Integrating Low-Rank and Group-Sparse Structures for Robust Multi-Task Learning

CS 584: Big Data Analytics

Material borrowed from SDM tutorial on Multitask learning
(https://www.siam.org/meetings/sdm12/zhou_chen_ye.pdf)

Standard Methodology: Independent Tasks

- Learn one task at a time
- Large problems can be broken into small, reasonably independent subproblems
  - Learn each subproblem separately
  - Merge the results

Motivation for Multi-Task Learning

- How to learn under the scenario of multiple related tasks?
- What if there are few data per task? Can we pool data across related tasks?
- How can we generalize well on given tasks and transfer to new tasks?

Example: Exam Score Prediction

Examination Scores Prediction[1] (Argyriou et. al.'08)

School 1 - Alverno High School

Student id	Birth year	Previous score	…	School ranking	…
72981	1985	95	…	83%	…

student-dependent school-dependent

→

Exam score
?

…

School 138 - Jefferson Intermediate School

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

student-dependent school-dependent

→

Exam score
?

School 139 - Rosemead High School

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

student-dependent school-dependent

→

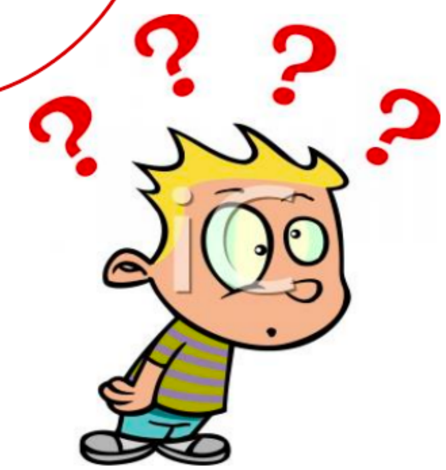
Exam score
?

[1]The Inner London Education Authority (ILEA)

Solution #1: Create Pool of All Tasks

Student id	Birth year	Previous score	School ranking	…	Exam Score
72981	1985	95	83%	…	?
31256	1986	87	72%	…	?
12381	1987	83	77%	…	?
…	…	…	…	…	…
21901	1986	87	72%	…	?

Students with **same Features** but **different Exam Scores**



School A

School B

Solution #2: Learn 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

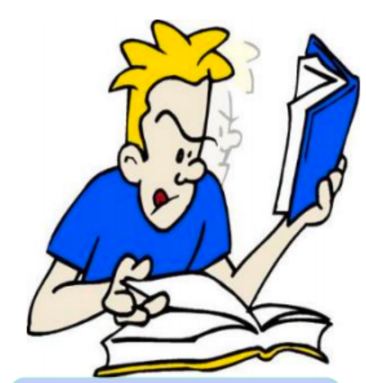
School 138 - Jefferson Intermediate School

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

Solution #3: Learn Multiple Tasks Simultaneously

School 1 - Alverno High School

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

task 1st

…

School 138 - Jefferson Intermediate School

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

task 138th

School 139 - Rosemead High School

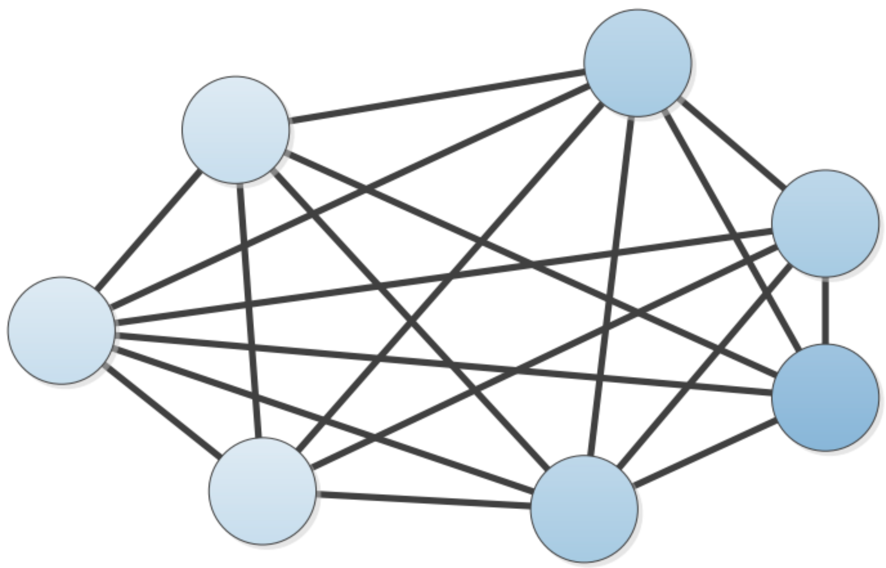
Student id	Birth year	Previous score	School ranking	…	Exam Score
12381	1986	83	77%	…	?

task 139th

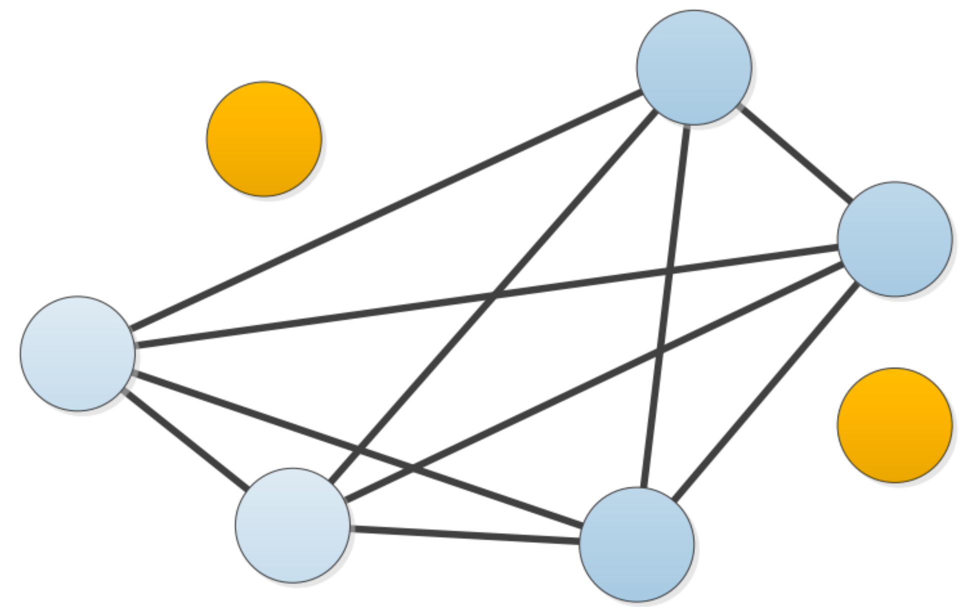
Learn tasks simultaneously
Model the tasks relationship

……

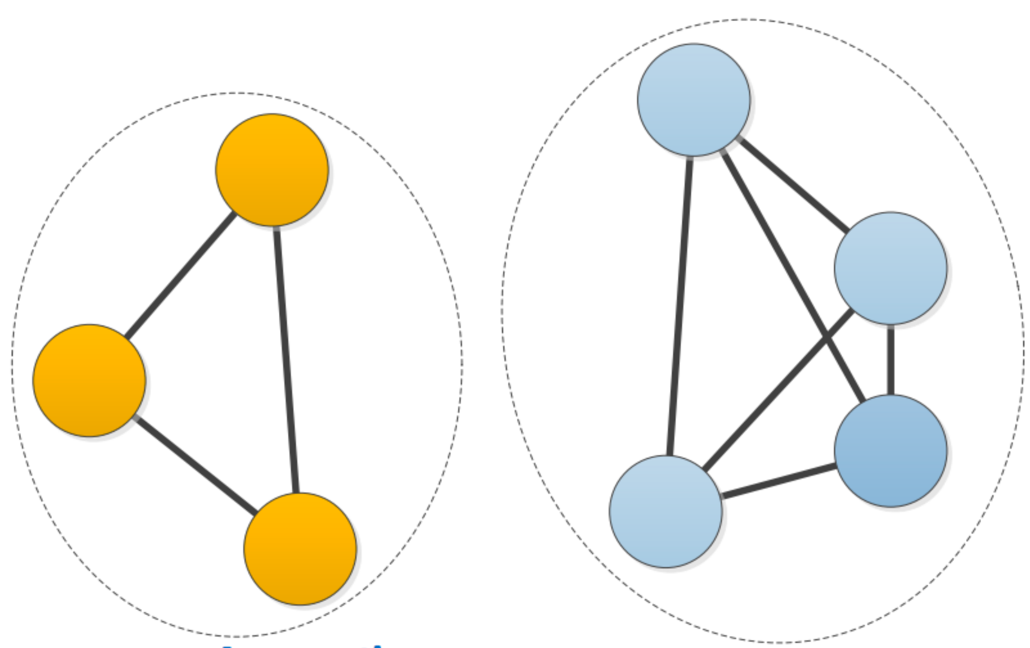
Capturing Shared Structures



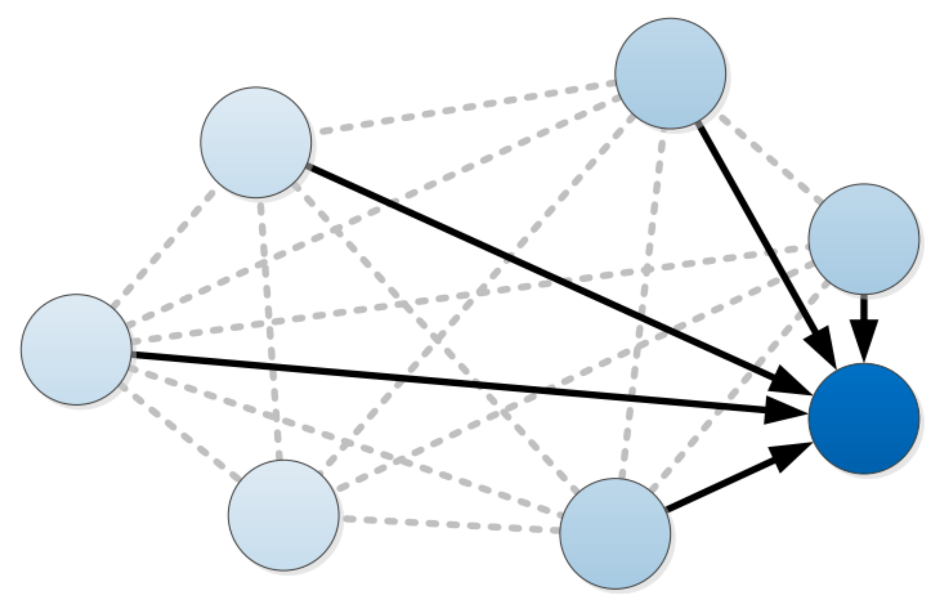
Assumption:
All tasks are related



Assumption:
There are outlier tasks



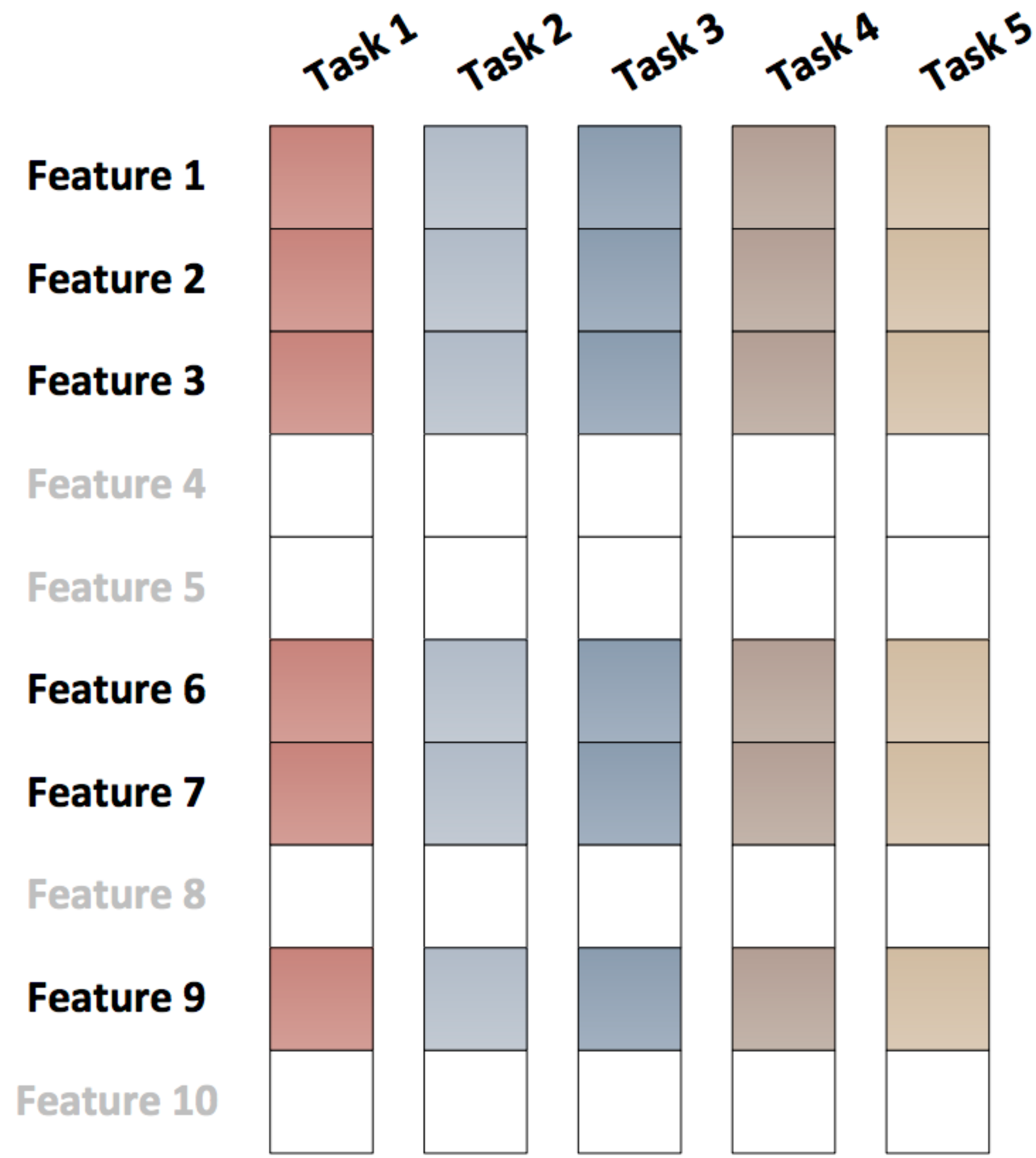
Assumption:
Tasks have group structures



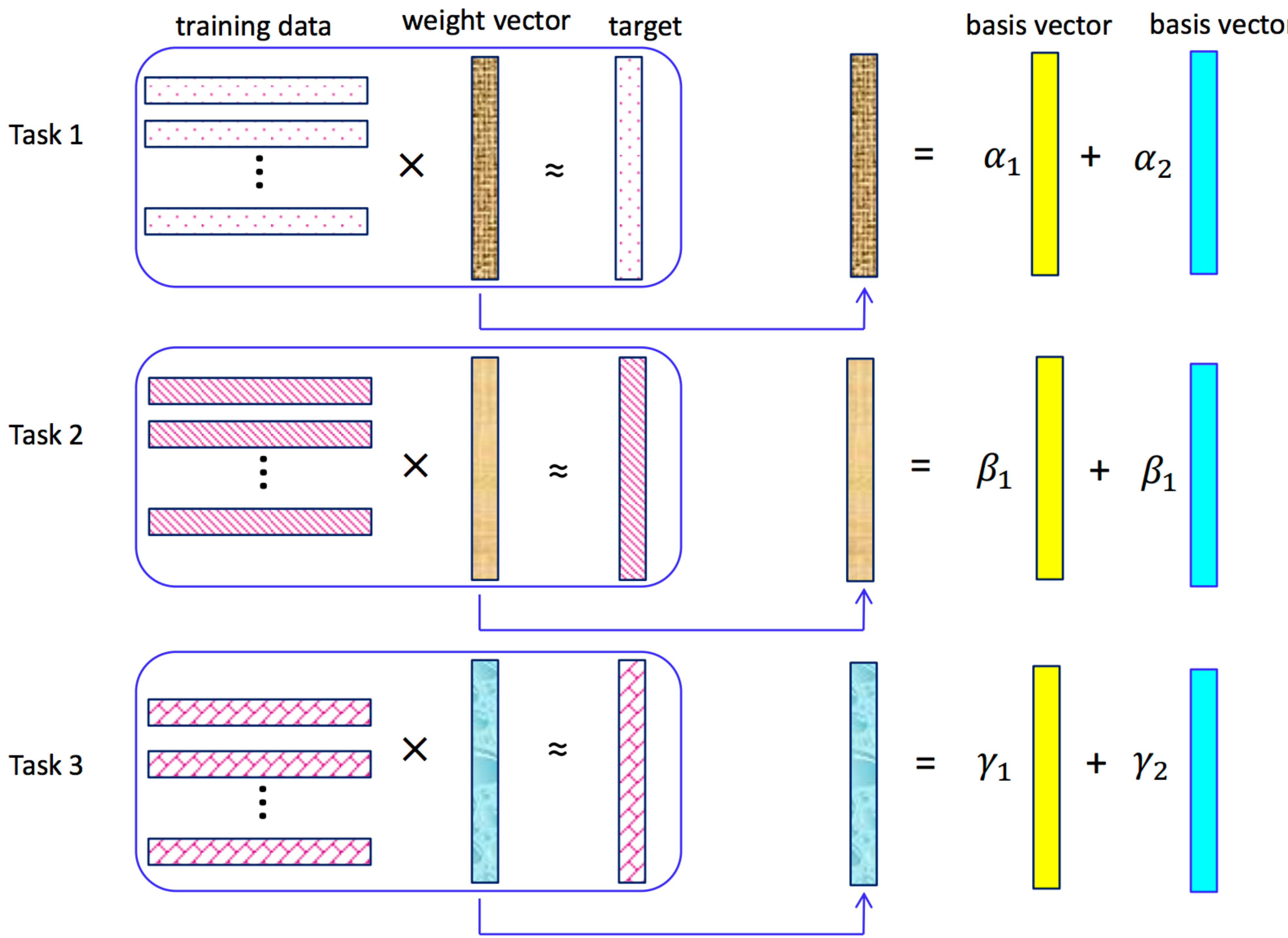
Assumption:
The relationship is not symmetric

Multi-Task Learning with Joint Feature Learning

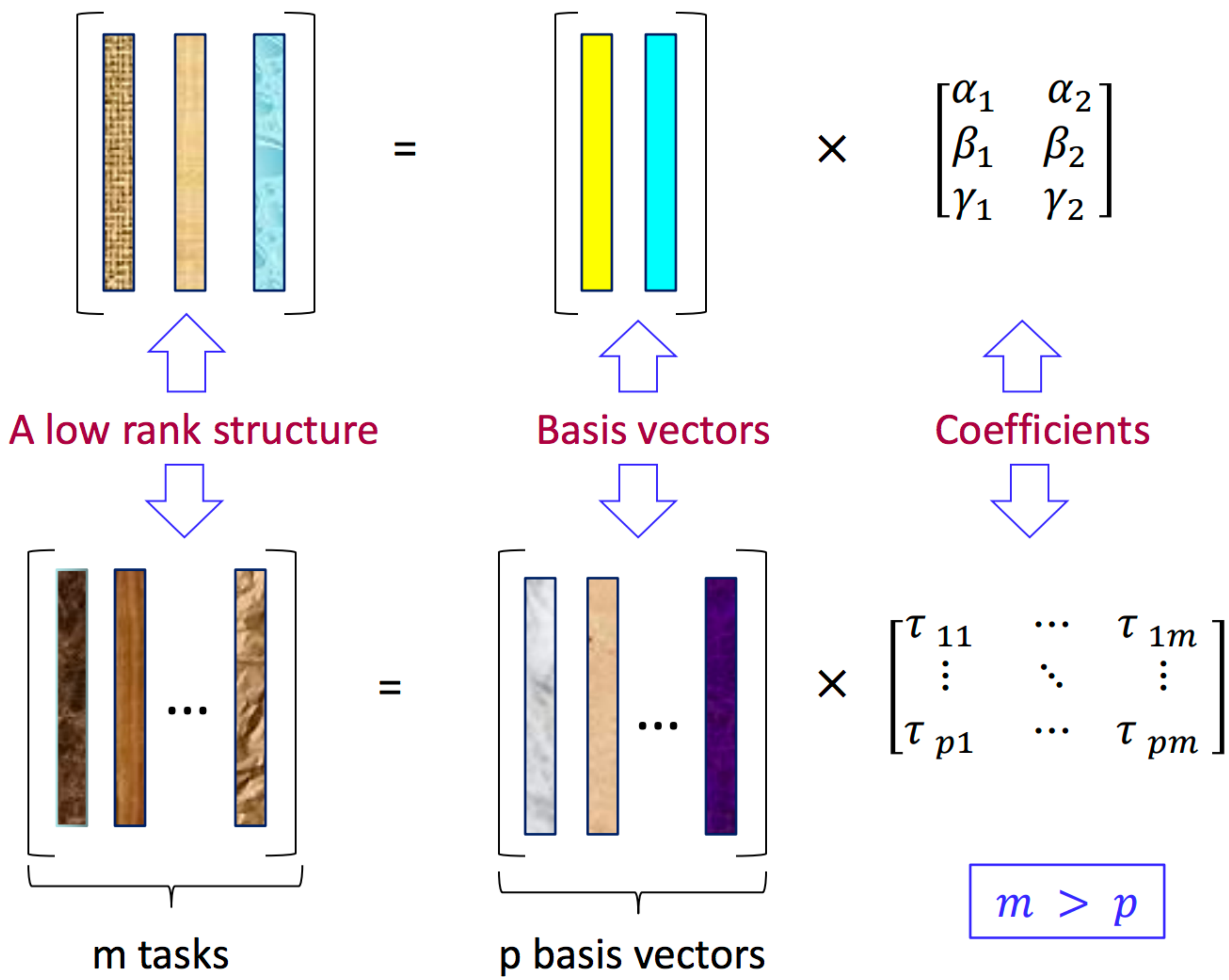
- Constrain all models to share a common set of features
- For example, scores from different schools may be determined by similar set of features
- Use regularization to constrain tasks to have a shared structure



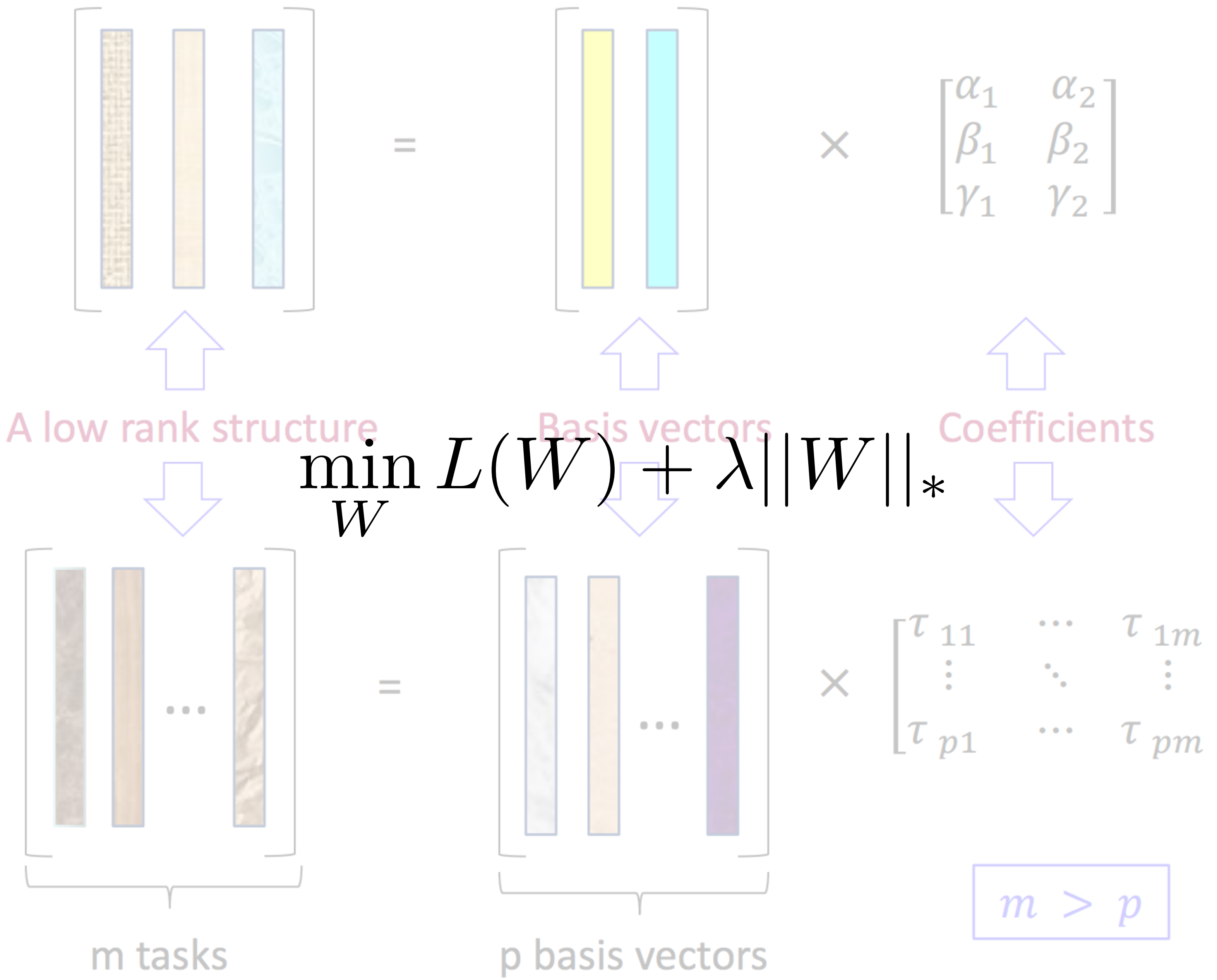
Trace-Norm Regularized MTL



Trace-Norm Regularized MTL (2)



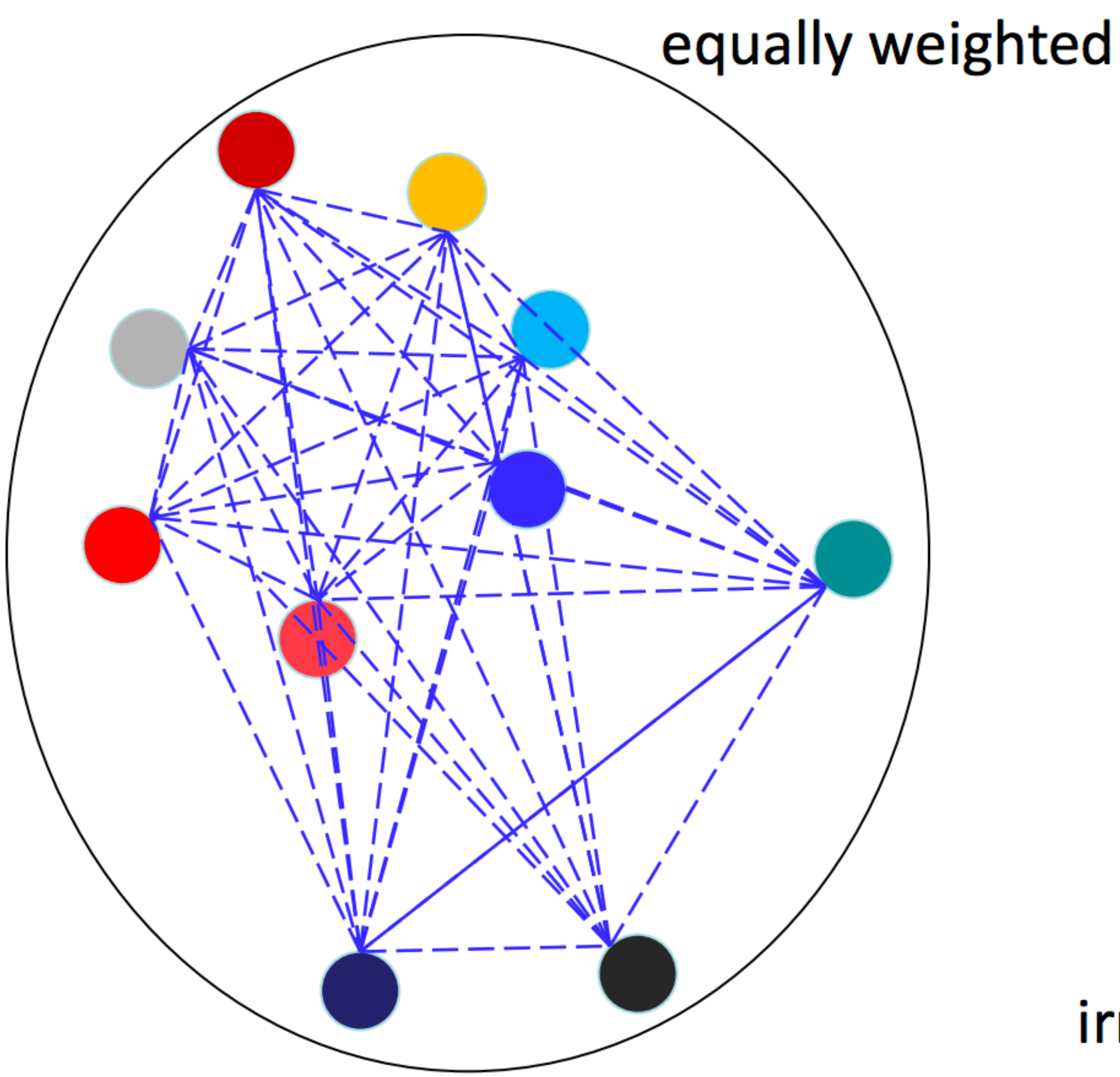
Trace-Norm Regularized MTL (3)



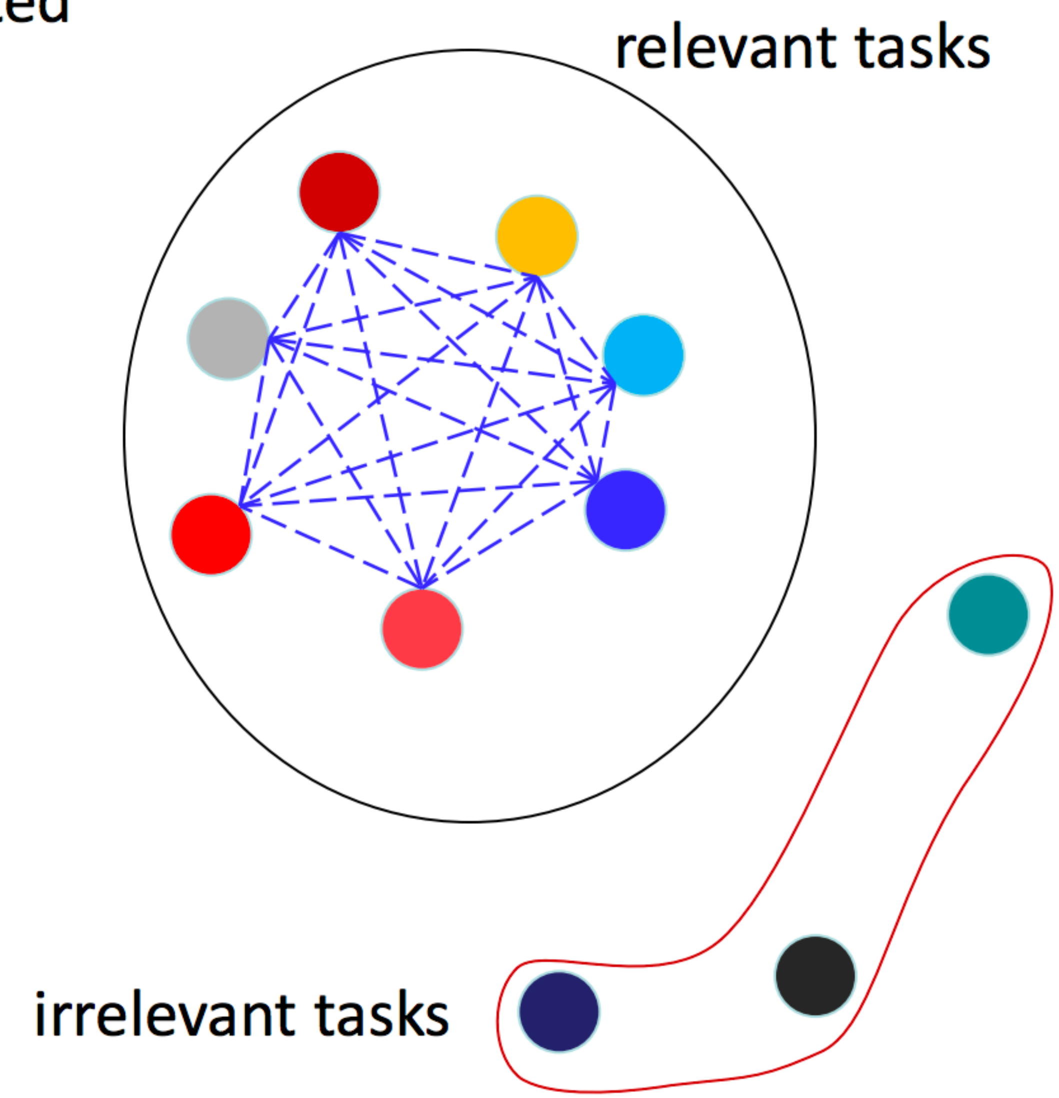
$$\min_{W} L(W) + \lambda ||W||_*$$

Robust Multi-Task Learning

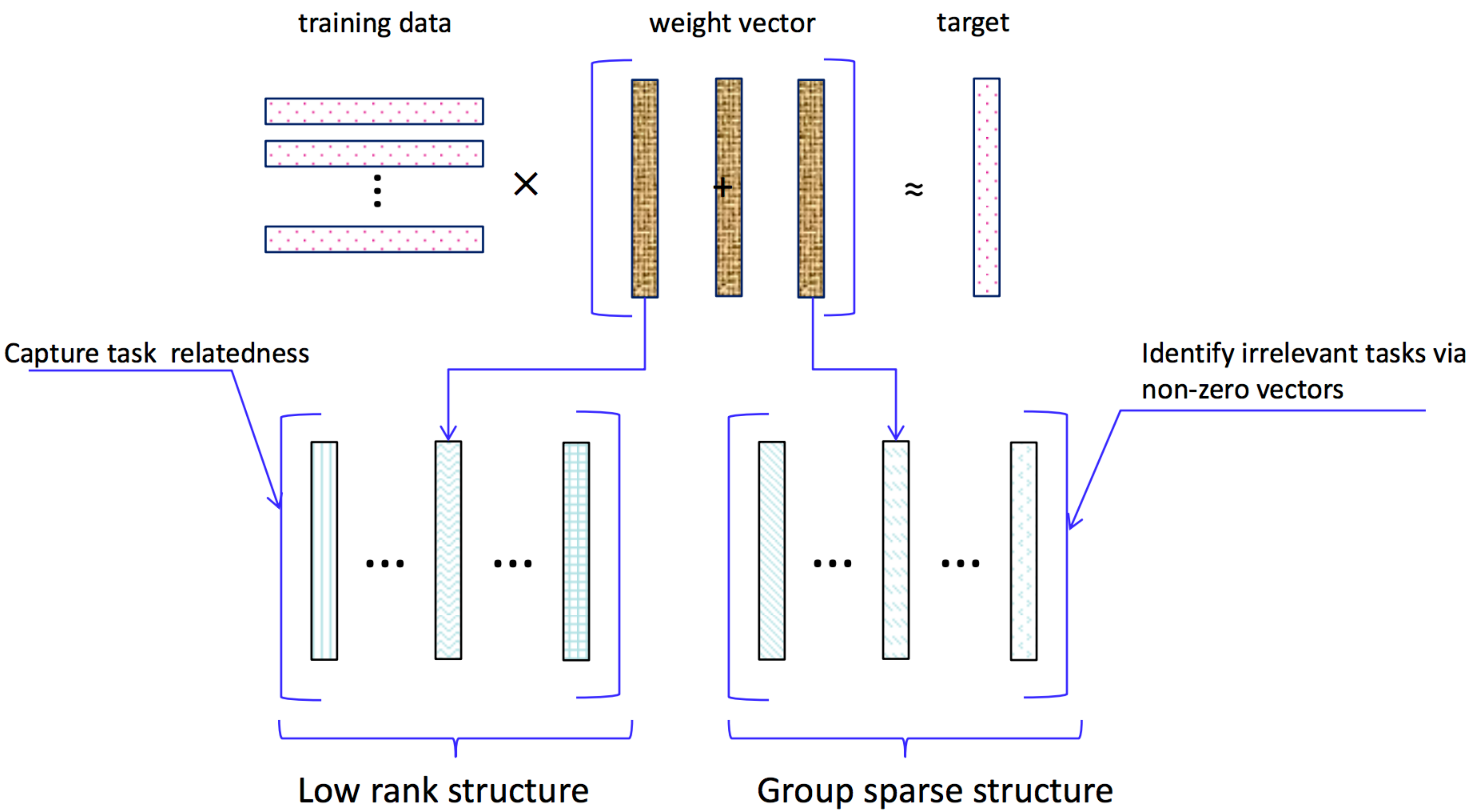
Existing MTL approaches



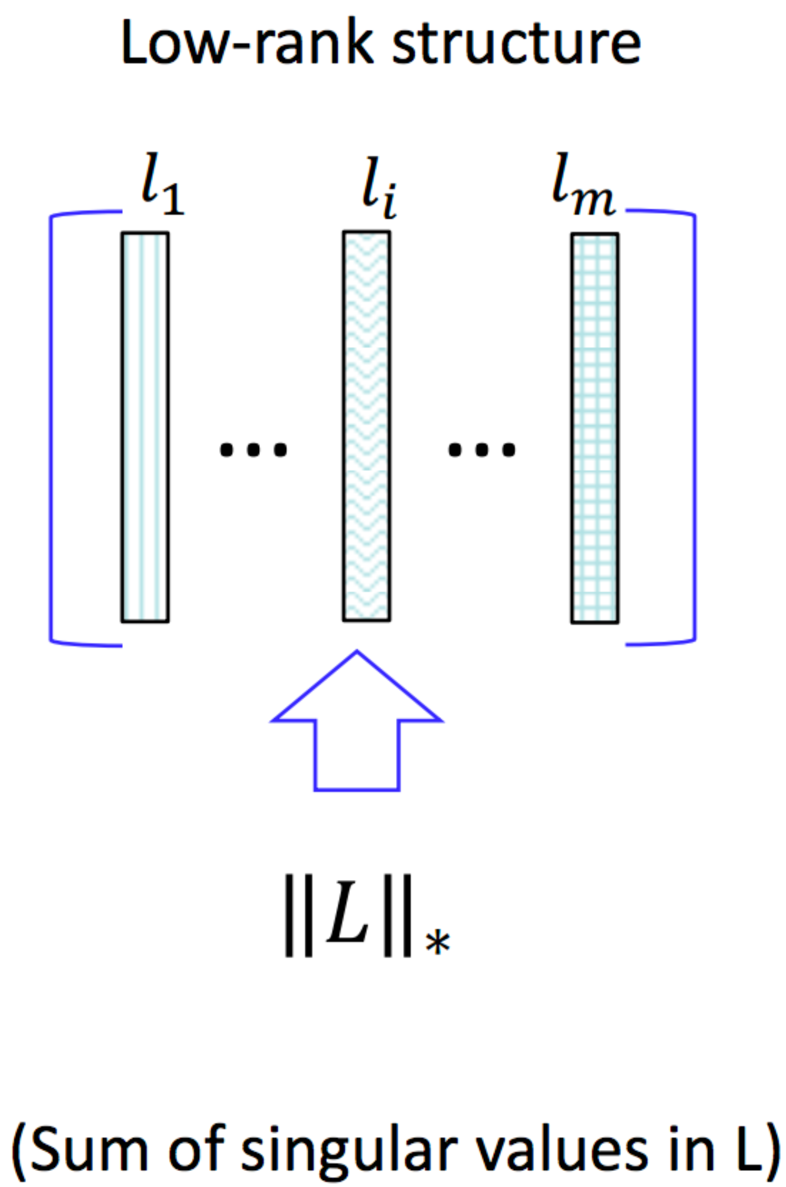
Robust MTL approaches

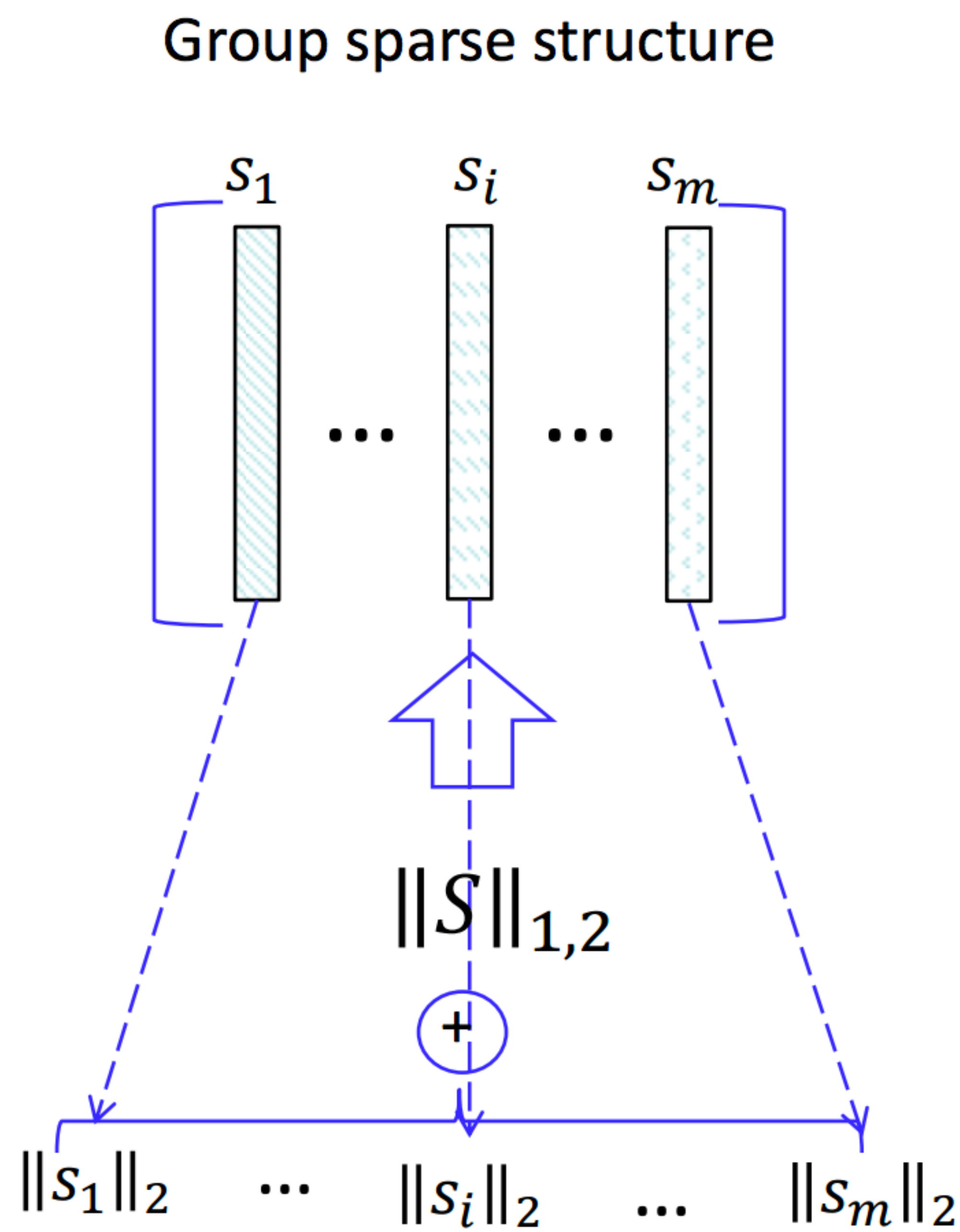


Low-Rank + Group Sparsity



Low-Rank + Group Sparsity (2)





Robust Multi-Task Learning Formulation

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

- Empirical loss on i^th task:

$$L_i(X_i(l_i + s), y_i) = ||X_i(l_i + s) - y_i)||_2^2$$

- Formulation:

$$\min \sum_i L_i(X_i(l_i + s), y_i) + \alpha||L||_* + \beta||S||_{1,2}$$

Robust Multi-Task Learning Formulation

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

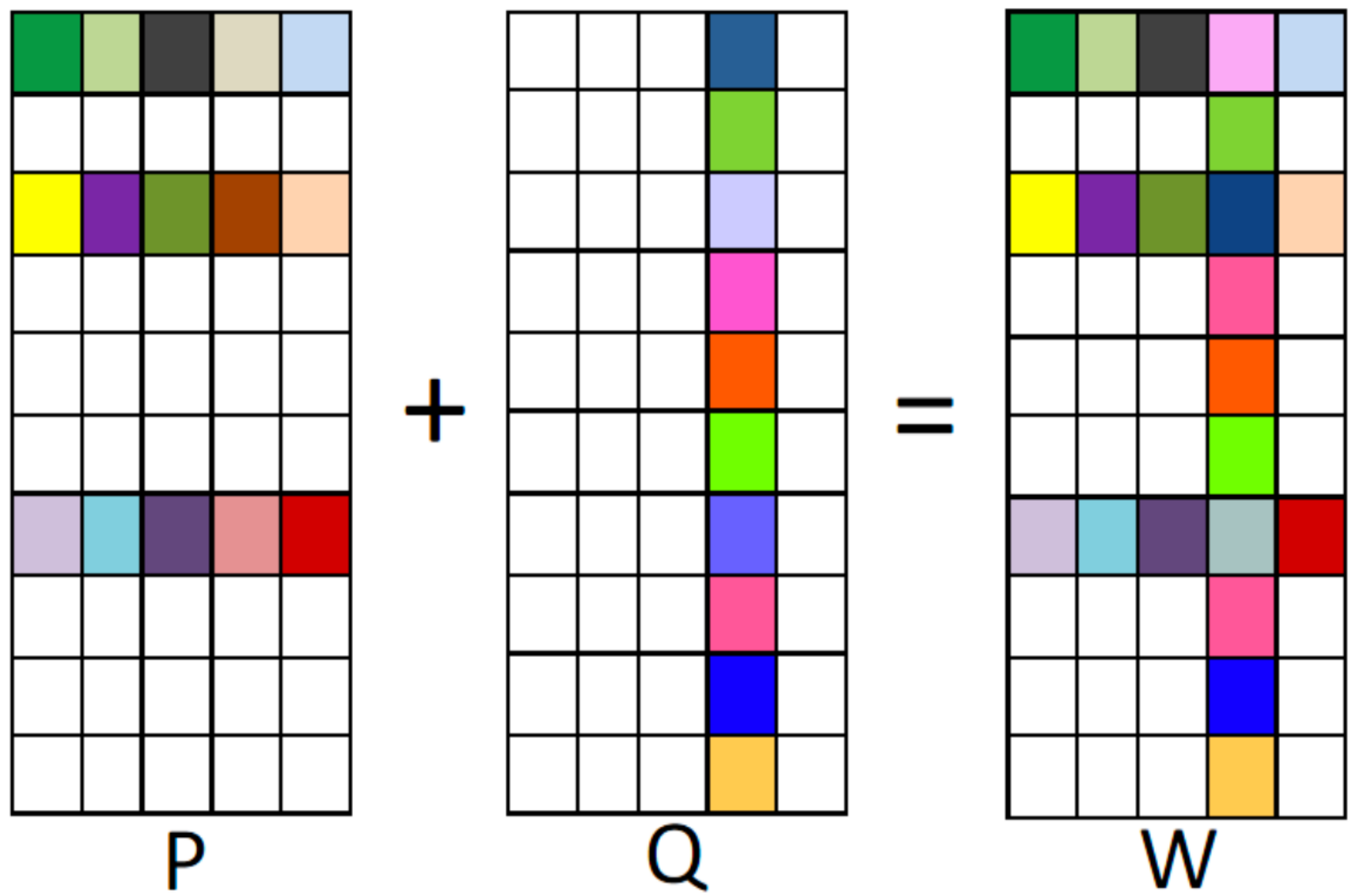
- Empirica

$||_2^2$

- Formulat

mir

$S||_{1,2}$



Algorithm

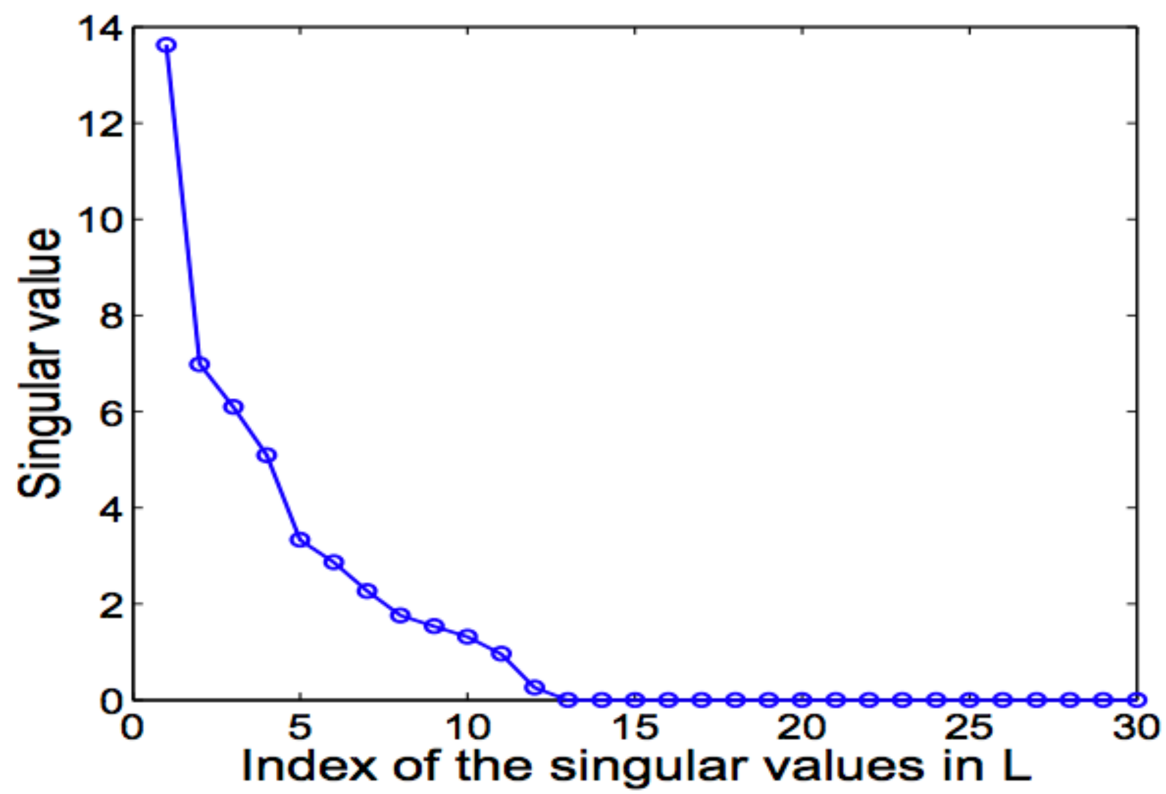
Use accelerated proximal method

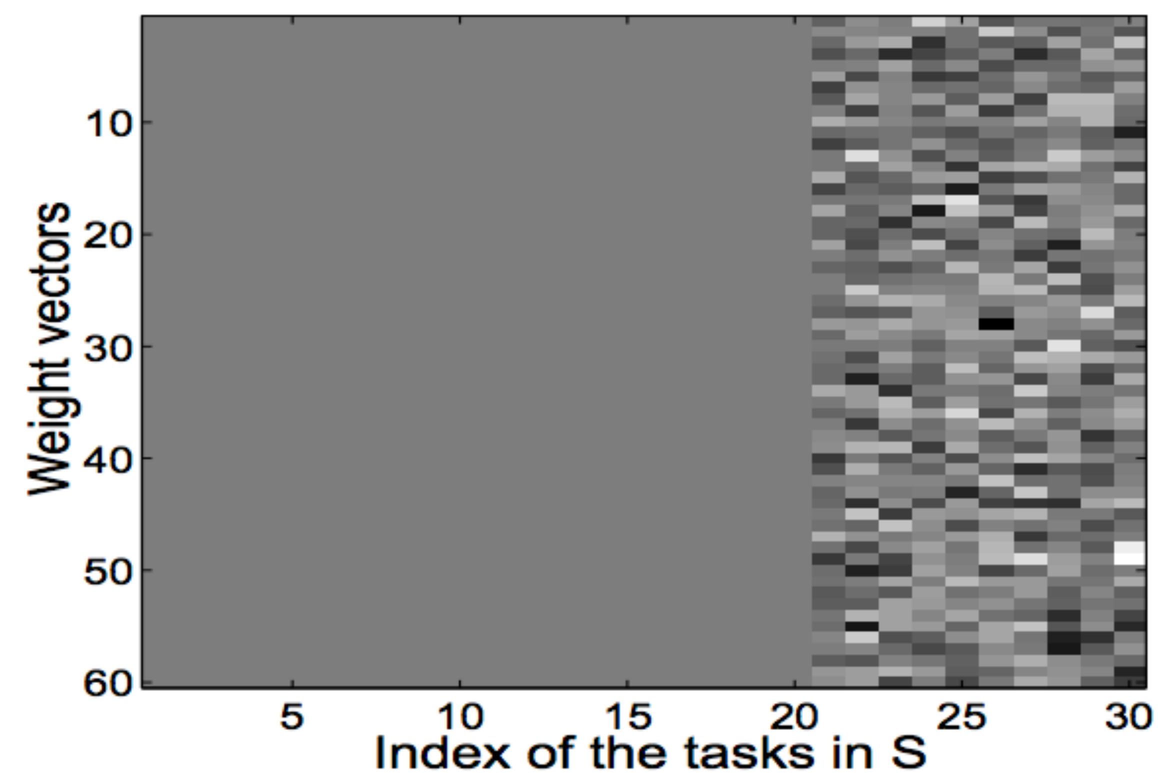
- Similar to proximal gradient descent
- Includes extrapolation step in the algorithm (sometimes referred to as momentum)
- Worst-case convergence is superior to standard method

Experiment: Synthetic Data

- 30 tasks — 20 related and 10 outlier
  - Each task has 50 samples with feature dimension of 60
  - Randomly generate low-rank component and set smallest 20 singular values at 0
  - Generate group-sparse component and set first 20 columns as zero vectors

Synthetic Data Results





Results are consistent with the setting of using the low-rank structure and the outlier-tasks

Experiments: Real Data

- School data — exam scores of 15,362 students from 139 secondary schools
  - Each student described by 27 attributes
  - Each school is a task
- SARCOS data — inverse dynamics prediction problem for 7 degrees-of-redox anthropomorphic robot arm with 48,933 observations to 7 joint torques
  - Each observation has 21 features
  - Each joint torque is a task

School Data Results

Measure	training ratio	Ridge	Lasso	TraceNorm	Sparse-LowRank	CMTL	Robust MTL
nMSE	10%	1.0398 ± 0.0038	1.0261 ± 0.0132	0.9359 ± 0.0370	0.9175 ± 0.0261	0.9413 ± 0.0021	$\mathbf{0.9130 \pm 0.0039}$
	20%	0.8773 ± 0.0043	0.8754 ± 0.0194	0.8211 ± 0.0032	0.8126 ± 0.0132	0.8327 ± 0.0039	$\mathbf{0.8055 \pm 0.0103}$
	30%	0.8171 ± 0.0090	0.8144 ± 0.0091	0.7870 ± 0.0012	0.7657 ± 0.0091	0.7922 ± 0.0052	$\mathbf{0.7600 \pm 0.0032}$
aMSE	10%	0.2713 ± 0.0023	0.2682 ± 0.0036	0.2504 ± 0.0102	0.2419 ± 0.0081	0.2552 ± 0.0032	$\mathbf{0.2330 \pm 0.0018}$
	20%	0.2303 ± 0.0003	0.2289 ± 0.0051	0.2156 ± 0.0015	0.2114 ± 0.0041	0.2131 ± 0.0071	$\mathbf{0.2018 \pm 0.0025}$
	30%	0.2165 ± 0.0021	0.2137 ± 0.0012	0.2089 ± 0.0012	0.2011 ± 0.0022	0.1922 ± 0.0102	$\mathbf{0.1822 \pm 0.0014}$

- Randomly selected 10%, 20%, 30% as training with rest as test with 15 random repetitions
- Multi-task algorithms (TraceNorm, Sparse-LowRank, CMTL, RMTL) outperform single-task learning
- RMTL outperforms all others

SARCOS Data Results

Measure	training size	Ridge	Lasso	TraceNorm	Sparse-LowRank	CMTL	Robust MTL
nMSE	50	0.2454 ± 0.0260	0.2337 ± 0.0180	0.2257 ± 0.0065	0.2127 ± 0.0033	0.2192 ± 0.0016	$\mathbf{0.2123 \pm 0.0038}$
	100	0.1821 ± 0.0142	0.1616 ± 0.0027	0.1531 ± 0.0017	0.1495 ± 0.0023	0.1568 ± 0.0037	$\mathbf{0.1456 \pm 0.0138}$
	150	0.1501 ± 0.0054	0.1469 ± 0.0028	0.1318 ± 0.0053	$\mathbf{0.1236 \pm 0.0004}$	0.1301 ± 0.0034	0.1245 ± 0.0015
aMSE	50	0.1330 ± 0.0143	0.1228 ± 0.0083	0.1122 ± 0.0064	0.1073 ± 0.0026	0.1156 ± 0.0011	$\mathbf{0.0982 \pm 0.0026}$
	100	0.1053 ± 0.0096	0.0907 ± 0.0023	0.0805 ± 0.0026	0.0793 ± 0.0047	0.0852 ± 0.0013	$\mathbf{0.0737 \pm 0.0083}$
	150	0.0846 ± 0.0045	0.0822 ± 0.0014	0.0772 ± 0.0023	$\mathbf{0.0661 \pm 0.0062}$	0.0755 ± 0.0025	0.0674 ± 0.0014

- Multi-task learning algorithms outperform single-task learning algorithms
- RTML performs comparably or better to all other competing algorithms
- Sparse-LowRank has similar performance to RTML => allowing each task to independently select discriminative features may improve robustness

Summary

- RTML algorithm captures task relationships using low-rank structure and identifies outlier tasks using group-sparse structure
- Adopt accelerated proximal method to solve optimization problem
- Theoretical analysis to obtain bound to characterize learning performance of RTML
- Experimental results on synthetic data and real-world data demonstrate effectiveness and efficiency of the algorithm