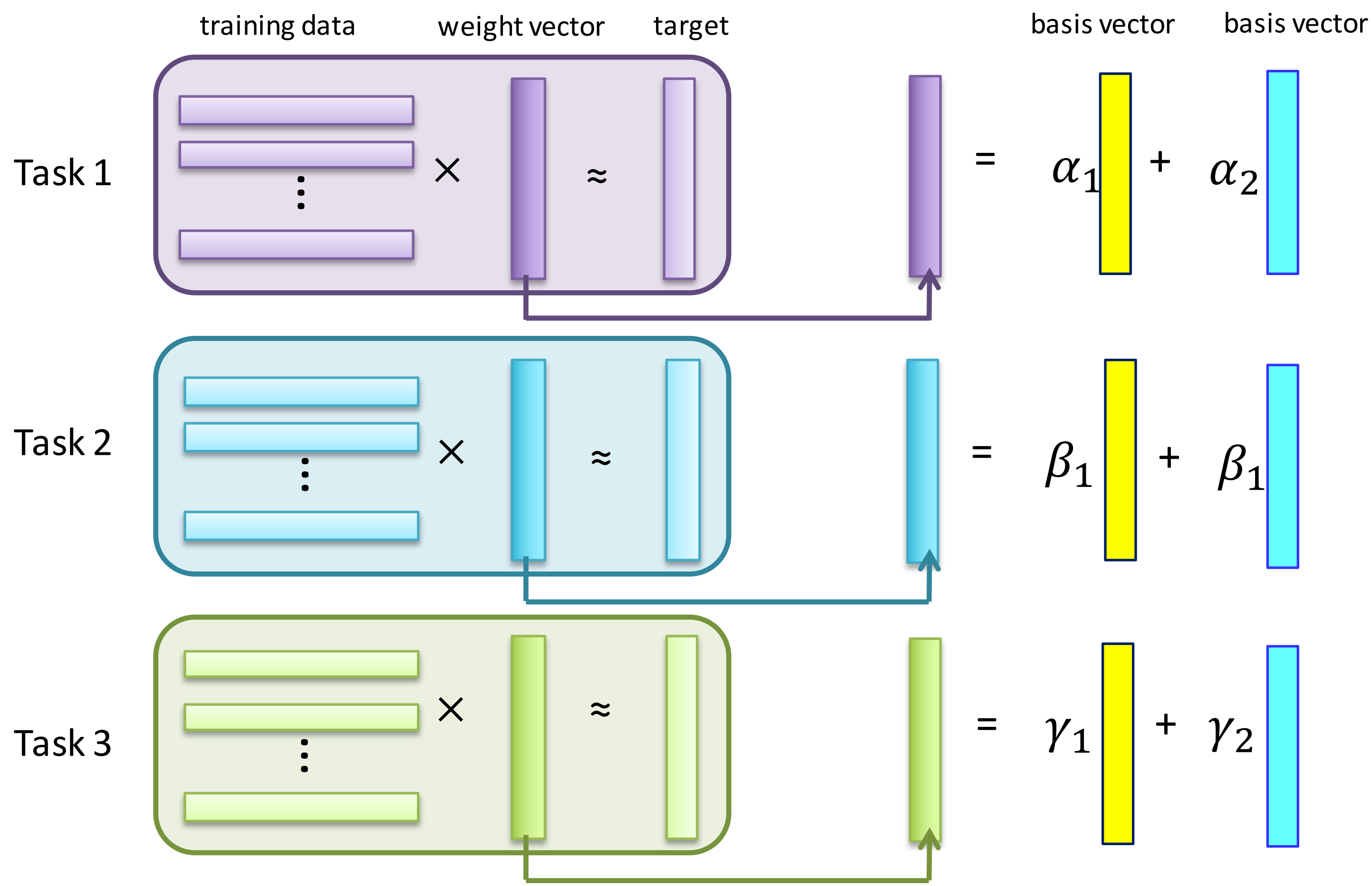
- Simultaneously captures a common set of features among relevant tasks and identifies outlier tasks.



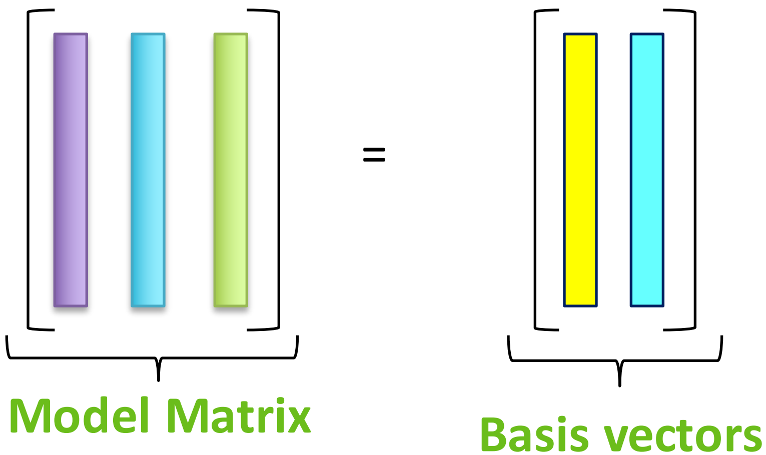
$$\min_{P,Q} \|Y - X(P+Q)\|_F^2 + \lambda_1 \|P\|_{1,q} + \lambda_2 \|Q^T\|_{1,q}$$

Low-Rank Structure for MTL

- Capture task relatedness via a shared low-rank structure



Low-Rank Structure for MTL (Cont.)



$$\times \begin{bmatrix} \alpha_1 & \alpha_2 \\ \beta_1 & \beta_2 \\ \gamma_1 & \gamma_2 \end{bmatrix}^T$$

Coefficients

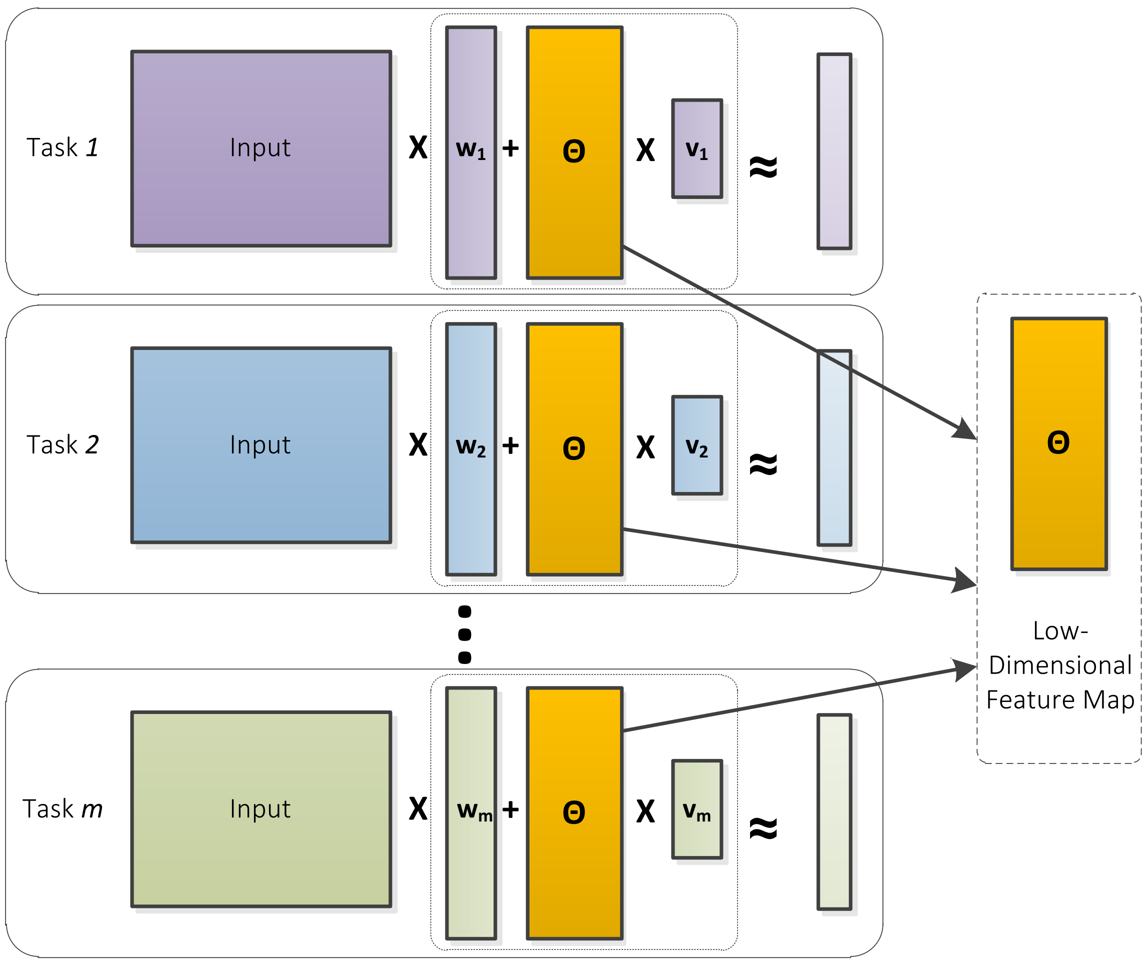
- Rank minimization formulation
  - $\min_W \text{Loss}(W) + \lambda \times \text{Rank}(W)$
- Rank minimization is *NP-Hard* for general loss functions thus we use convex relaxation: trace norm minimization
  - $\min_W \text{Loss}(W) + \lambda \times \|W\|_*$

Regularization
Encourages low-rank on the model matrix

Alternating Structure Optimization (ASO)

Ando and Zhang, 2005 JMRL

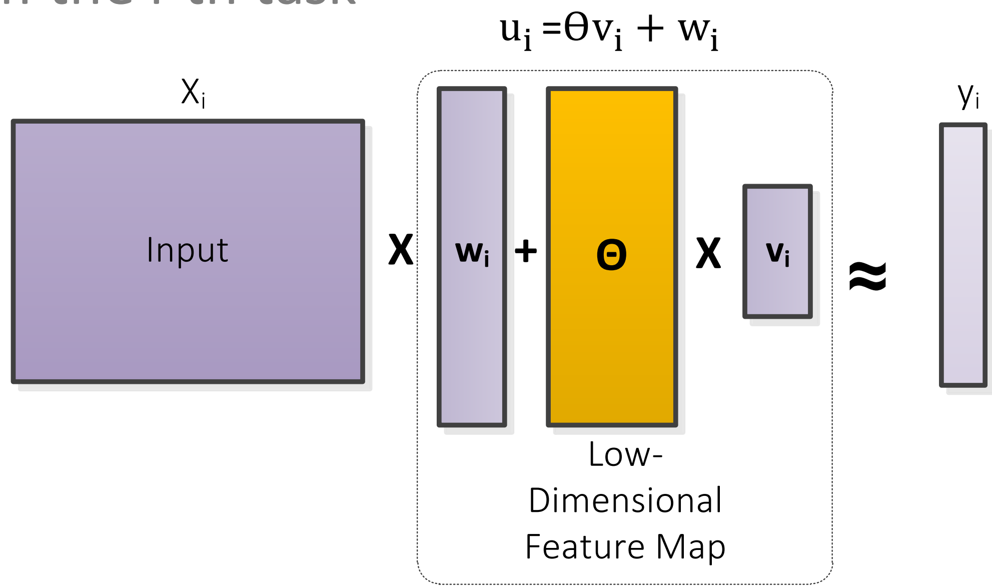
- ASO assumes that the model is the sum of two components: a task specific one and a shared low dimensional subspace.



Alternating Structure Optimization (ASO)

Ando and Zhang, 2005 JMRL

- Learning from the i-th task



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

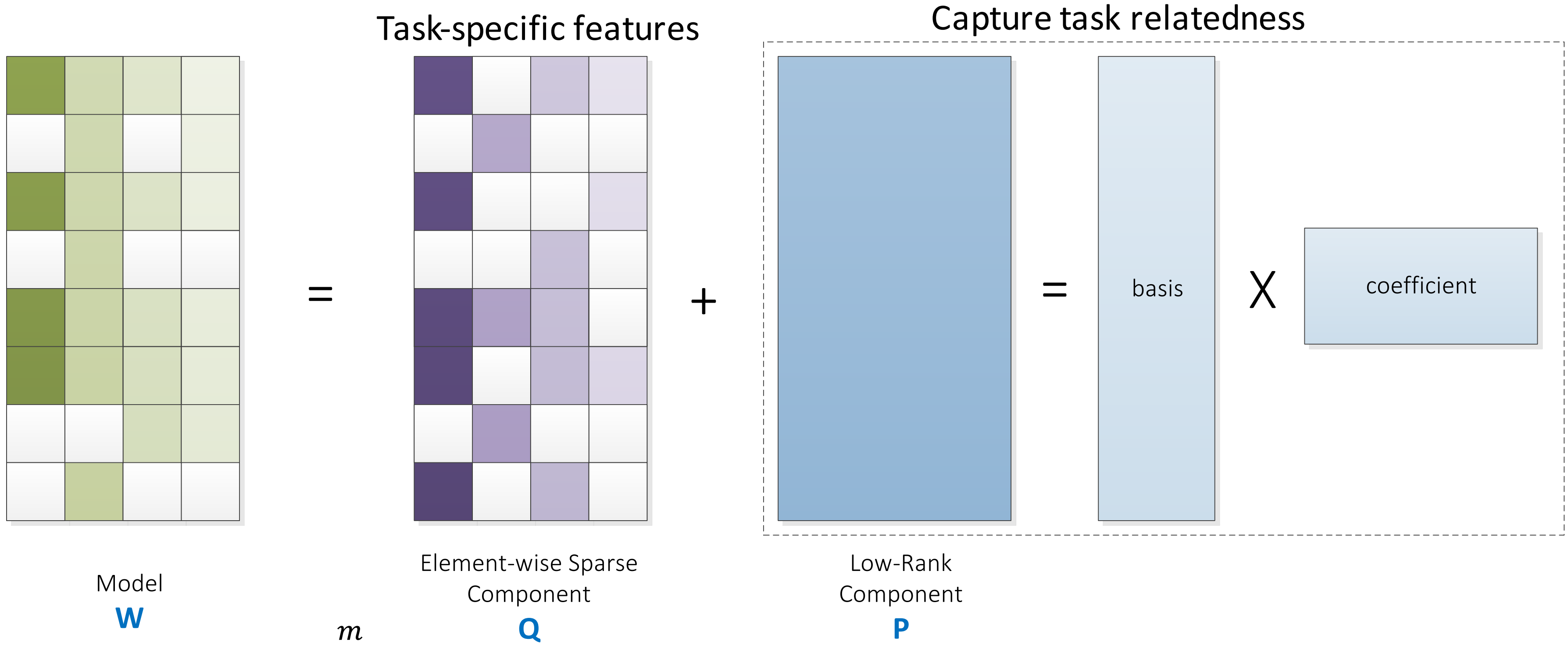
subject to $\Theta^T\Theta = I$

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

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.



$$\min_{P,Q} \sum_{i=1}^{m} \mathcal{L}_i(X_i(P_i + Q_i), y_i) + \lambda \|Q\|_1$$

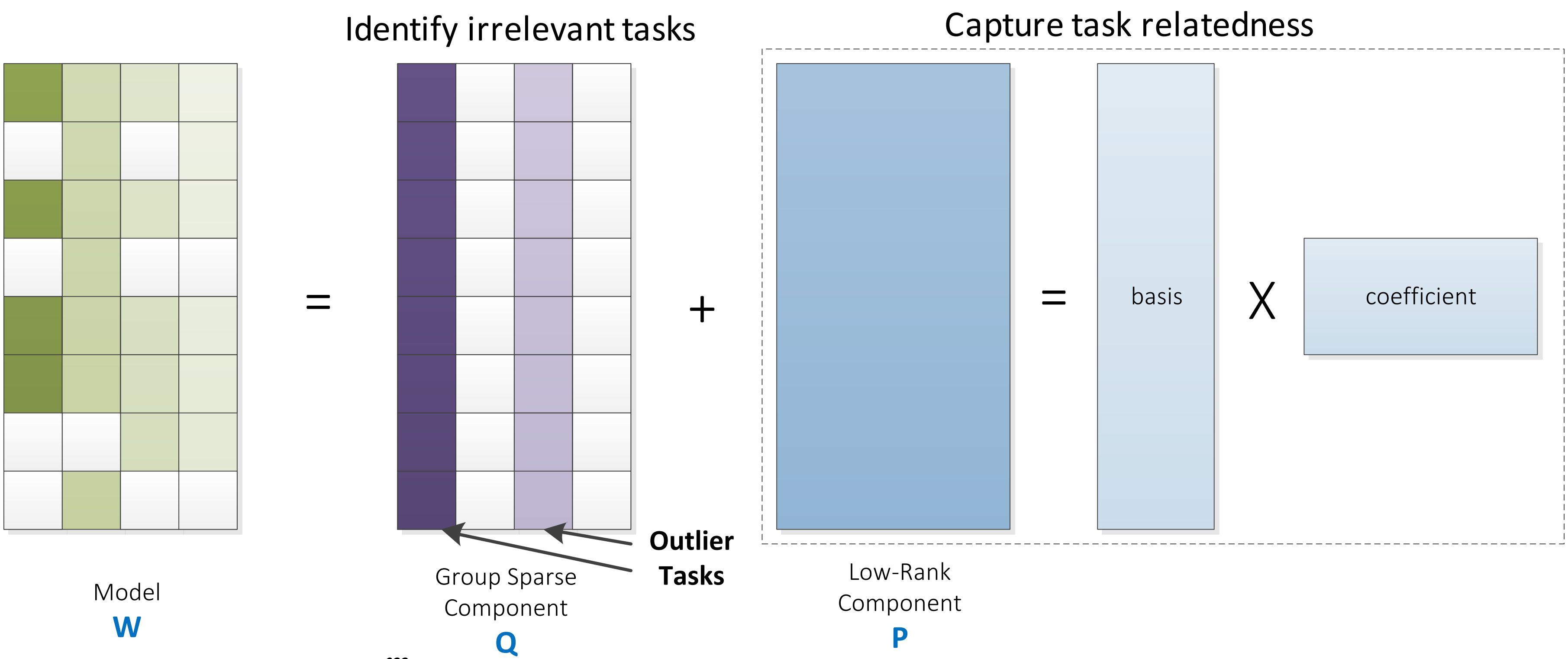
subject to $\|P\|_* \leqslant \eta$

Convex formulation

Robust Low-Rank in MTL

Chen *et. al.* 2011 KDD

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



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

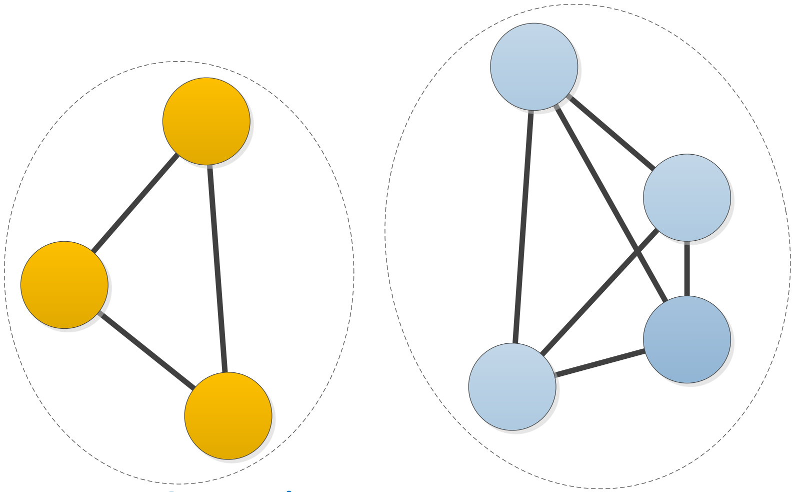
Summary

- 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

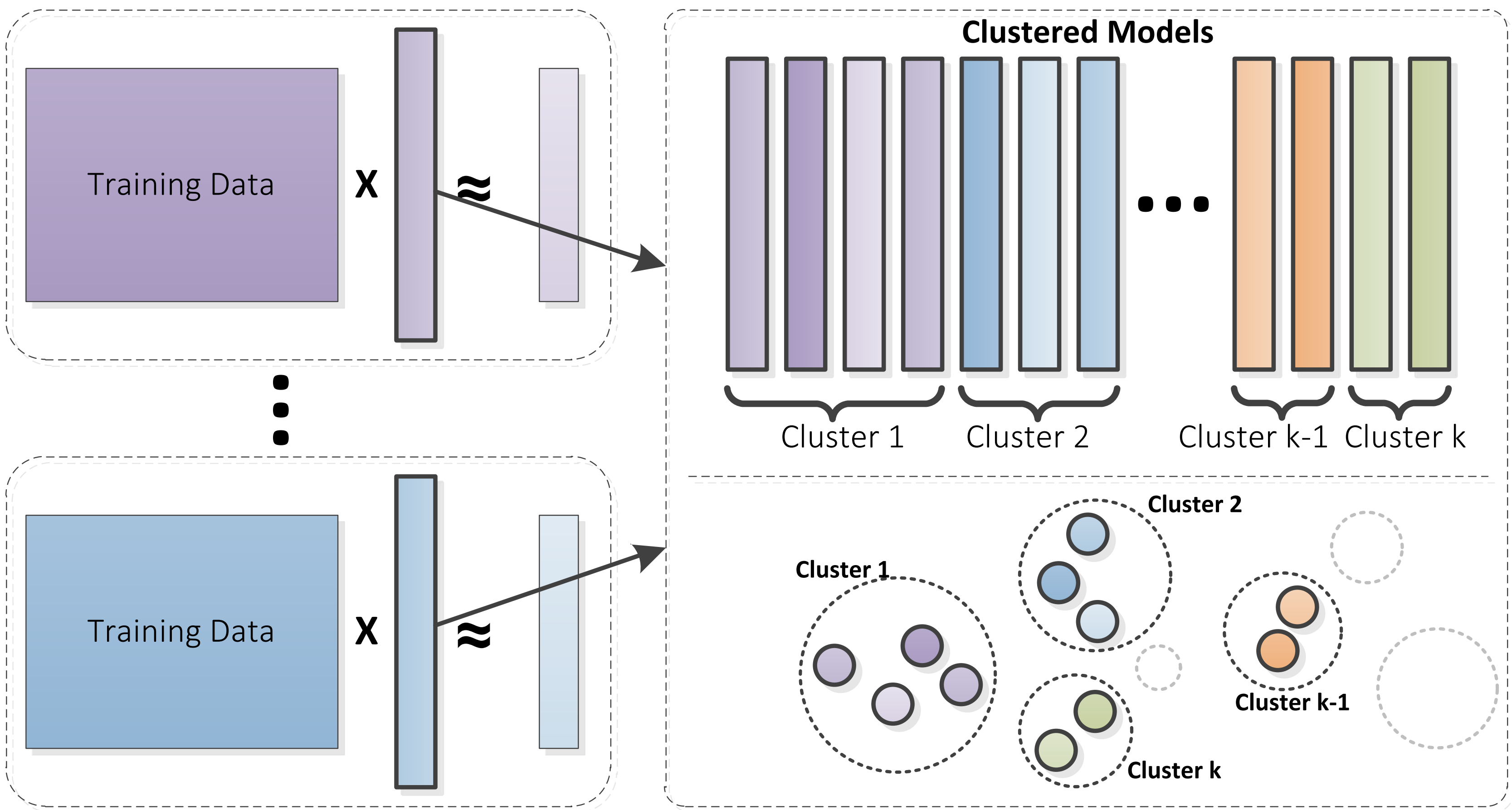


Assumption:
Tasks have group structures

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:

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

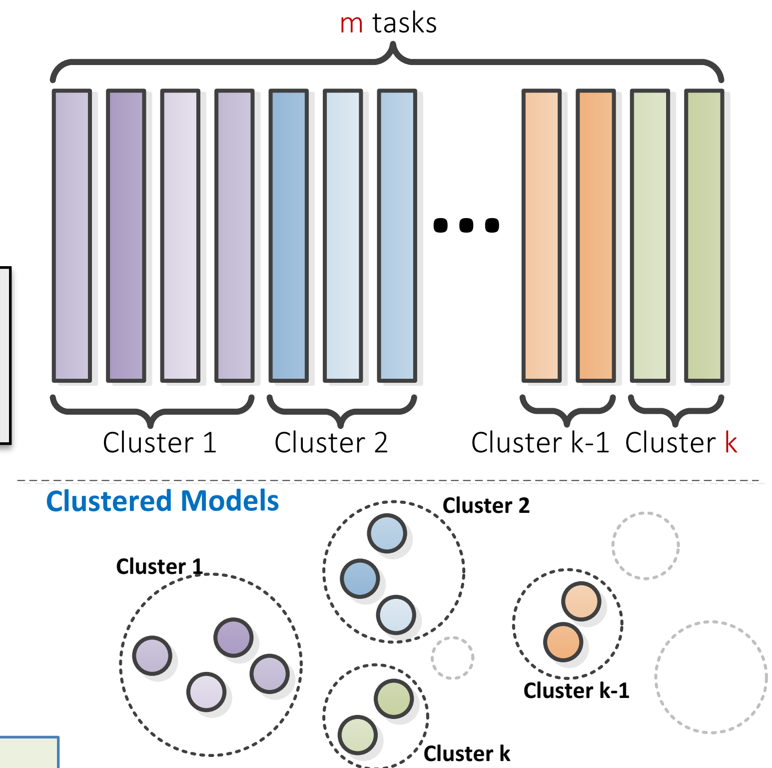
I_j index set of j^{th} cluster

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



task number m > cluster number k

Clustered Multi-Task Learning

Zhou *et. al.* 2011 NIPS

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

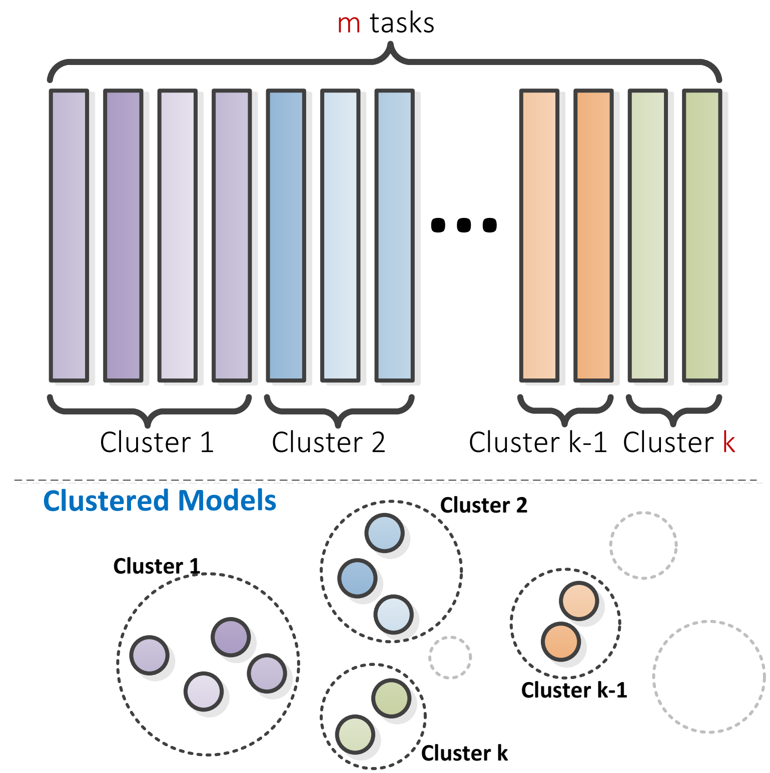
$$\min_{F} \operatorname{tr}(W^T W) - \operatorname{tr}(F^T W^T W F)$$

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

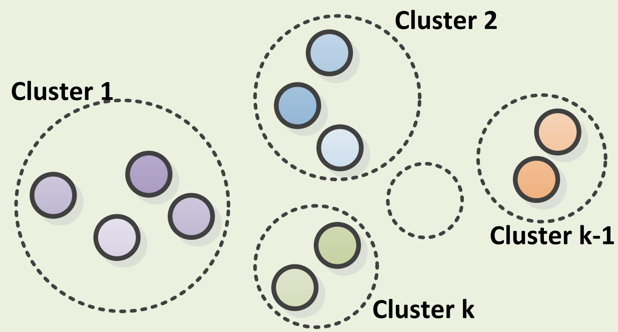
Clustered Multi-Task Learning

Zhou *et. al.* 2011 NIPS

- Clustered multi-task learning (CMTL) formulation

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

capture cluster structures

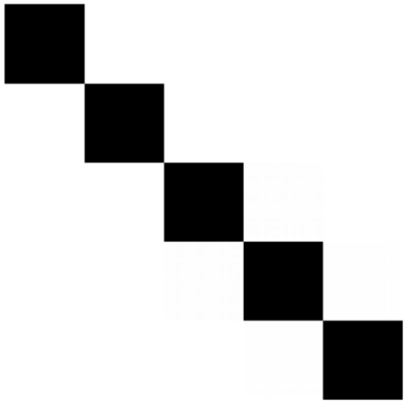


Improves generalization performance

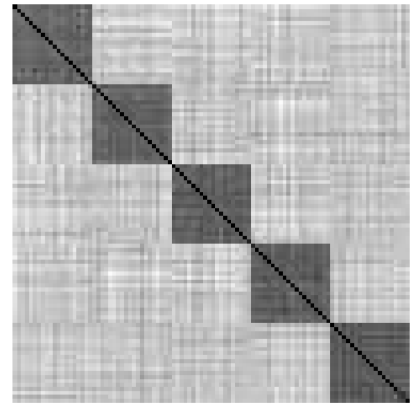
- CMTL has been shown to be equivalent to another class of MTL called ASO
  - Given the dimension of the shared low-rank subspace in ASO and the cluster number in clustered multi-task learning (CMTL) are the same.

Convex Clustered Multi-Task Learning

Zhou *et. al.* 2011 NIPS



Ground Truth

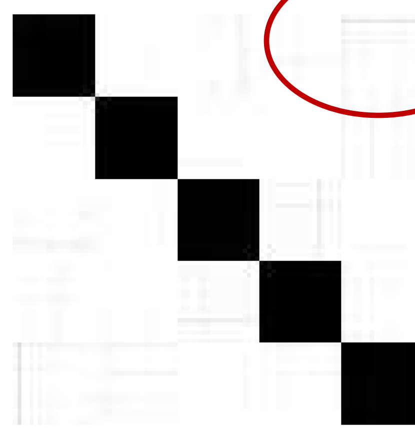


Mean Regularized MTL

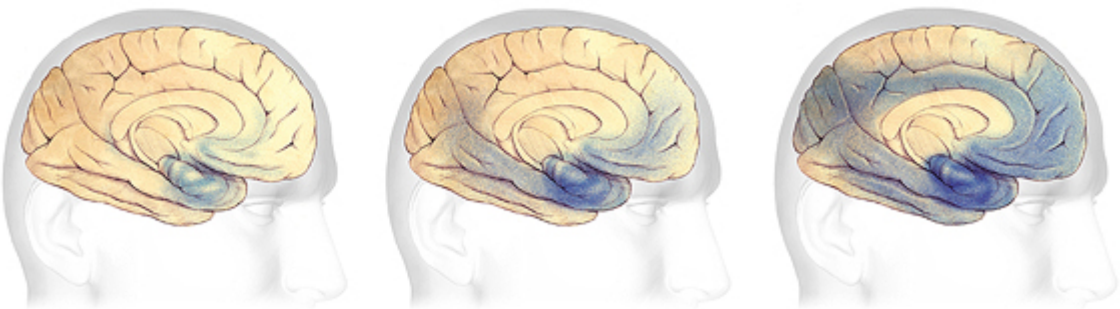
noise introduced by relaxations

Low rank can also well capture cluster structure

Trace Norm Regularized MTL



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

Alzheimer's disease

Also called: senile dementia

ABOUT SYMPTOMS TREATMENTS

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

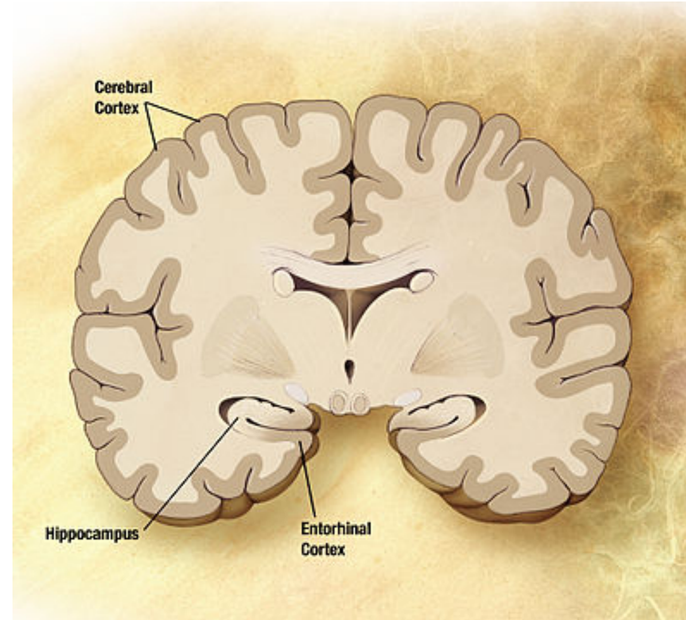
Very common

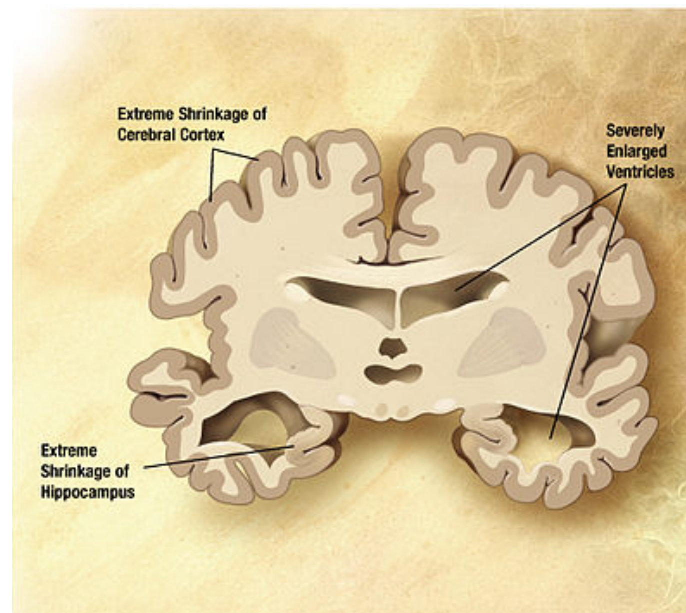
More than **3 million** US cases per year

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

Consult a doctor for medical advice

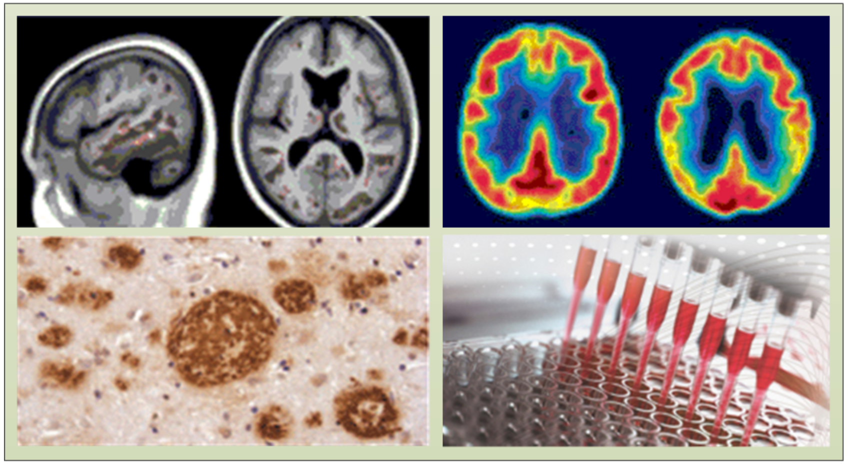
Sources: Mayo Clinic and others.

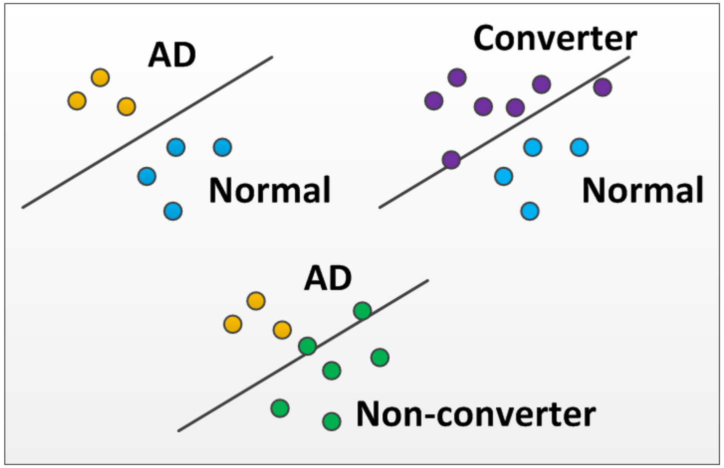




Background (cont.)

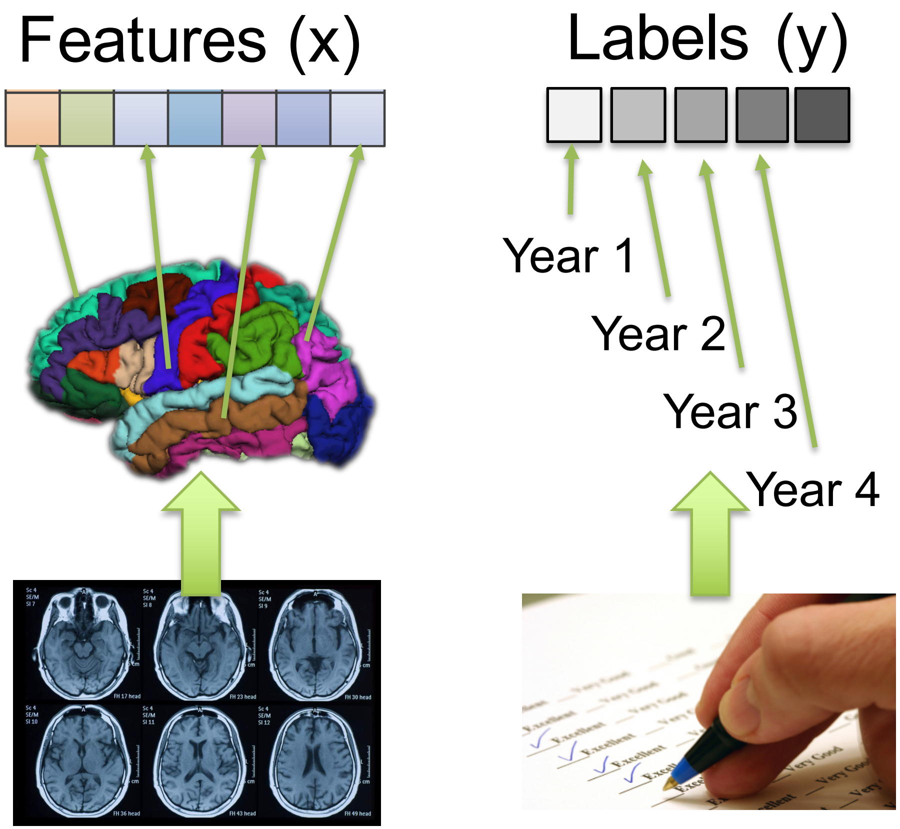
- NIH in 2003 funded the Alzheimer’s Disease Neuroimaging Initiative (ADNI), facilitating a public available database for using neuroimaging data in predicting the progression of AD.





Disease Progression

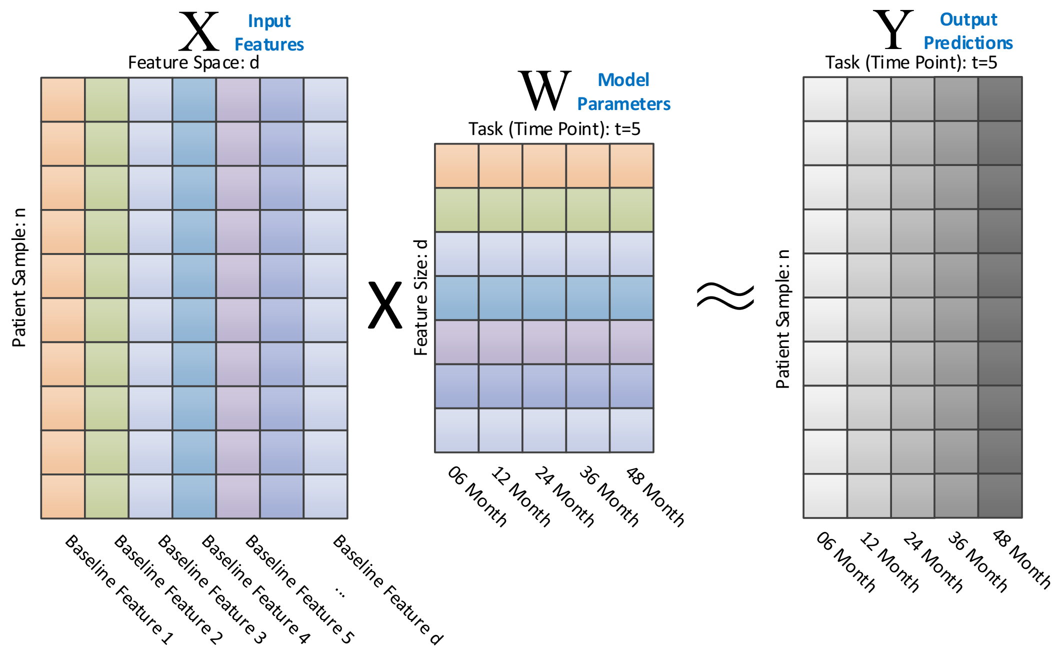
- Clinical scores are used to evaluate the cognitive status
  - MMSE, ADAS-Cog and etc.
- Disease progression
  - Prediction of clinical scores from neuroimaging features
  - Build one regression model at each time point.



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)

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

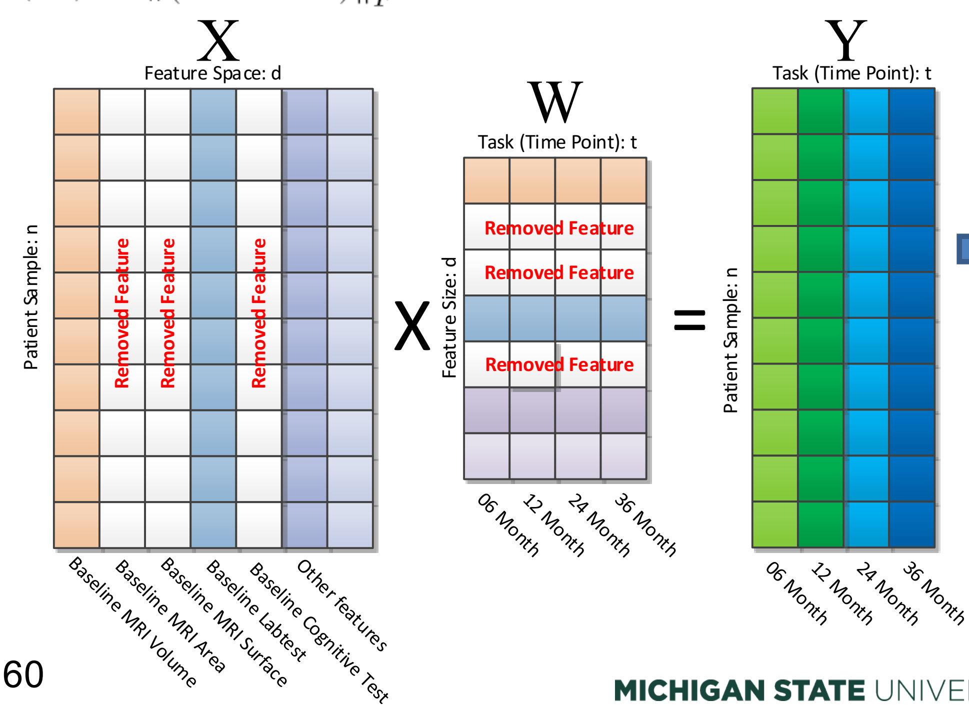
Loss Function
Performs regression

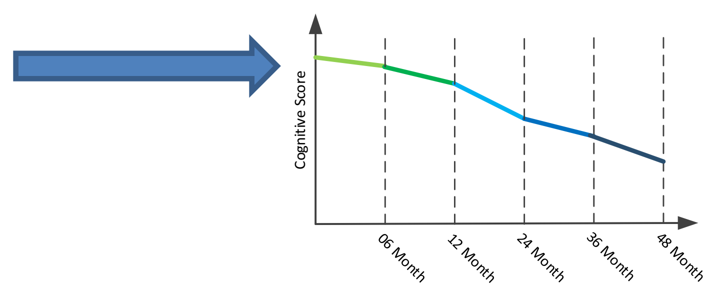
$$L(W) = \|(XW - Y)\|_F^2$$

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)

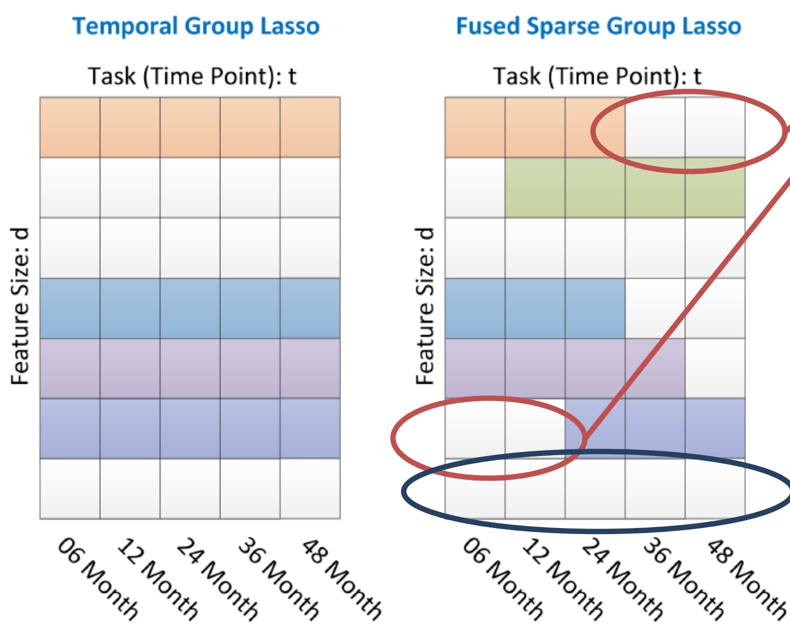
$$\min_{W} \boxed{L(W)} + \boxed{\lambda_1 \|W\|_1} + \boxed{\lambda_2 \left\|RW^T\right\|_1} + \boxed{\lambda_3 \|W\|_{2,1}}$$

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*

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

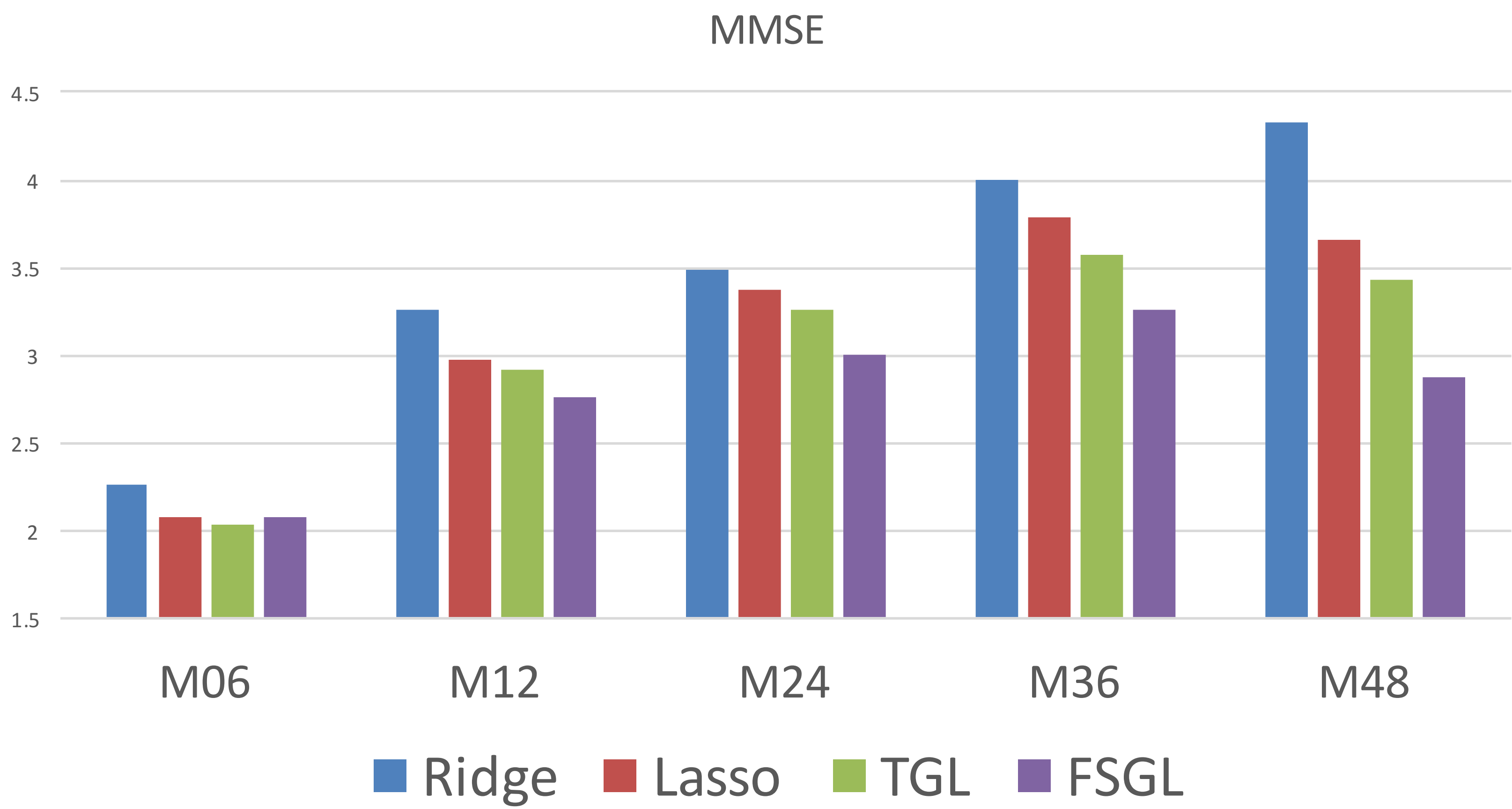
$$\pi_{GL}(\mathbf{v}) = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{v}\|_2^2 + \lambda_3\|\mathbf{w}\|_2. \quad (6)$$

Then the following holds:

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

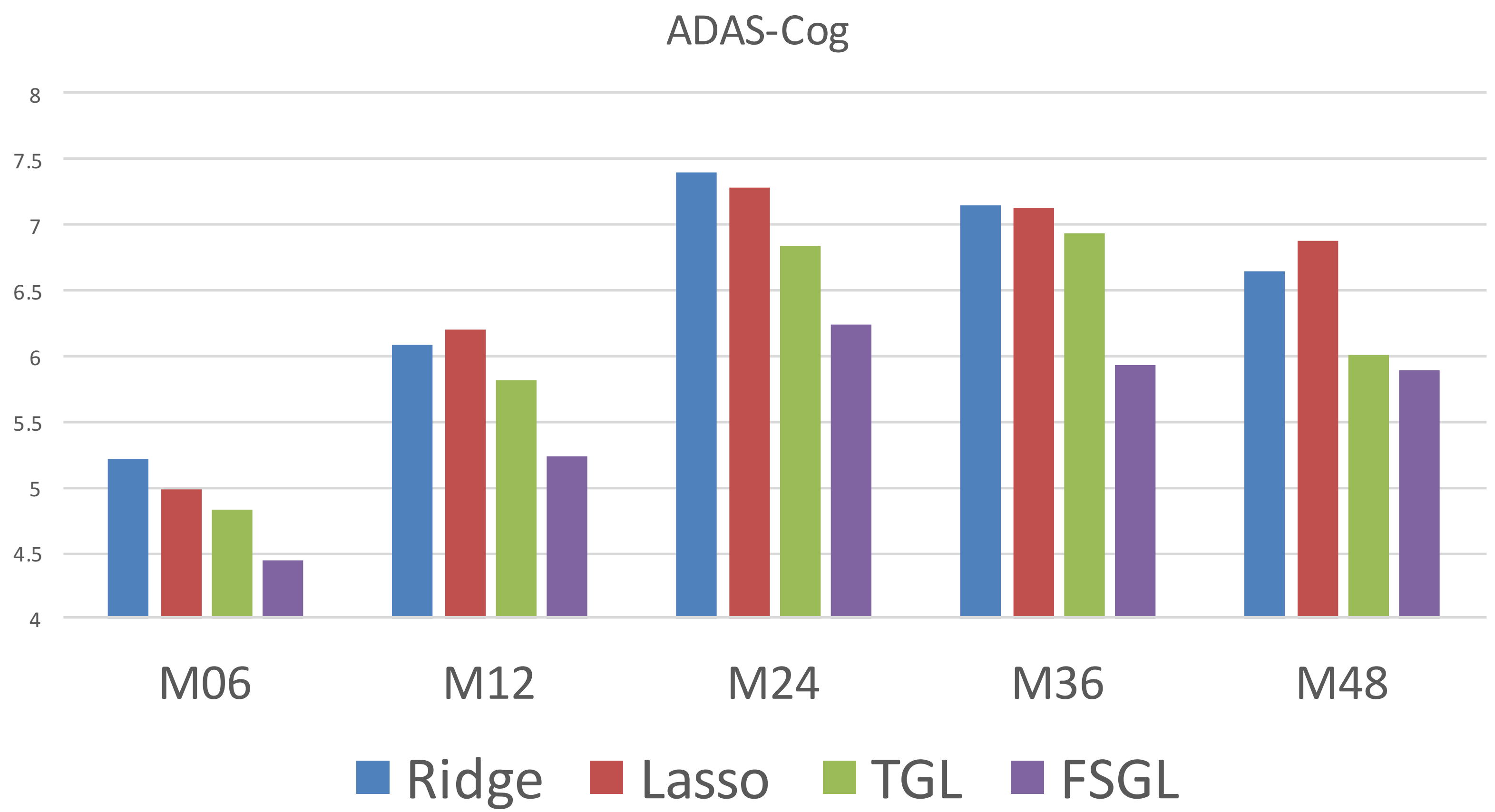
Performance

- 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

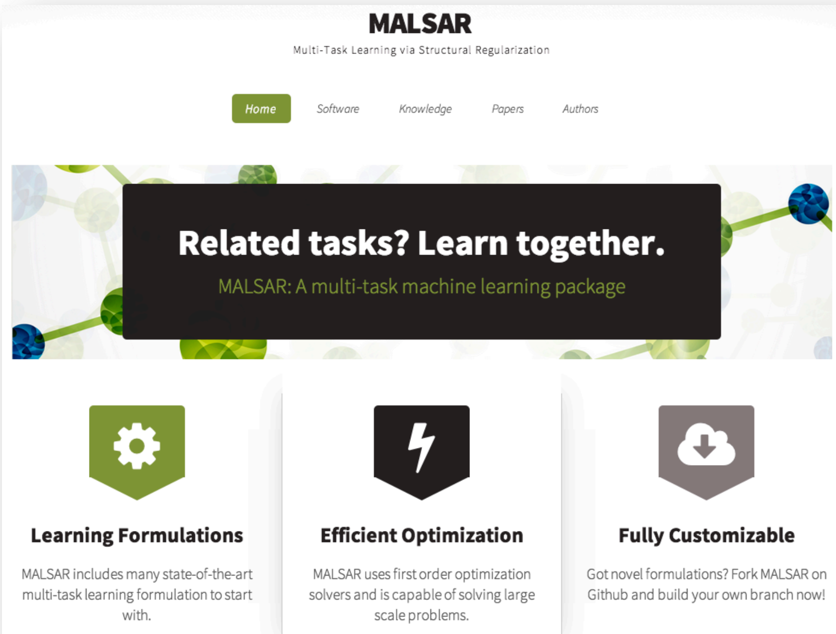


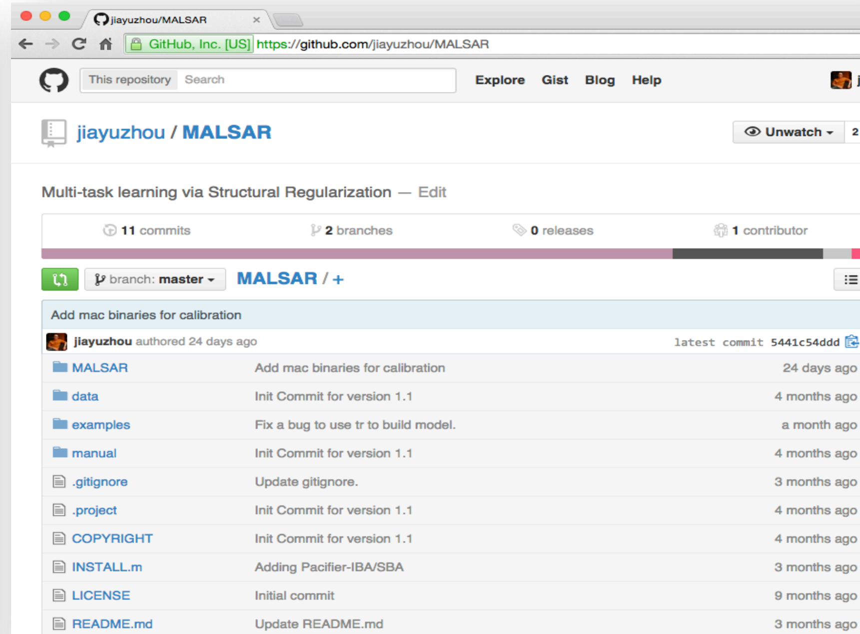
MALSAR: Multi-Task Learning via Structural Regularization

Multi-Task Learning Software

MALSAR

MULTI-TASK LEARNING VIA STRUCTURAL REGULARIZATION

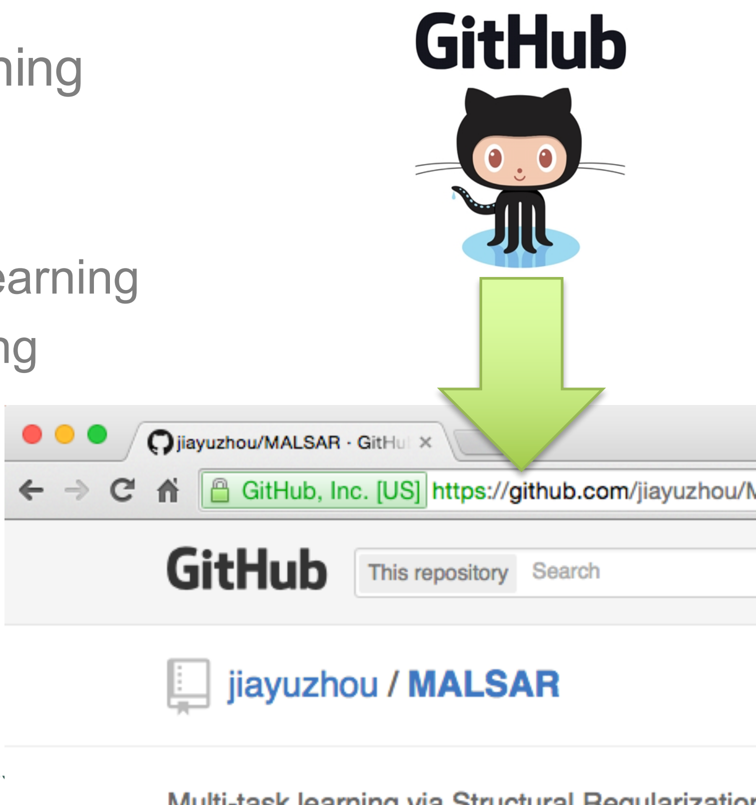




- Firstly introduced my MTL **tutorial** at **SDM** in 2012
- Over 40 research works using MALSAR are published in KDD, NIPS, TPAMI, ICCV, ICDM, ICIP, COLING, MICCAI, ACM-MM, etc.
- Used as **course material** to analyze compound profiling in the *Strasbourg Summer School* in France
- A core component in the $11mi *NIH-BD2K* grant

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

MALSAR/example_Dirty.m

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

Create a random MTL dataset

Invoke an MTL algorithm

```
clear;
clc;
close;

addpath('../MALSAR/functions/dirty/'); % load function
addpath('../MALSAR/c_files/prf_lbm/'); % load projection c libraries.
addpath('../MALSAR/utils/'); % load utilities

%rng('default');     % reset random generator. Available from Matlab 201

%generate synthetic data.
dimension = 500;
sample_size = 50;
task = 50;
X = cell(task ,1);
Y = cell(task ,1);
for i = 1: task
    X{i} = rand(sample_size, dimension);
    Y{i} = rand(sample_size, 1);
end

opts.init = 0;      % guess start point from data.
opts.tFlag = 1;     % terminate after relative objective value does not
opts.tol = 10^-4;   % tolerance.
opts.maxIter = 500; % maximum iteration number of optimization.

rho_1 = 350;%   rho1: group sparsity regularization parameter
rho_2 = 10;%   rho2: elementwise sparsity regularization parameter

[W funcVal P Q] = Least_Dirty(X, Y, rho_1, rho_2, opts);
```

Thanks!