Recent Advances in Multi-Task Learning

Jiayu Zhou

Assistant Professor

Computer Science and Engineering

Michigan State University

Road Map

- **Part I**: Introduction to Multi-Task Learning (MTL)
- **Part II**: Distributed MTL and Privacy Protection
- **Part III**: Interactive Multi-Task Learning
- **Part IV**: Future Directions of MTL

Road Map

- **Part I**: Introduction to Multi-Task Learning (MTL)
- **Part II**: Distributed MTL and Privacy Protection
- **Part III**: Interactive Multi-Task Learning
- **Part IV**: Future Directions of MTL

Multiple Tasks

- Examination Scores Prediction[1]

School 1 - Alverno High School

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

student-dependent school-dependent

School 138 - Jefferson Intermediate School

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

student-dependent school-dependent

School 139 - Rosemead High School

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

student-dependent school-dependent

[1]The Inner London Education Authority (ILEA)

Learning Multiple Tasks

- Learning each task independently

School 1 - Alverno High School

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

task 1st

Excellent

Excellent

task 138th

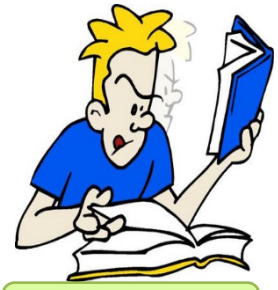
School 138 - Jefferson Intermediate School

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

School 139 - Rosemead High School

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

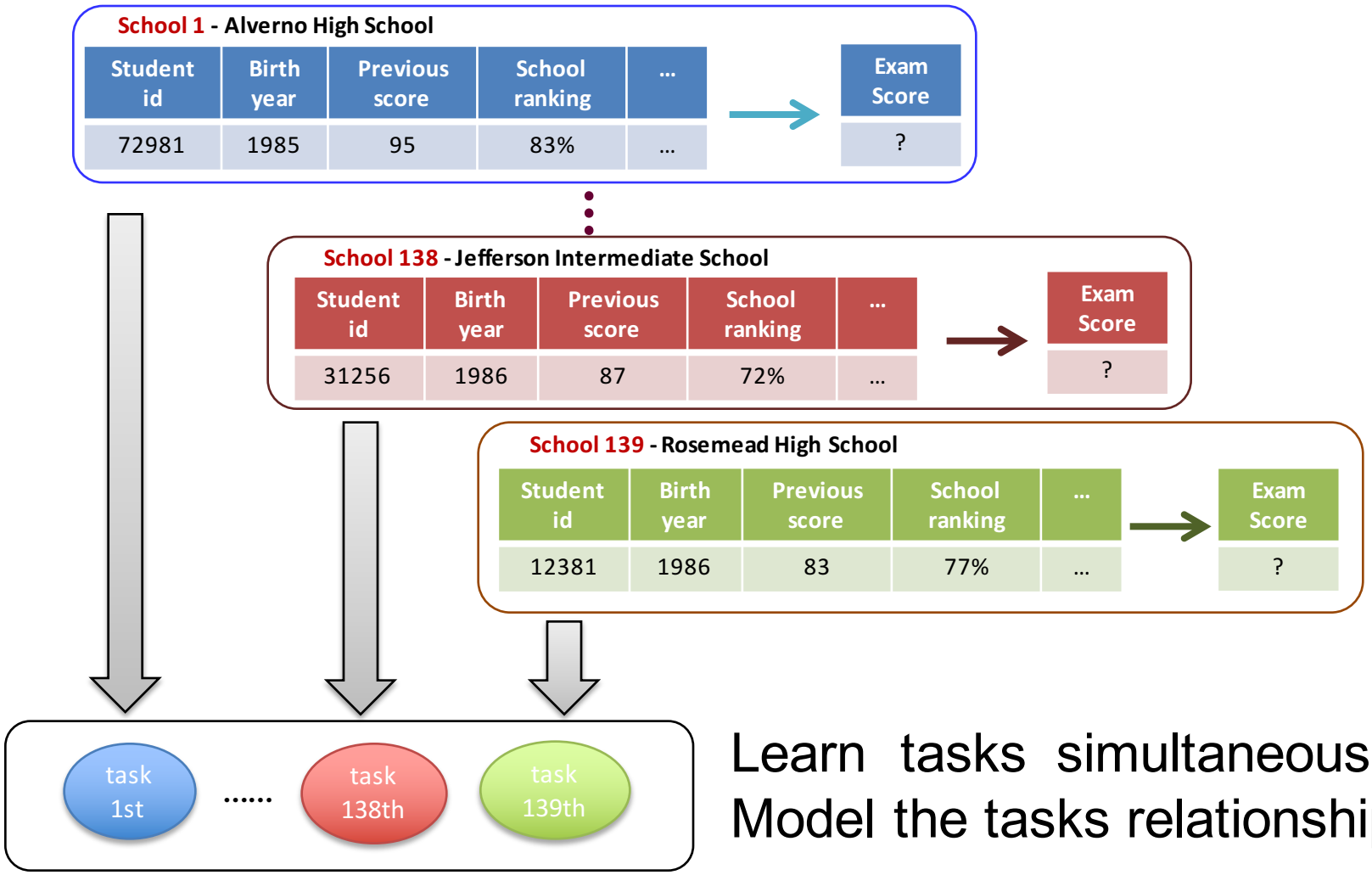
task 139th



Excellent

Learning Multiple Tasks

- Leaning multiple tasks simultaneously



Learn tasks simultaneously
Model the tasks relationship

More Applications of Multi-Task Learning

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

Collaborative ordinal regression

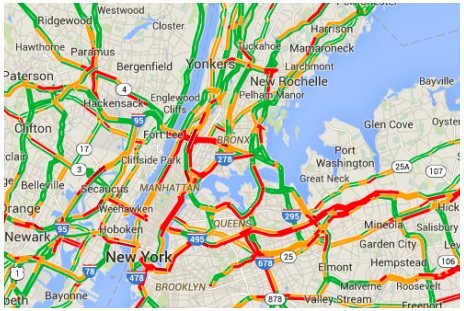
[Yu et. al. NIPS'06]

Disease progression modeling

[Zhou et. al. KDD'11, 12]

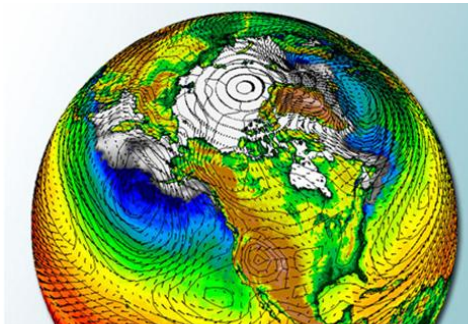
Web image and video search

[Wang et. al. CVPR'09]



Traffic prediction

[Heinze et. al. ETRR' 16]

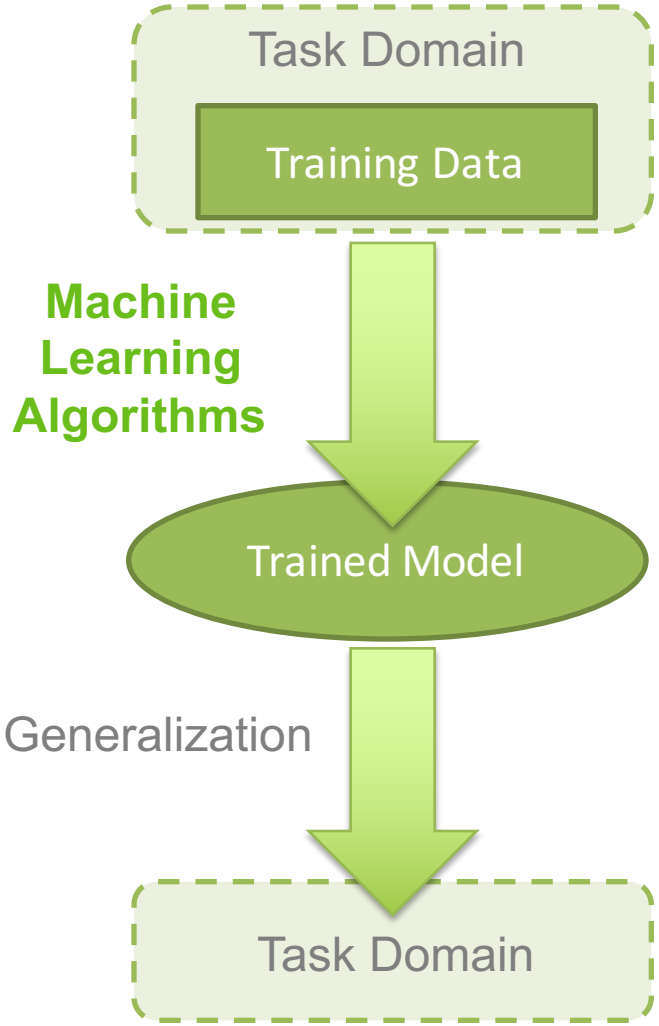


Climate Modeling

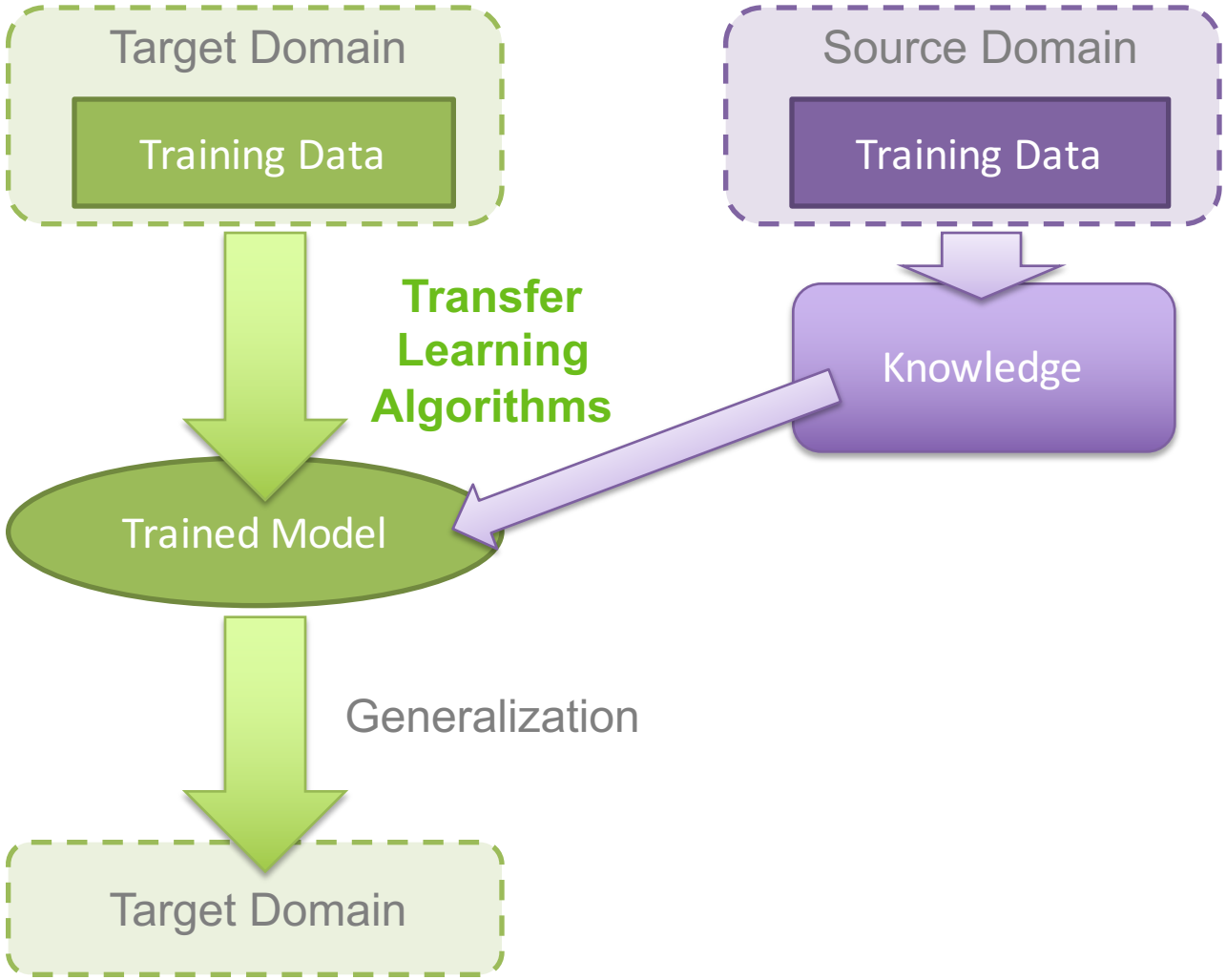
[Xu et. al. BigData'16]

Traditional Machine Learning

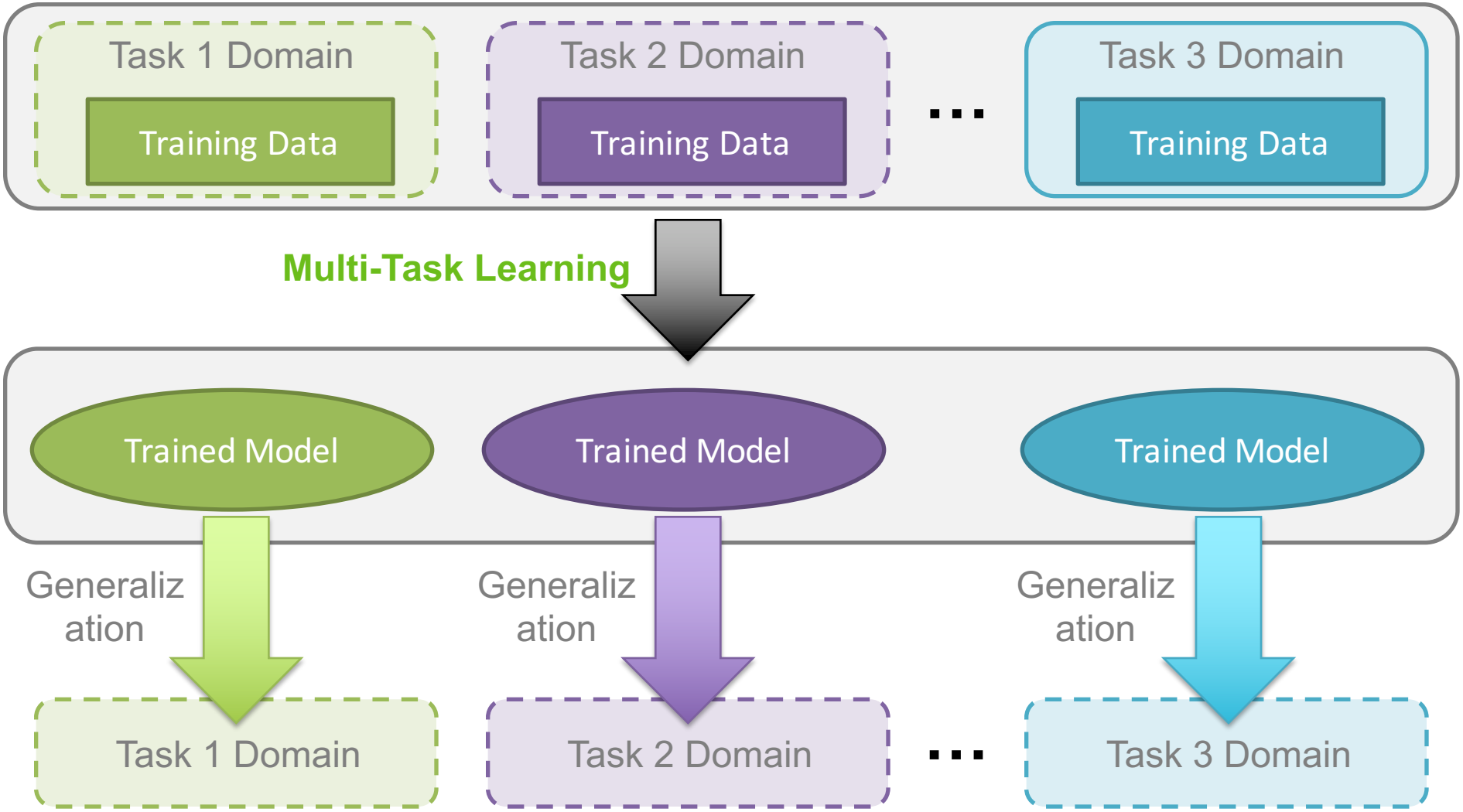
- Elements of machine learning on single task
  - The problem (task/domain)
  - Training data
  - Learning algorithms
  - Trained model
  - Applying model on unseen data (generalization)



Transfer Learning



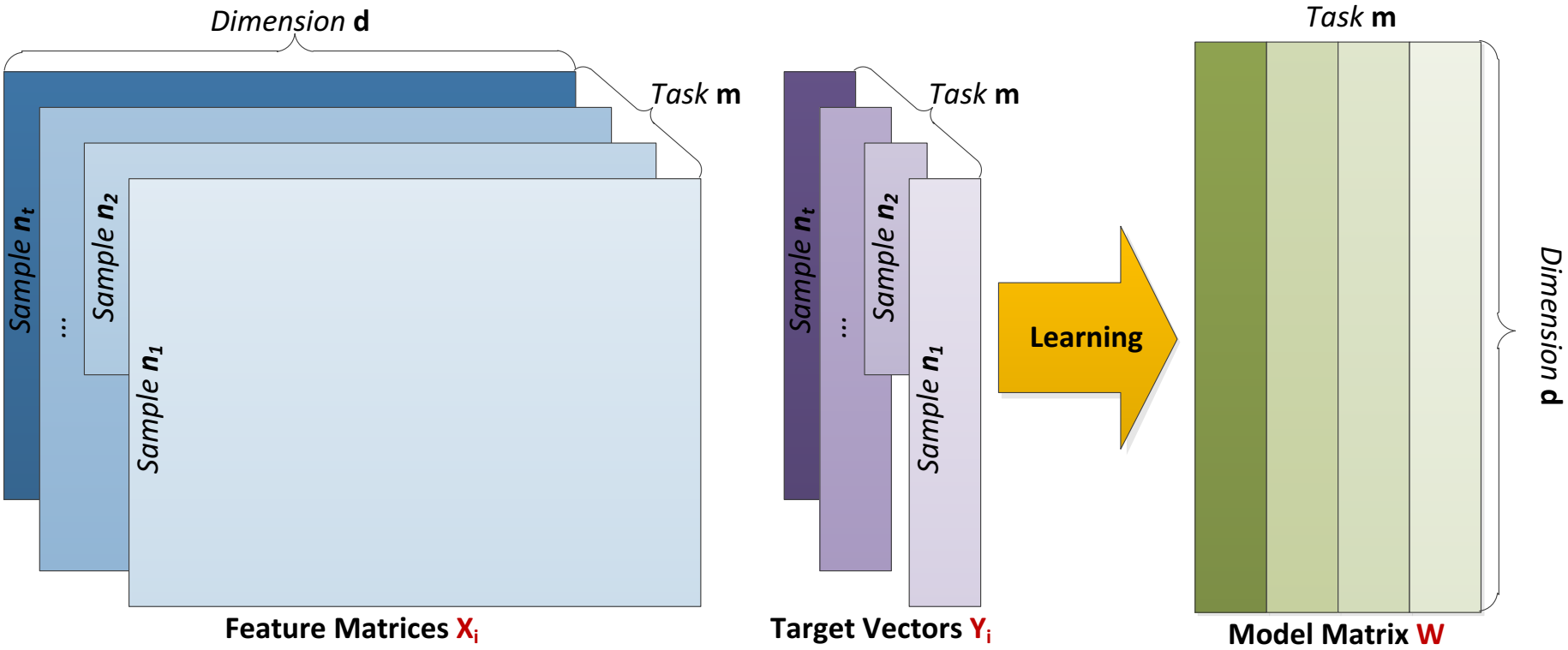
Multi-Task Learning



Multi-Task Learning in Big Data

- When does multi-task learning/transfer learning work?
  - When we don't have **enough data** to fit good models.
  - Do we still need it in the Big Data era?
- We may *still* not have enough data.
  - Incorporate high-dimensional features
  - Small sample size when it comes to a specific problem of interest
  - Systematic data bias

Linear Multi-Task Learning



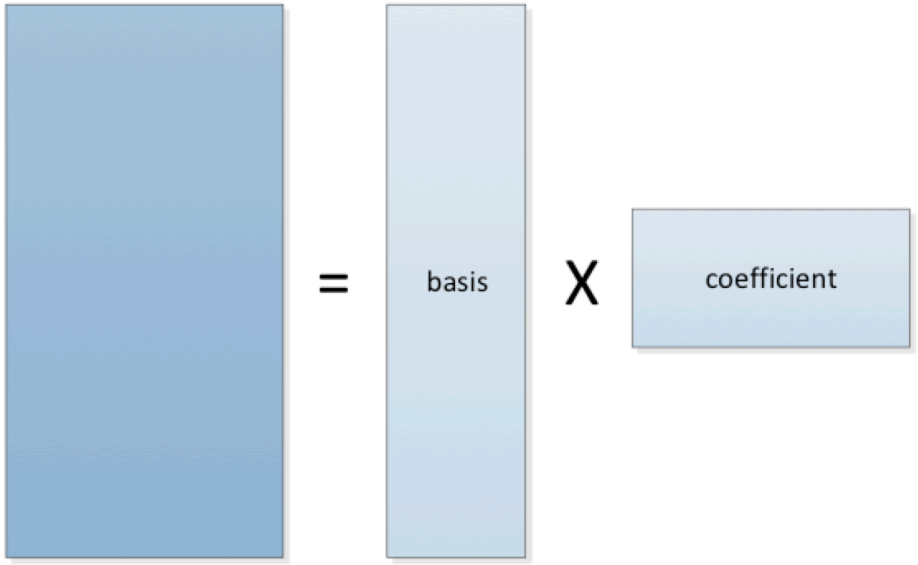
Regularized Multi-Task Learning

- The optimization problem
  - A *smooth* multi-task loss function and a *non-smooth* task relationship regularization.

$$\min_W \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(w_i) + \lambda R(W)$$

  - The loss function captures the original learning problems
    - Logistic regression for classification
    - Least squares for regression
  - The regularization enforcing task relationship

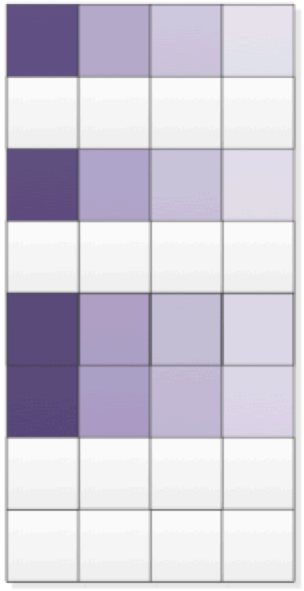
Example: Shared Subspace



$$\min_{W} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(w_i) + \lambda \|W\|_*$$

- The model vectors share a low-dimensional linear subspace.
- Knowledge transfer is done via learning the subspace and projection task into it.

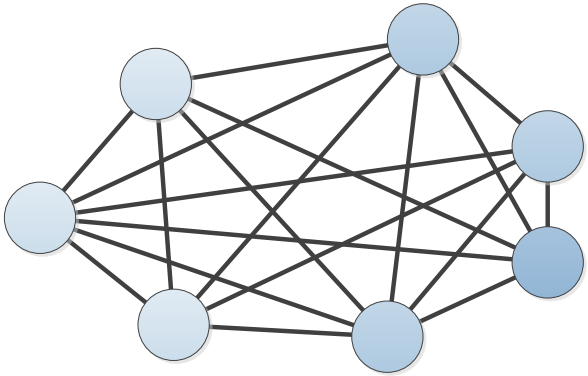
Example: Shared Feature Subset



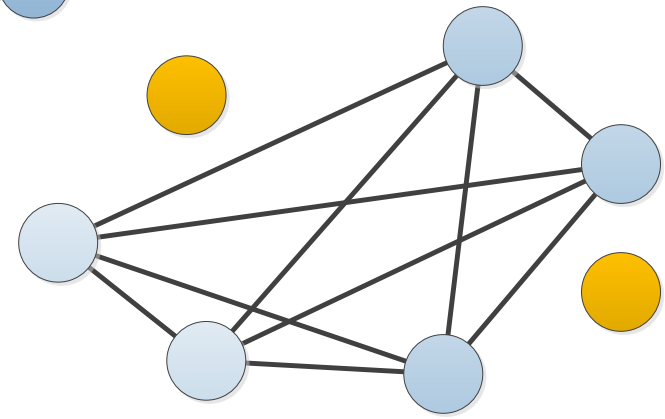
$$\min_W \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(w_i) + \lambda \|W\|_{2,1}$$

- Performs joint feature selection
- All the tasks share the same set of features.

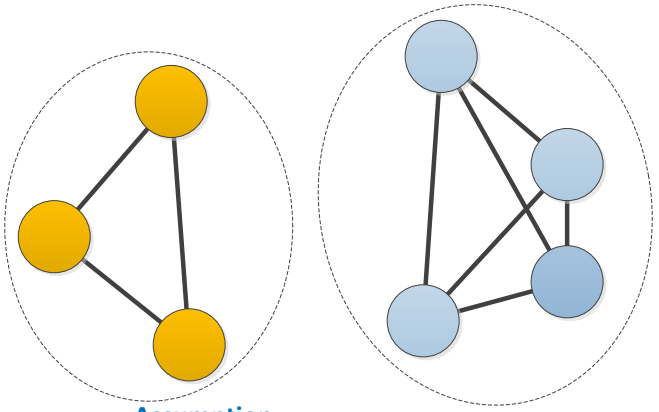
Task Relationship and Assumptions



Assumption:
All tasks are related

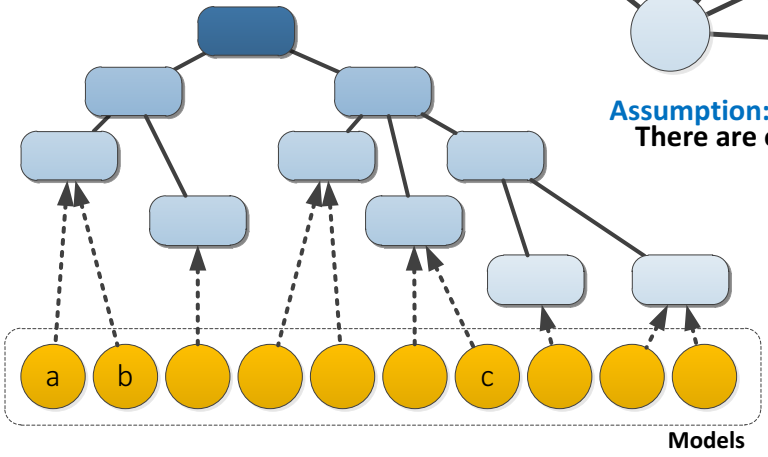


Assumption:
There are outlier tasks

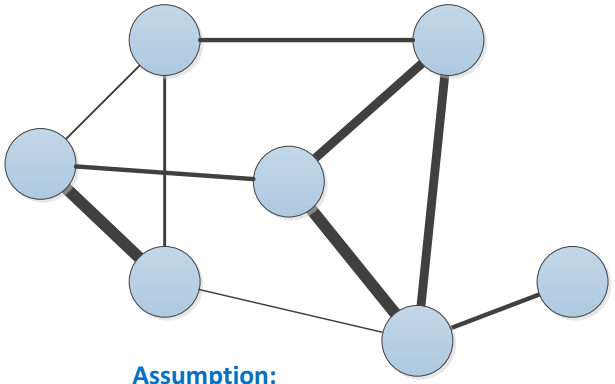


Assumption:
Tasks have group structures

Assumption:
Tasks have tree structures



Models



Assumption:
Tasks have graph/network structures

Solving Regularized MTL

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

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

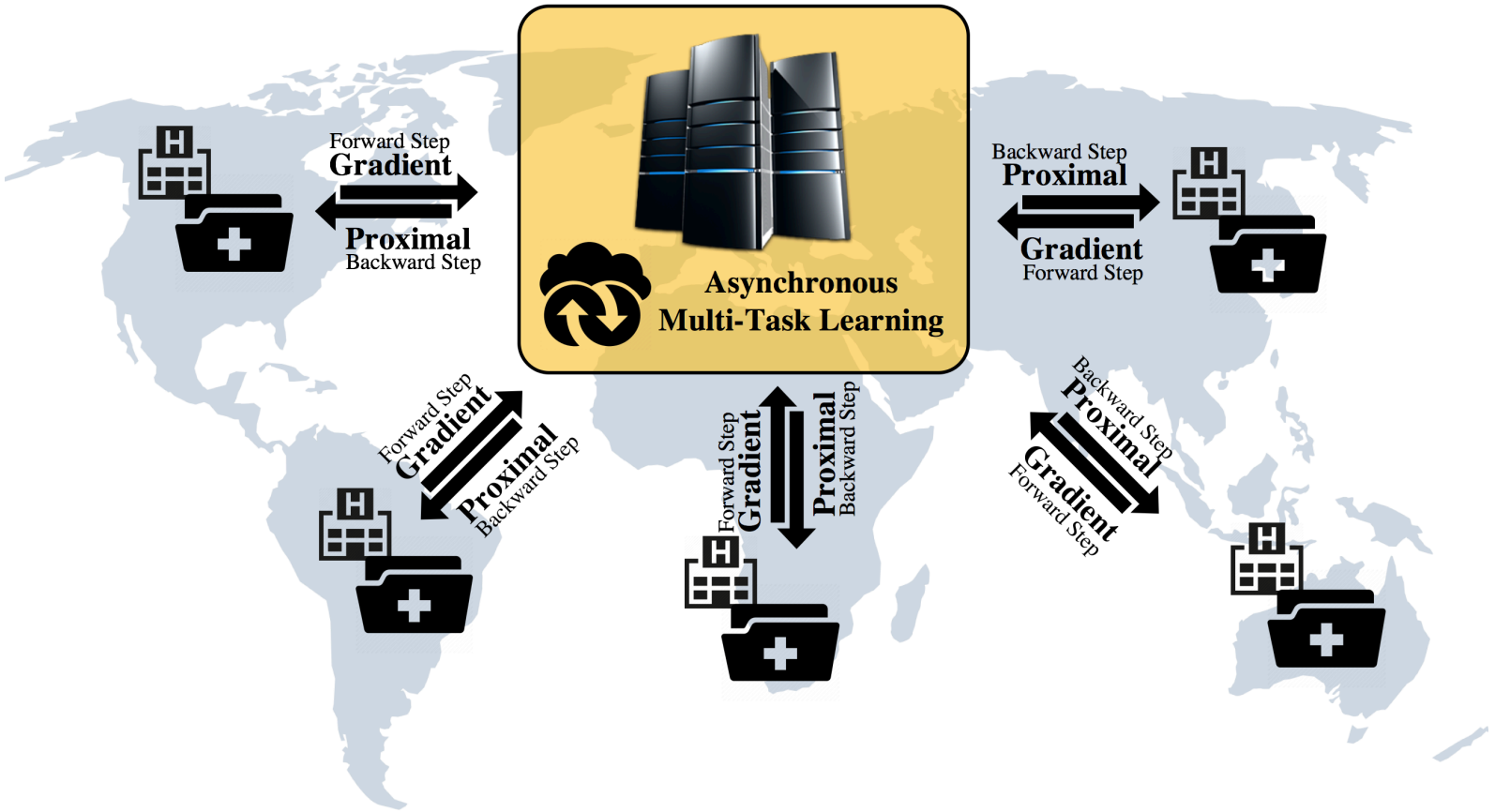
Summary Regularized Multi-Task Learning

- Designed to incorporate various assumptions from domain knowledge
- Trained using large-scale optimization algorithms on big data
- The key is to design the regularization term that couples the tasks.

Road Map

- **Part I**: Introduction to Multi-Task Learning (MTL)
- **Part II**: Distributed MTL and Privacy Protection
- **Part III**: Interactive Multi-Task Learning
- **Part IV**: Future Directions of MTL

Distributed multi-task learning (DMTL)



- Baytas et al. "Asynchronous multi-task learning" **ICDM 2016**
- Xie et al. "Privacy-Preserving Distributed Multi-Task Learning with Asynchronous Updates." **KDD 2017**

Distributed gradient of MTL

- Given a multi-task formulation

$$\min_W \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(w_i) + \lambda R(W)$$

- Proximal gradient descent solver

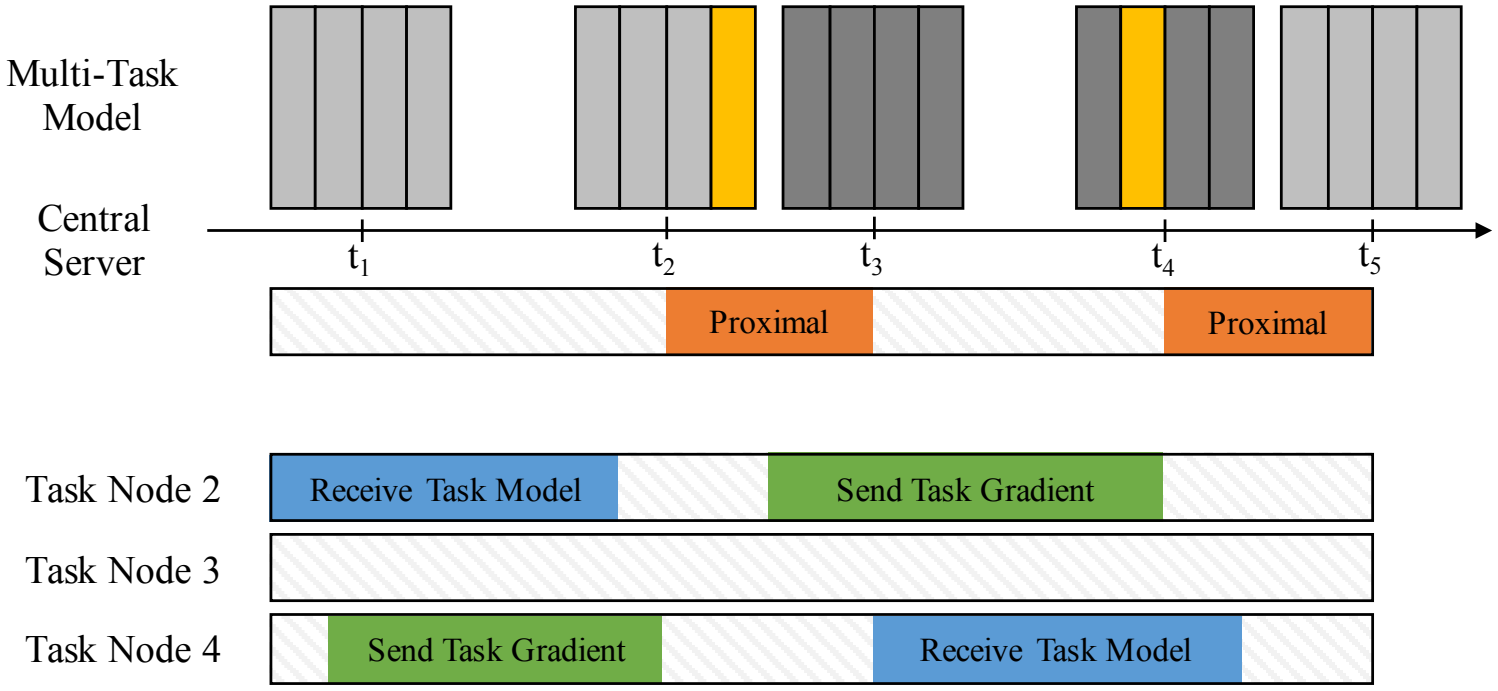
$$W^+ = \text{Prox}\left(W - \eta \nabla_W \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(w_i)\right) = \text{Prox}\left(W - \eta \left[\frac{\nabla_{w_1} \ell_1(w_1)}{n_1} \dots \frac{\nabla_{w_t} \ell_t(w_t)}{n_t}\right]\right)$$

- Seems to be perfect with distributed data. Any issues?

Asynchronous Gradient [Baytas ICDM16]

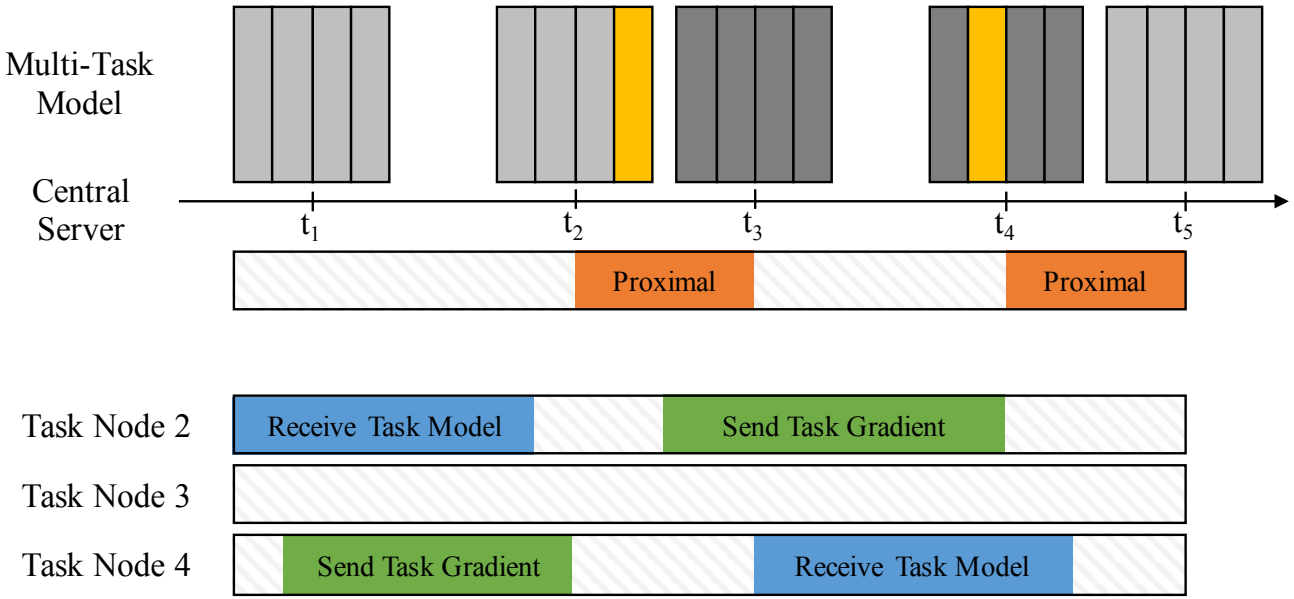
$$w_t^{k+1} = w_t^k + \eta_k \left(F_{BFS}\left(\hat{w}^k\right) - w_t^k \right)$$

$$F_{BFS}\left(\hat{w}^k\right) = \left(\text{Prox}_{\eta\lambda}\left(\hat{w}^k\right)\right)_t - \eta \nabla \ell_t \left(\left(\text{Prox}_{\eta\lambda}(\hat{w}^k)\right)_t \right)$$



Baytas et al. "Asynchronous multi-task learning" **ICDM 2016**

Asynchronous Gradient [Baytas ICDM16]



Theorem 1. *Let* $(V^k)_{k\geq 0}$ *be the sequence generated by the proposed AMTL with* $\eta_k \in [\eta_{\min}, \frac{c}{2\tau/\sqrt{T}+1}]$ *for any* $\eta_{\min} > 0$ *and* $0 < c < 1$, *where* τ *is the maximum delay. Then* $(V^k)_{k\geq 0}$ *converges to an* V^**-valued random variable almost surely. If the MTL problem in Eq. III.1 has a unique solution, then the sequence converges to the unique solution.*

Baytas et al. "Asynchronous multi-task learning" **ICDM 2016**

Two issues of AMTL

- Cannot directly be applied to model-decompose MTL, a class of general MTL with great flexibility.

$$\min_{W=P+Q} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(p_i + q_i) + \lambda R(P) + \gamma S(Q)$$

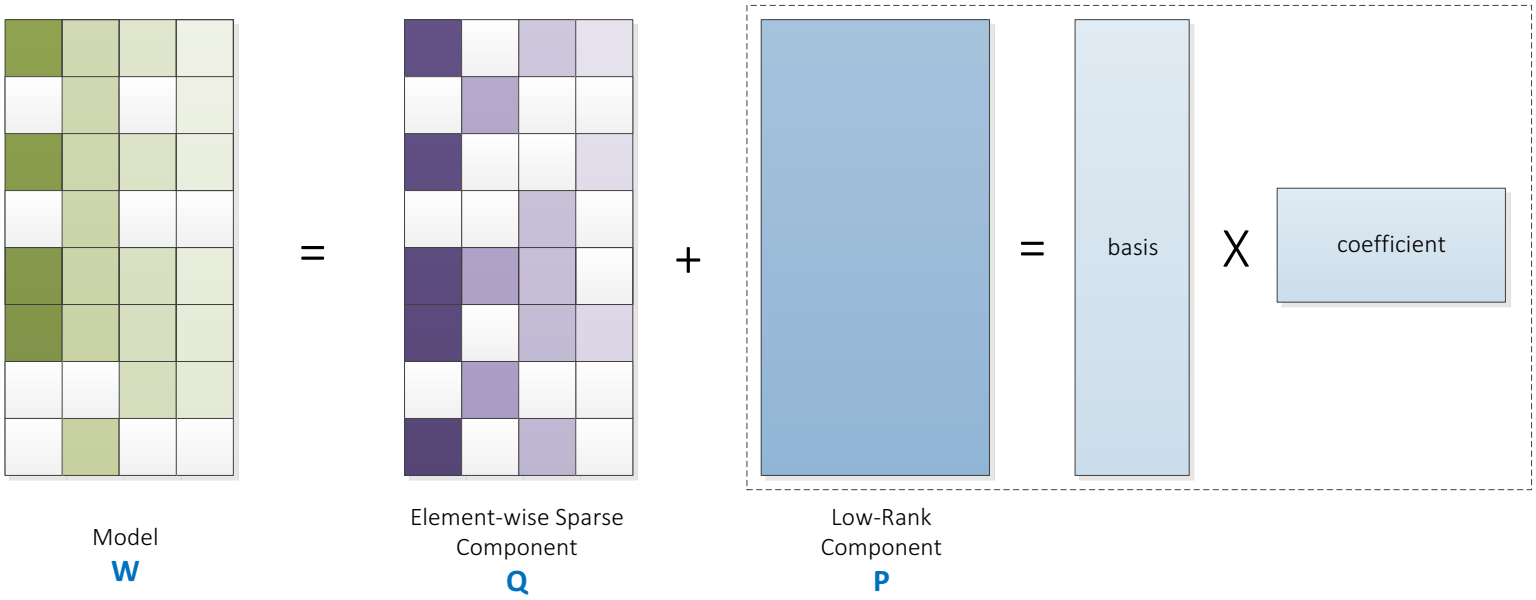
- Cannot ensure the privacy of the data sources.
  - Is the gradient operator safe to protect the data?

Model-Decompose MTL

$$\min_{W=P+Q} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(p_i + q_i) + \lambda R(P) + \gamma S(Q)$$

- Dirty models: allow a shared component (P) and task specific component (Q).
- Robust models: can be used to detect outliers (Q)
- "Pure" multi-task learning is a special case (by setting gamma to infinity).

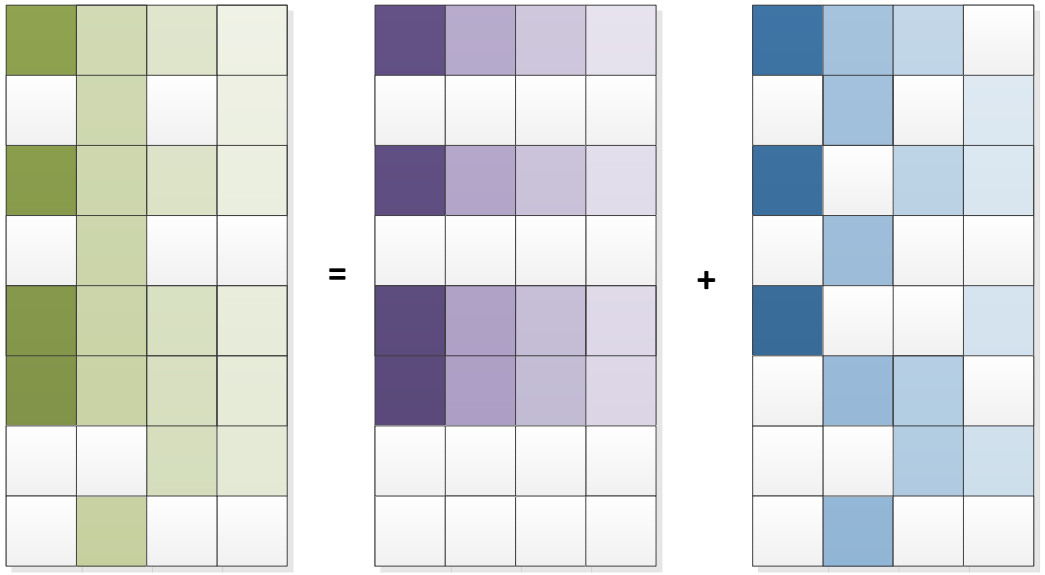
Dirty Models



$$\min_{P,Q} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(p_i + q_i) + \lambda \|P\|_* + \gamma \|Q\|_1$$

Chen, Jianhui, Ji Liu, and Jieping Ye. "Learning incoherent sparse and low-rank patterns from multiple tasks." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 5.4 (2012): 22.

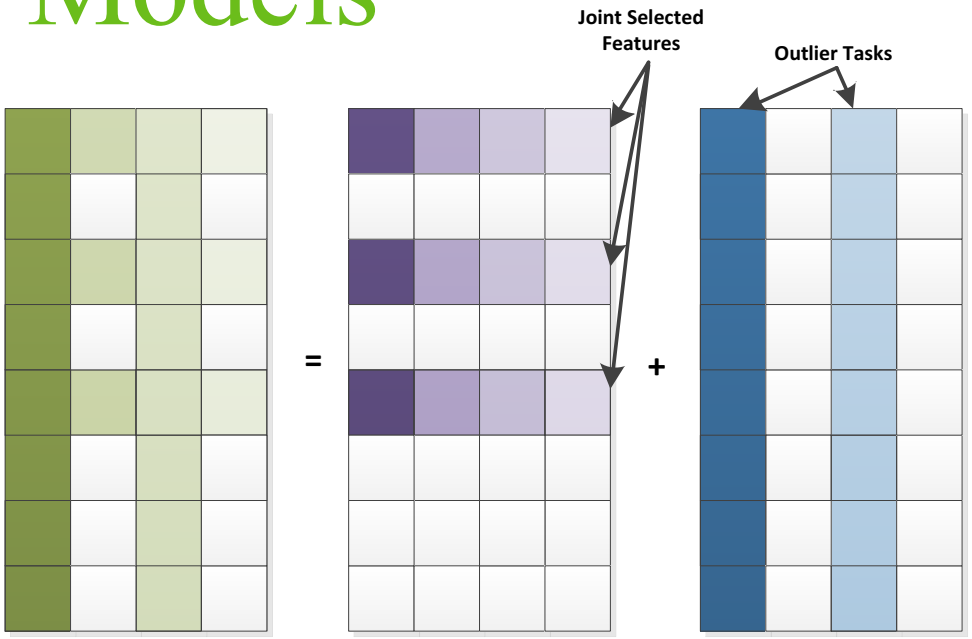
Dirty Models (cont.)



$$\min_{P,Q} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(p_i + q_i) + \lambda \|P\|_{1,q} + \gamma \|Q\|_1$$

Jalali, Ali, et al. "A dirty model for multi-task learning." *Advances in Neural Information Processing Systems*. 2010.

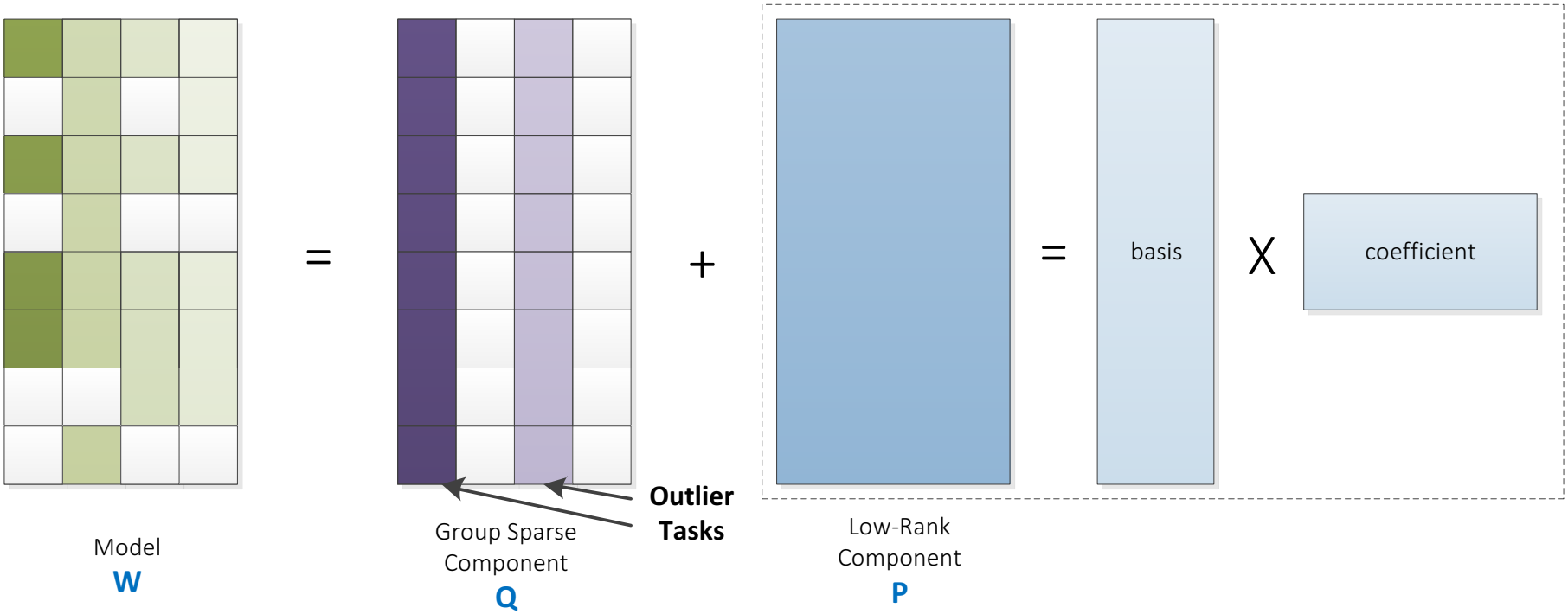
Robust Models



$$\min_{P,Q} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(p_i + q_i) + \lambda \|P\|_{1,q} + \gamma \|Q^T\|_{1,q}$$

Gong, Pinghua, Jieping Ye, and Changshui Zhang. "Robust multi-task feature learning." *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

Robust Models (cont.)



$$\min_{P,Q} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(p_i + q_i) + \lambda \|P\|_* + \gamma \|Q^T\|_{1,q}$$

Chen, Jianhui, Jiayu Zhou, and Jieping Ye. "Integrating low-rank and group-sparse structures for robust multi-task learning." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.

Distributed Learning of Model-Decompose MTL

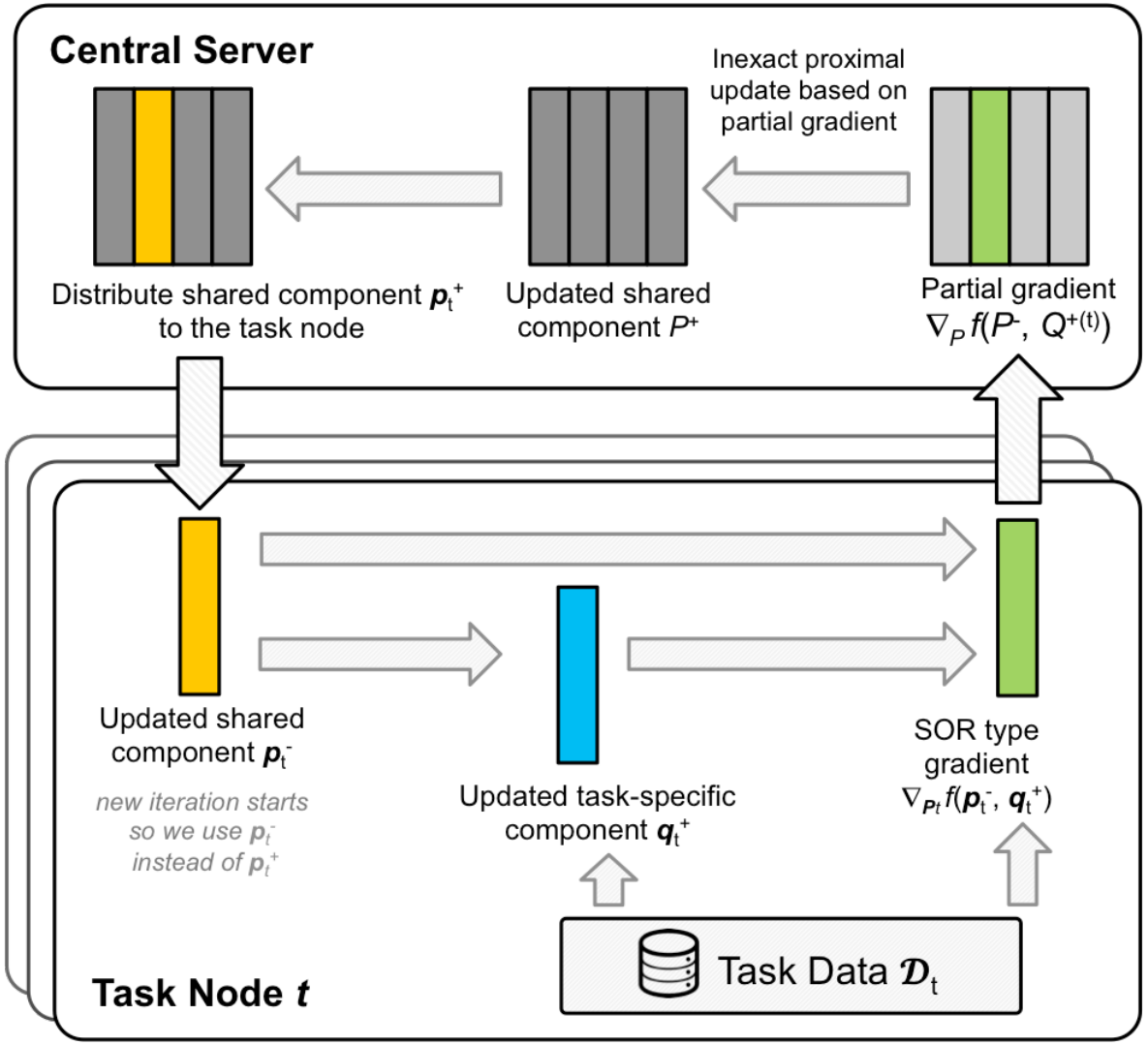
$$\min_{W=P+Q} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(p_i + q_i) + \lambda R(P) + \gamma S(Q)$$

- Optimization via block coordinate descent

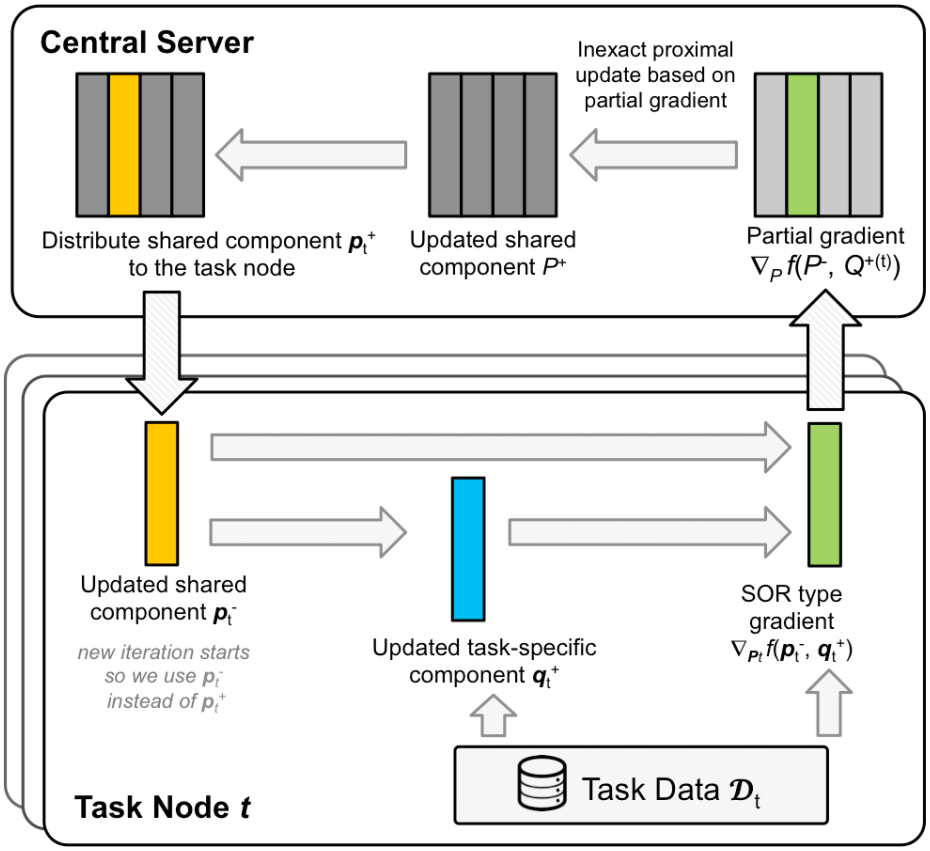
$$\begin{aligned} q_k^+ &= \text{argmin}_{q_k} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(p_i^- + q_i) + \gamma \sum_{i=1}^{t} s(q_k) \\ &= \text{argmin}_{q_k} \frac{1}{n_i} \ell_i(p_i^- + q_i) + \gamma s(q_k) \\ P^+ &= \text{prox}_R^{\alpha\lambda}(P^- - \alpha \nabla_P f(P^-, Q^+)) \end{aligned}$$

  - Given p, the update of q can be obtained at each local server.
  - Given the gradient from each local server, the update of P can be done in central server.

Distributed Learning of Model-Decompose MTL

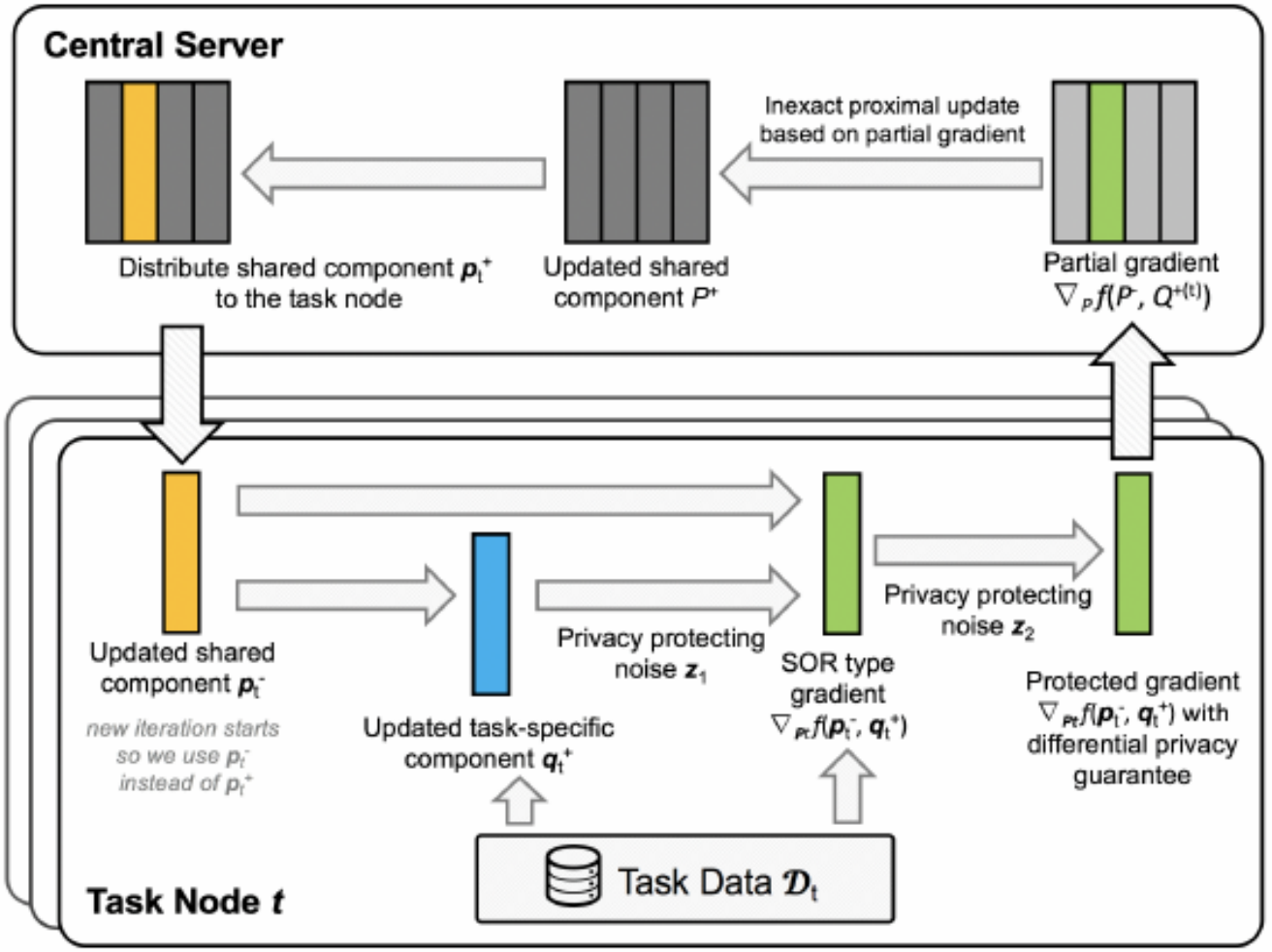


Distributed Learning of Model-Decompose MTL

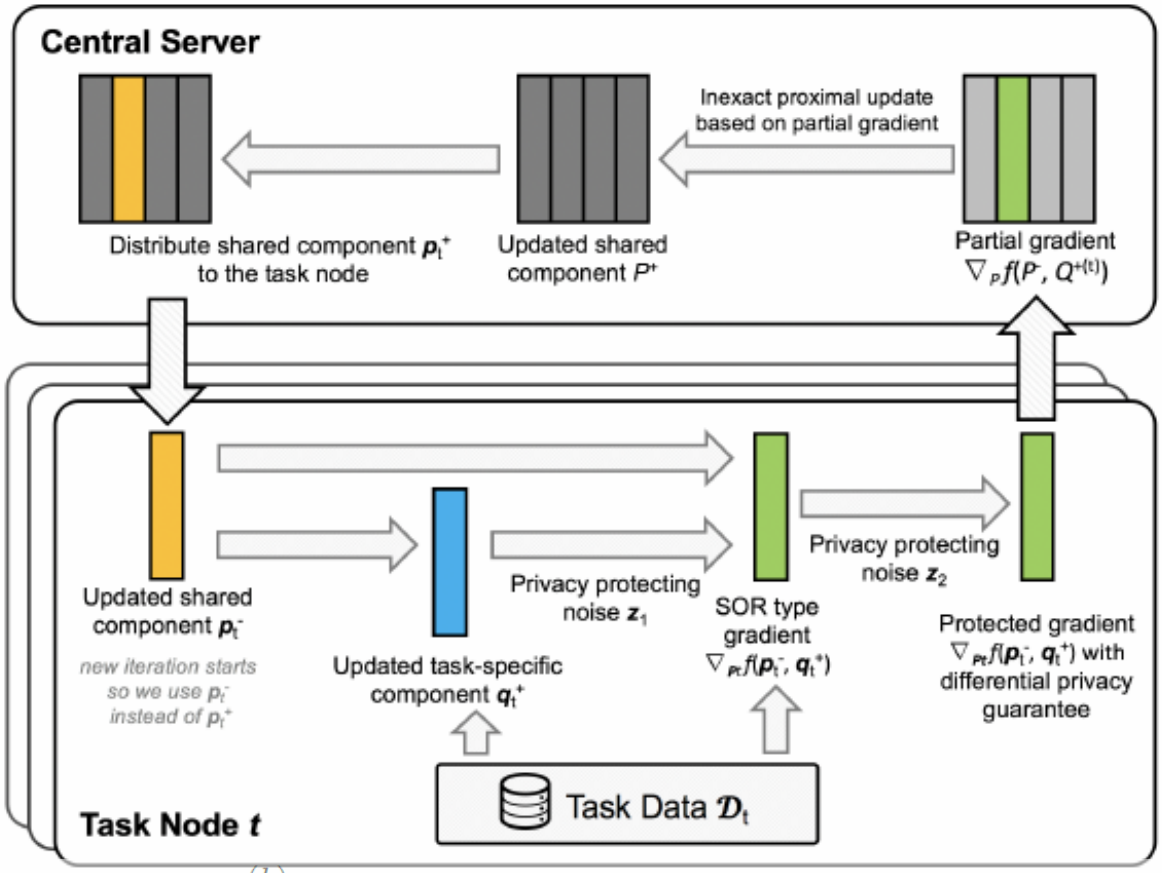


- During the learning an attacker at the server can collect gradient information
- Client data can be recovered by solving a sensing problem when enough gradients are available

DMTL with Differential Privacy



- Xie et al. "Privacy-Preserving Distributed Multi-Task Learning with Asynchronous Updates." **KDD 2017**

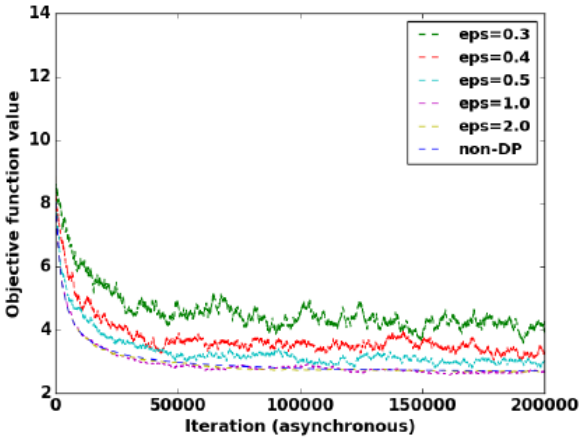


Theorem 1. *$\nabla_t^{(k)}$ guarantee 2ϵ-differential privacy with respect to the data set $\{\mathbf{X}_t, \mathbf{y}_t\}$ for classification problem with $\mathbf{y}_{t,n_t} \in \{+1, -1\}$.*

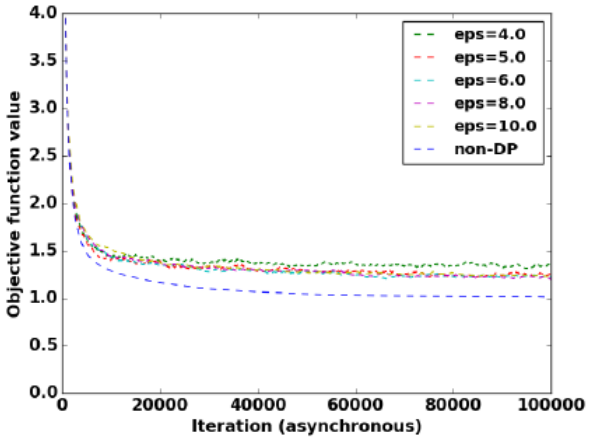
Theorem 2. *$\nabla_t^{(k)}$ guarantees $(2\epsilon, \delta_1 + \delta_2)$-differential privacy with respect to the data set $\{\mathbf{X}_t, \mathbf{y}_t\}$ for regression problem with $\mathbf{y}_{t,n_t} \in [+1, -1]$.*

- Xie et al. "Privacy-Preserving Distributed Multi-Task Learning with Asynchronous Updates." **KDD 2017**

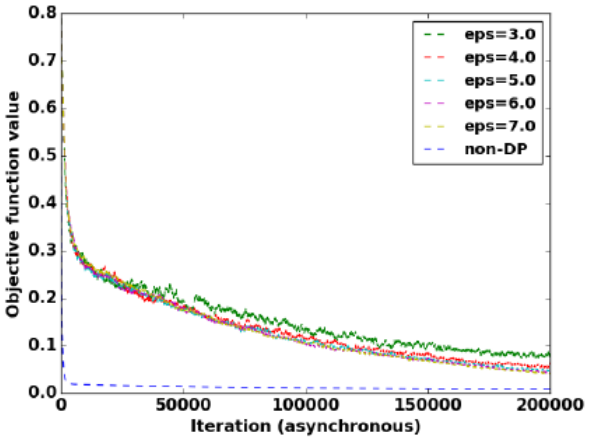
Experiments



(a) Synthetic (C)



(b) Alzheimer (C)



(c) Synthetic (R)

Table 2: Value of the objective function with respect to the number of iterations.

Road Map

- **Part I**: Introduction to Multi-Task Learning (MTL)
- **Part II**: Distributed MTL and Privacy Protection
- **Part III**: Interactive Multi-Task Learning
- **Part IV**: Future Directions of MTL

Task Relationship in MTL

- Most multi-task learning formulations assume certain task relationship
- We can alternatively *learn* such relationship from data [Zheng and Yeung 2010]

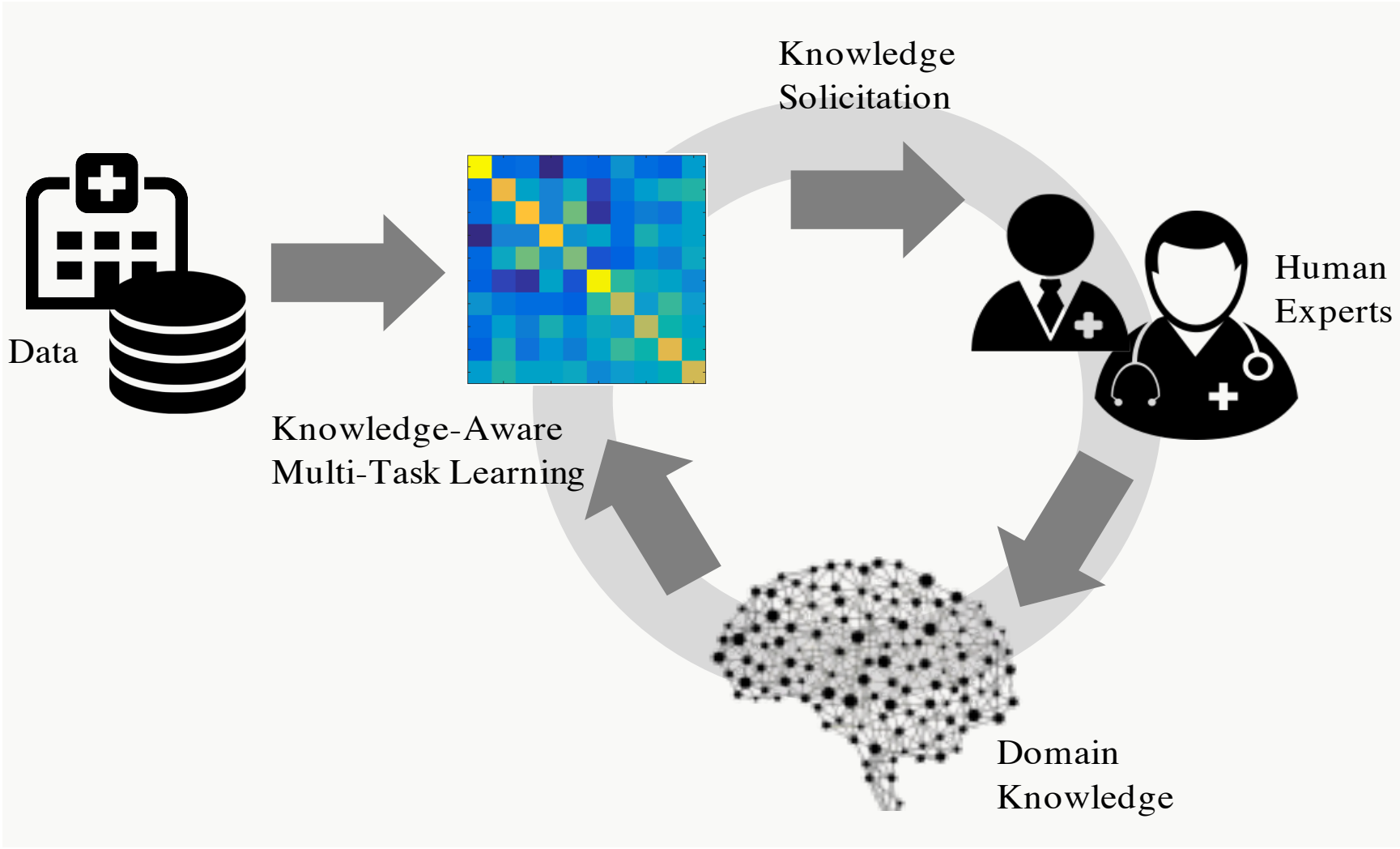
$$\min_{W,\Omega} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(w_i) + \frac{\lambda_1}{2} \mathrm{tr}(WW^T) + \frac{\lambda_2}{2} \mathrm{tr}(W\Omega^{-1}W^T)$$

$$\text{s.t. } \Omega \succeq 0, \mathrm{tr}(\Omega) = 1.$$

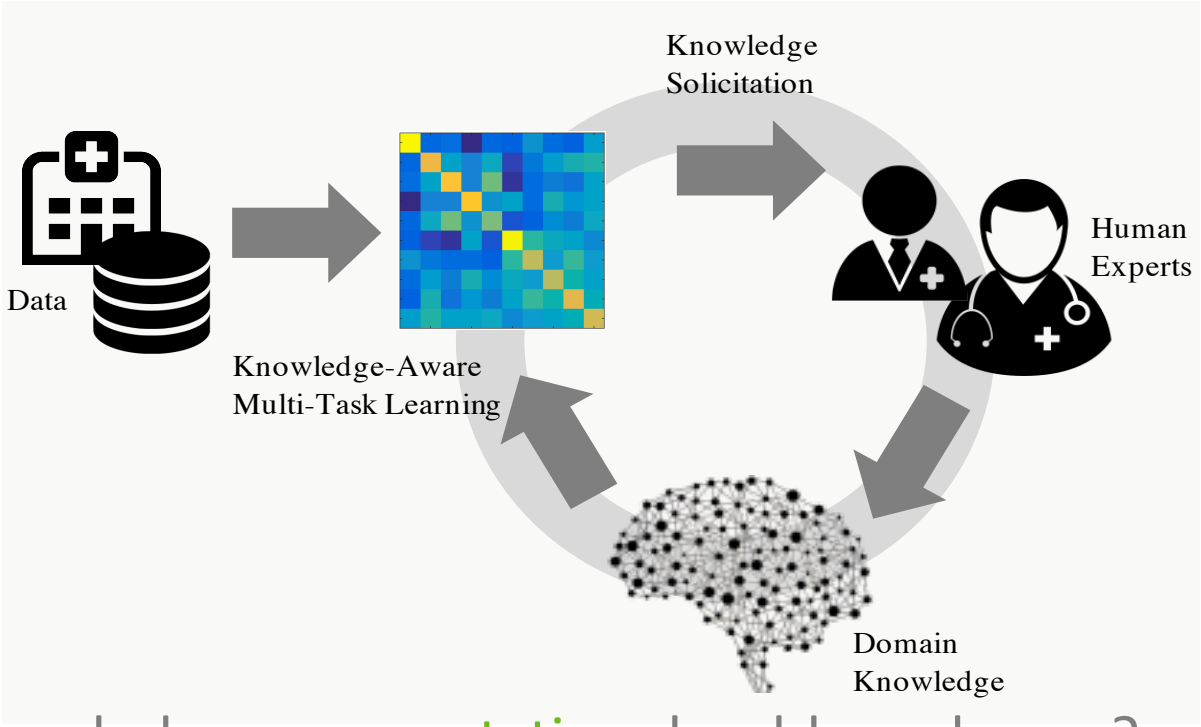
- Potential issues?

Zhang, Yu, and Dit-Yan Yeung. "A convex formulation for learning task relationships in multi-task learning." *UAI* 2010.

Incorporating Domain Knowledge via Interactive MTL

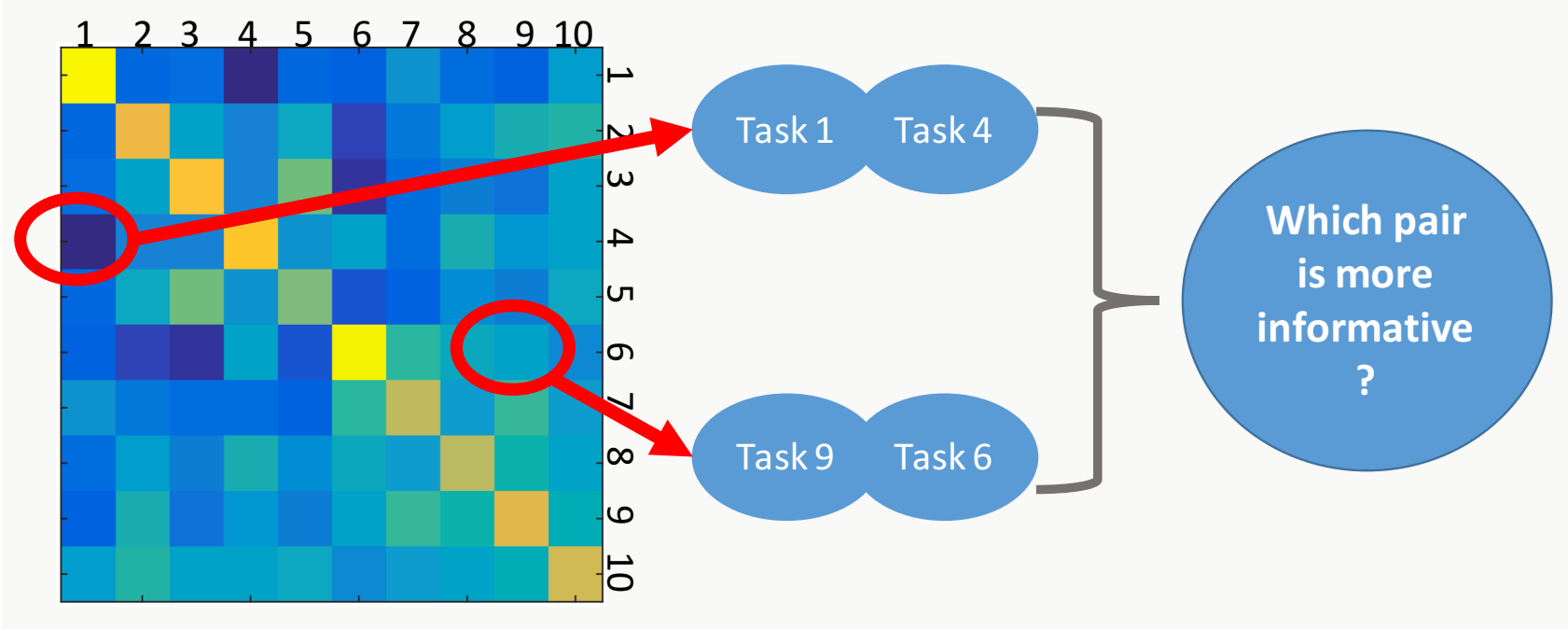


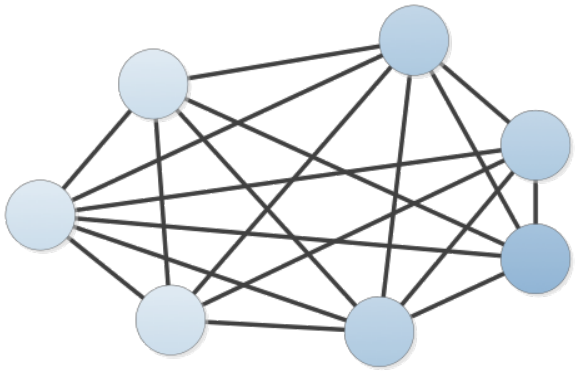
Incorporating Domain Knowledge via Interactive MTL



- Key Challenges:
  - What type of knowledge representation should we choose?
  - How to integrate domain knowledge to MTL algorithm?
  - How to solicit knowledge efficiently?

Partial Orders as Task Knowledge





$$\mathcal{T} = \{\mathbf{\Omega} : \mathbf{\Omega}_{i_1,j_1} \geq \mathbf{\Omega}_{i_2,j_2} \ \forall (i_1, j_1, i_2, j_2) \in S\}$$

Integrate Domain Knowledge in MTL

- Knowledge-aware MTL (kMTL) that considers constraints from task knowledge

$$\min_{W,\Omega} \sum_{i=1}^{t} \frac{1}{n_i} \ell_i(w_i) + \frac{\lambda_1}{2} \mathrm{tr}(WW^T) + \frac{\lambda_2}{2} \mathrm{tr}(W\Omega^{-1}W^T)$$
$$\text{s.t. } \Omega \succeq 0, \mathrm{tr}(\Omega) = 1, \Omega \in \mathcal{T}.$$

Integrate Domain Knowledge in MTL

- Solving via block coordinate descent
  - Step 1: Optimizing w.r.t. W when Ω is fixed
  - Step 2: Optimizing w.r.t. Ω when W is fixed
  - Step 3: Project Ω to the feasible set [2]

Theorem 1. *Suppose that* $\mathcal{T} = \{\mathbf{\Omega} : \mathbf{\Omega}_{i1,j1} \geq \mathbf{\Omega}_{i2,j2} + c\}$, *then, for any* $\mathbf{\Omega} \in \mathbb{R}^{K \times K}$, *the projection of* $\mathbf{\Omega}$ *to the convex set* $\mathcal{T}$ *is given by:*

$$\text{Proj}(\mathbf{\Omega}) = \mathbf{\Omega} \text{ if } \mathbf{\Omega} \in \mathcal{T},$$

otherwise

$$\text{Proj}(\mathbf{\Omega}) = \mathbf{\Omega}^* = \begin{cases} \mathbf{\Omega}^*_{i1,j1} = \frac{1}{2}(\mathbf{\Omega}_{i1,j1} + \mathbf{\Omega}_{i2,j2} + c) \\ \mathbf{\Omega}^*_{i2,j2} = \frac{1}{2}(\mathbf{\Omega}_{i1,j1} + \mathbf{\Omega}_{i2,j2} - c) \\ \mathbf{\Omega}^*_{p,q} = \mathbf{\Omega}_{p,q}, \ \forall (p,q) \neq (i1,j1) \text{ and } (i2,j2) \end{cases}$$

[2] Chang S, Qi G J, Aggarwal C C, et al. Factorized similarity learning in networks[C]//2014 IEEE International Conference on Data Mining. IEEE, 2014: 60-69.

Efficient Knowledge Solicitation

- After we obtain Ω from the model, we can present it to domain experts and check if Ω is reasonable.
  - Is that easy?
- In our work, we borrow the idea of active learning, and identify a few key elements from Ω.
  - Use the discrepancy between the learned Ω and task relationship inferred directly by W.
  - The selected pairs will be labeled by human experts

Interactive Multi-task Relationship Learning

Algorithm $(\mathbf{\Omega}, \mathbf{W}, \mathbf{b}) = \text{iMTRL}(\text{data, parameters})$

Require: Training sets $\{\mathbf{X}^k, \mathbf{y}^k\}_k^K$, number of selected queries $\mathbf{q}$.
regularization parameters λ_1, λ_2, positive number c, $\mathcal{T}^0 = \emptyset$

1: **for** $i = 1, ..., n$ **do**
2: $(\mathbf{\Omega}^i, \mathbf{W}^i, \mathbf{b}^i) = \text{kMTIL}(\{\mathbf{X}^k, \mathbf{y}^k\}_k^K, \mathcal{T}^{i-1}, \lambda_1, \lambda_2, c)$
3: $\mathcal{T}^i = \text{query}(\mathbf{W}^i, \mathbf{\Omega}^i, \mathbf{q}_i)$
4: $\mathcal{T}^i = \mathcal{T}^i \cup \mathcal{T}^{i-1}$
5: **end for**
6: $\mathbf{\Omega} = \mathbf{\Omega}^i$, $\mathbf{W} = \mathbf{W}^i$, $\mathbf{b} = \mathbf{b}^i$
7: **return** $\mathbf{\Omega}$, $\mathbf{W}$, $\mathbf{b}$

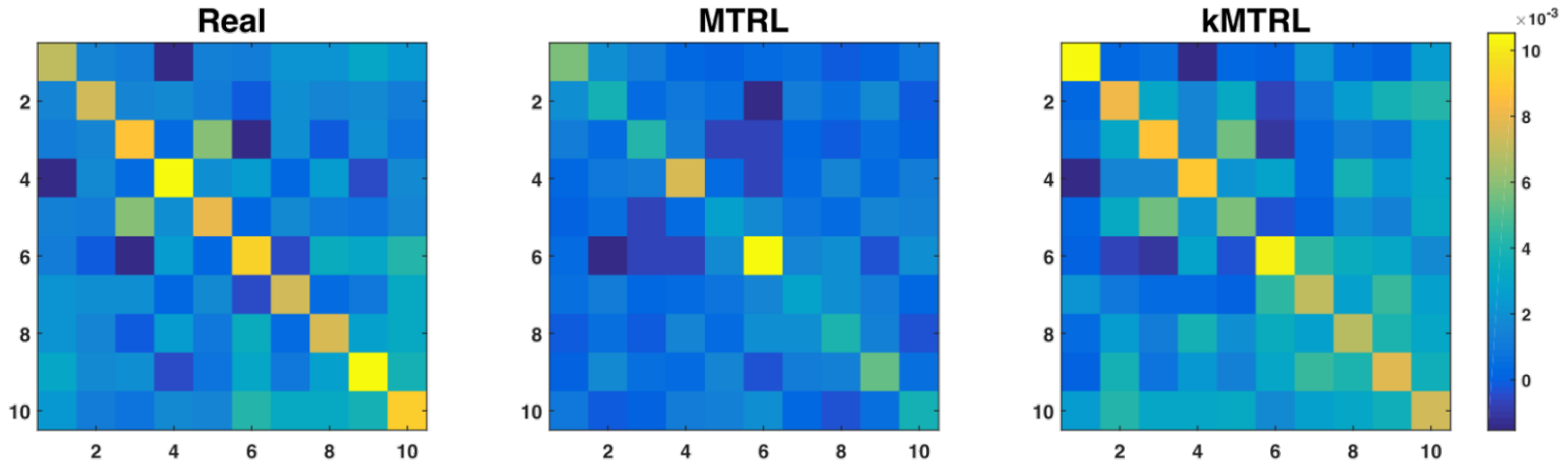
Experimental Results

TABLE II

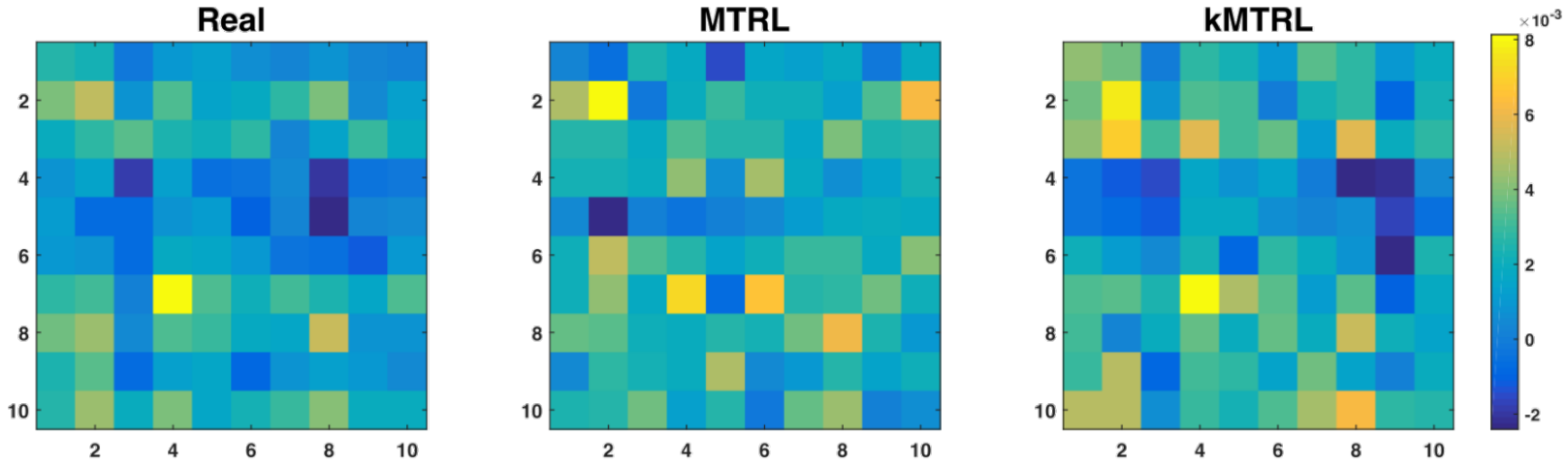
THE AVERAGE RMSE COMPARISON OF COMPETING METHODS ON THE SCHOOL DATASET AND MMSE DATASET. THE FIRST COLUMN IS THE PERCENTAGE OF TRAINING SAMPLES IN EACH TASK. THE KMTRL METHODS OUTPERFORMS ALL OTHER METHODS

School	RR	MTL-L	MTL-l21	MTRL	kMTRL-20	kMTRL-40	kMTRL-60	kMTRL-80
5%	1.1737±0.0041	1.1799± 0.0047	1.176± 0.0043	1.0615 ± 0.0167	1.0584 ± 0.0128	1.0553 ± 0.0155	**1.0551** ± 0.0158	1.0551 ± 0.0159
10%	1.1428±0.0306	1.1485 ± 0.0293	1.1477 ± 0.0282	0.9872 ± 0.0057	0.9823 ± 0.0030	0.9805 ± 0.0014	**0.9803** ± 0.0018	0.9803 ± 0.0018
15%	1.0665±0.0395	1.0699 ± 0.0405	1.0700 ± 0.0399	0.9491 ± 0.0060	0.9334 ± 0.0057	**0.9321** ± 0.0081	0.9322 ± 0.0083	0.9323 ± 0.0082
20%	0.9756±0.0157	0.9774 ± 0.0153	0.9776 ± 0.0149	0.9047 ± 0.0031	0.8966 ± 0.0123	0.8906 ± 0.0123	0.8844 ± 0.0022	**0.8843** ± 0.0019
MMSE	RR	MTL-L	MTL-l21	MTRL	kMTRL-5	kMTRL-10	kMTRL-15	kMTRL-20
2%	0.9503± 0.1467	0.9319±0.1497	0.9314±0.1693	0.9106 ± 0.0976	0.9113 ± 0.0982	**0.9058** ± 0.0926	0.9058 ± 0.0926	0.9058 ± 0.0926

Alzheimer's Disease Study



(a) Intra-region covariance

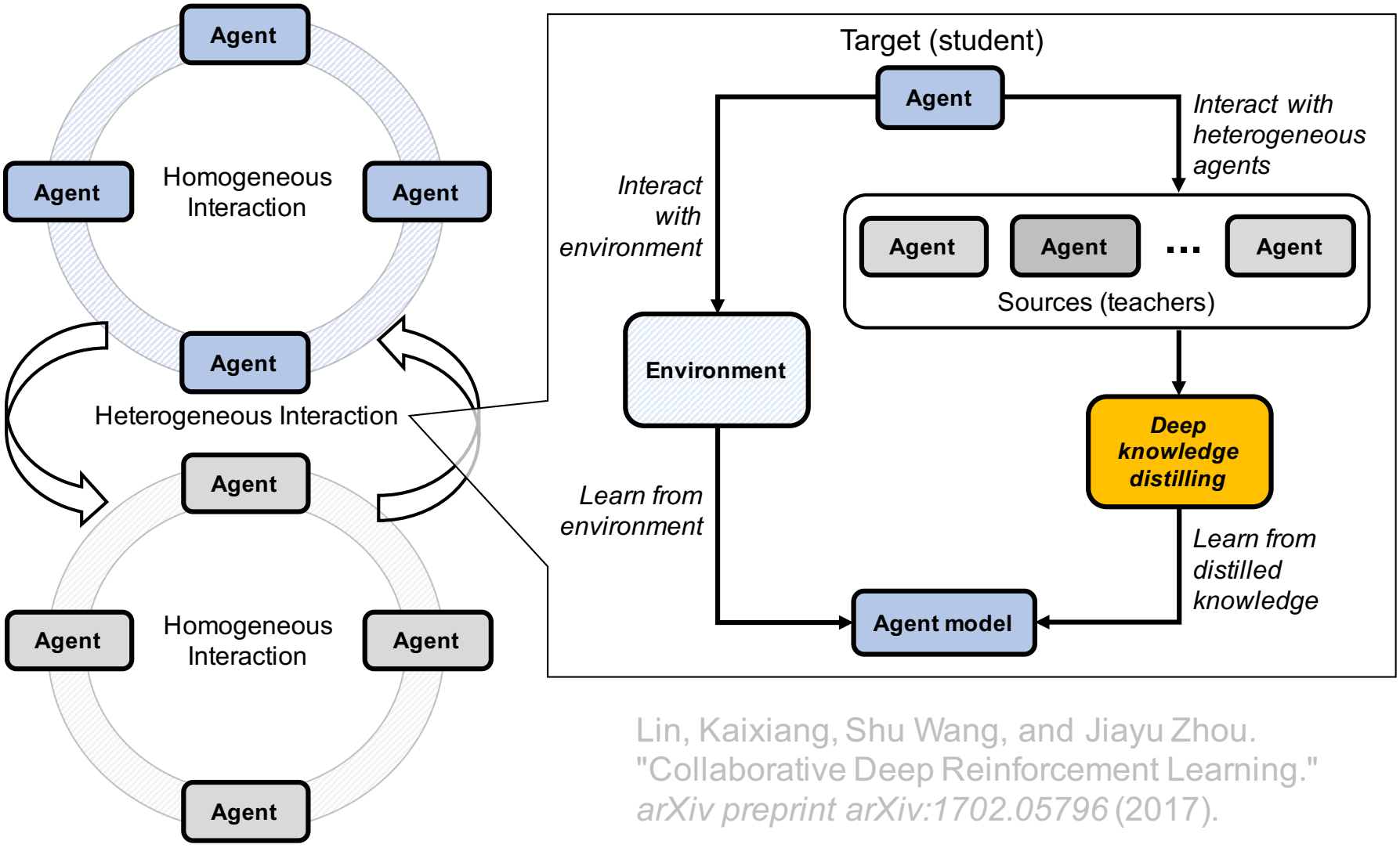


(b) Inter-region covariance

Road Map

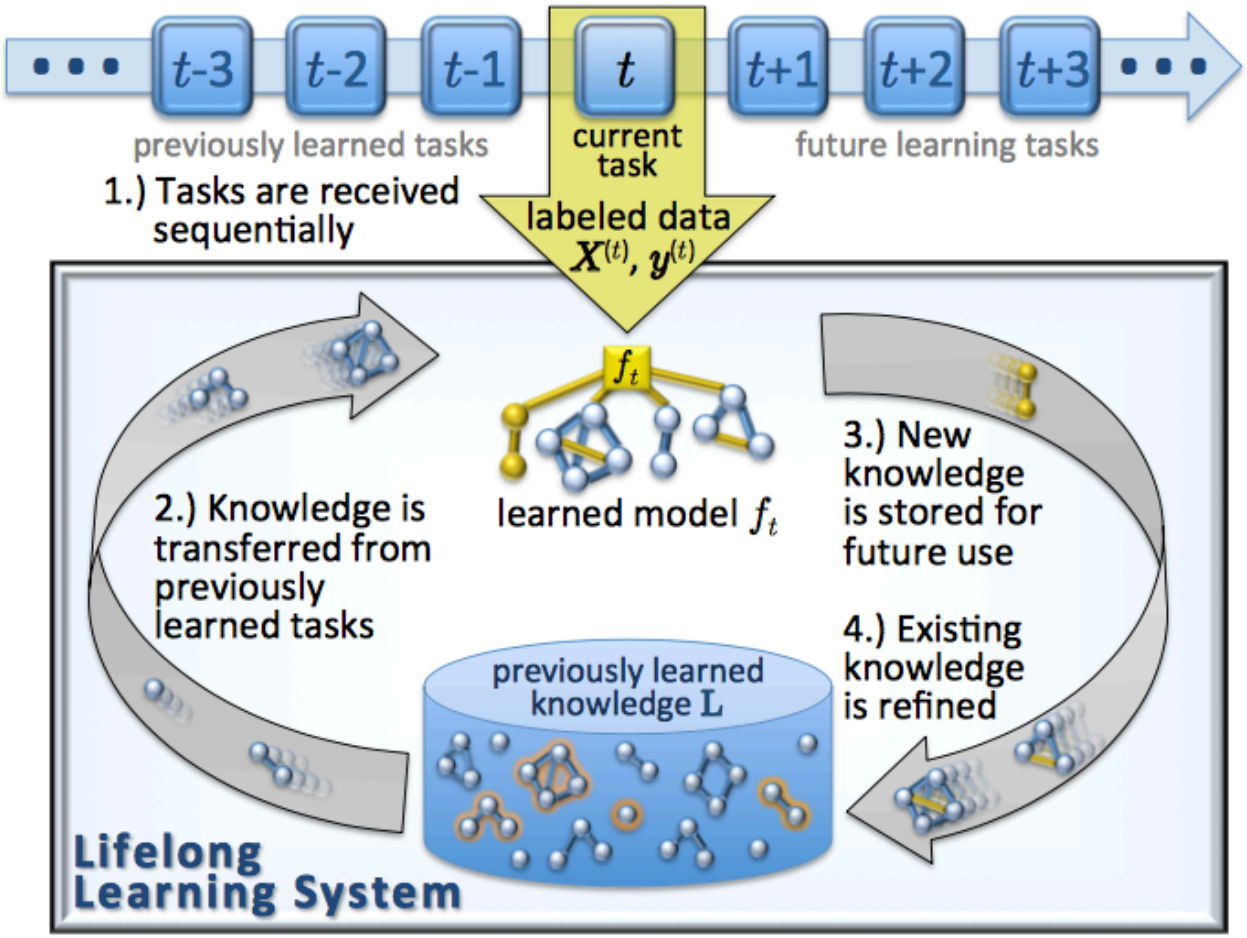
- **Part I**: Introduction to Multi-Task Learning (MTL)
- **Part II**: Distributed MTL and Privacy Protection
- **Part III**: Interactive Multi-Task Learning
- **Part IV**: Future Directions of MTL

MTL in Deep Reinforcement Learning



Lin, Kaixiang, Shu Wang, and Jiayu Zhou. "Collaborative Deep Reinforcement Learning." *arXiv preprint arXiv:1702.05796* (2017).

Life-Long Machine Learning



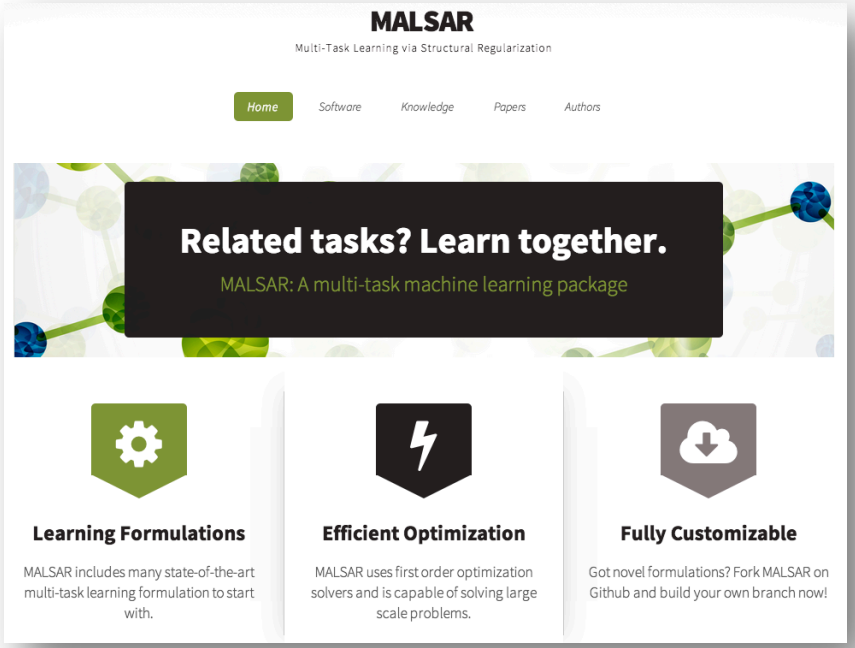
Ruvolo, Paul, and Eric Eaton. "ELLA: An Efficient Lifelong Learning Algorithm." *ICML (1)* 28 (2013): 507-515.

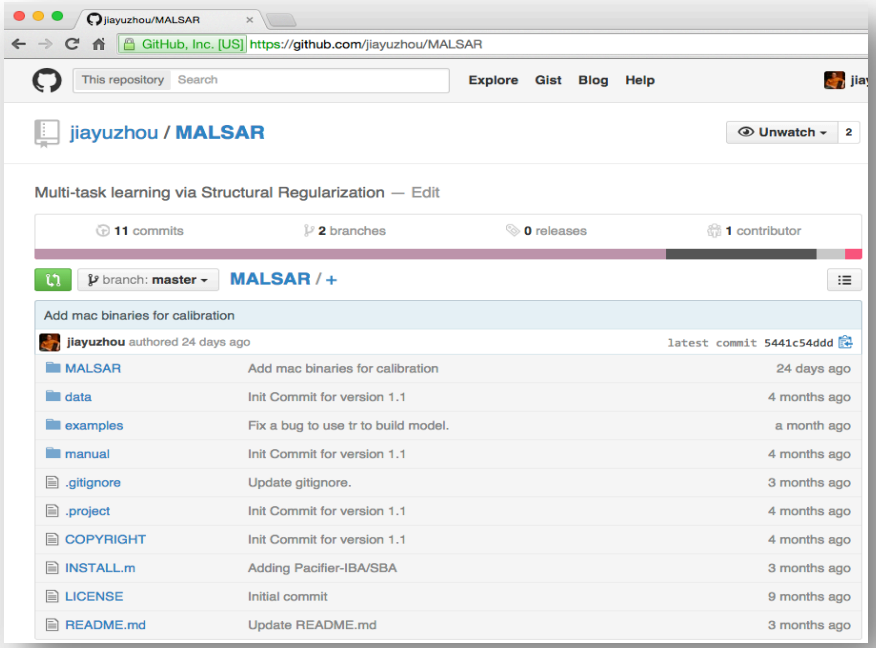
When to Transfer Knowledge?

- Transfer learning and multi-task learning assume that *learning tasks are related*.
- How do we know that tasks are indeed related?
  - So far there is no research done advising us *when NOT to transfer*
  - Intuitively, if the *cost* of modeling task relatedness is higher than the *benefit* of transfer, we should avoid transfer learning.
  - How do we quantify the cost?

MALSAR

MULTI-TASK LEARNING VIA STRUCTURAL REGULARIZATION

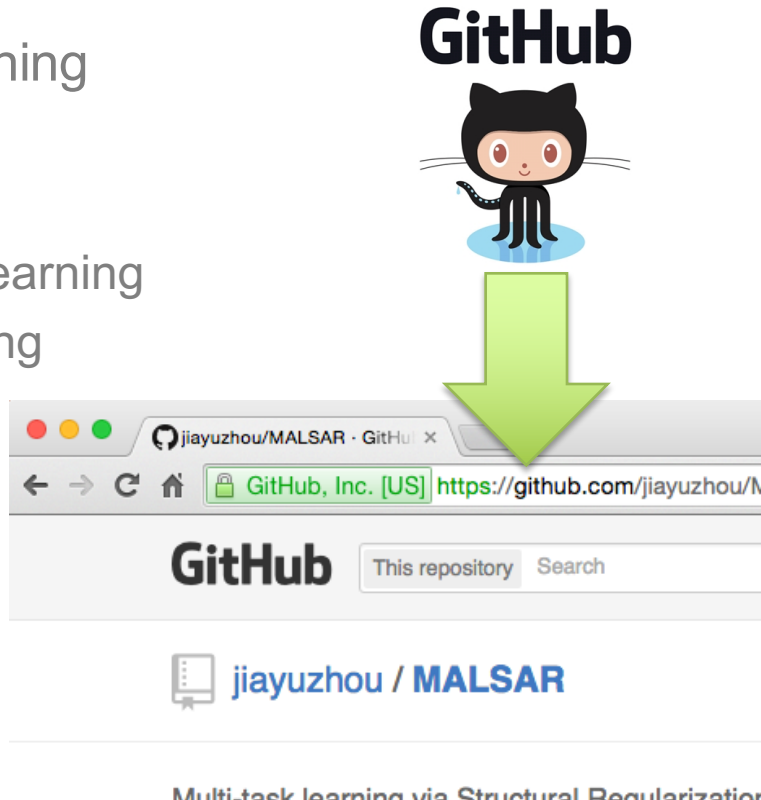




- Firstly introduced my MTL **tutorial** at **SDM** in 2012
- Over 40 research works using MALSAR are published in KDD, NIPS, TPAMI, ICCV, ICDM, ICIP, COLING, MICCAI, ACM-MM, etc.
- Used as **course material** to analyze compound profiling in the *Strasbourg Summer School* in France
- Supported by *two NSF grants (IIS-1615597, IIS-1565596)*. 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

Create a random MTL dataset

Invoke an MTL algorithm

MALSAR/example_Dirty.m

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

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

ILLIDAN Lab

MICHIGAN STATE UNIVERSITY

- Intelligent Data Analytics Lab @ MSU
  - *PhD Students*: Kaixiang Lin, Liyang Xie, Qi Wang, Mengying Sun, Inci M. Baytas, Andy Tang

Acknowledgement

- National Science Foundation
  - *IIS-1615597, IIS-1565596, EF-1638679*
- Office of Naval Research
  - *N00014-17-1-2265, N00014-14-1-0631*
- National Institutes of Health
  - *P30AG053760*
- NVidia Corporation

Thanks!