Learning Multiple Tasks with Multilinear Relationship Networks

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S. Yu

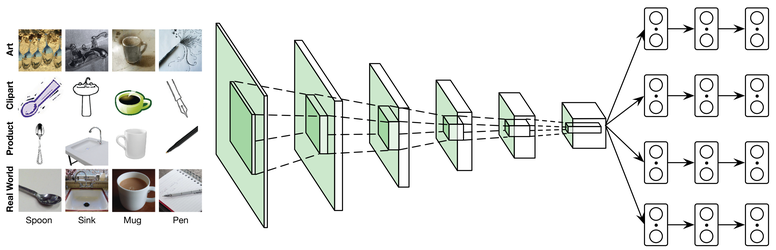
School of Software, National Engineering Lab for Big Data Software (NEL-BDS), Tsinghua University, China

Summary

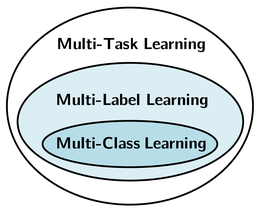
- A multilinear relationship network for multi-task deep learning
- Two main contributions:
  - Multilinear relationships across different tasks, features, and classes
  - Deep relationships across multiple task-specific layers of deep networks
- State-of-the-art results on standard object recognition datasets
- Open Problems
  - Efficient implementations for inverse and Kronecker product of tensors
  - Rank deficiency in covariance matrices of tasks, features, and classes
- Code available @ `https://github.com/thuml/Xlearn`

Multi-Task Deep Learning

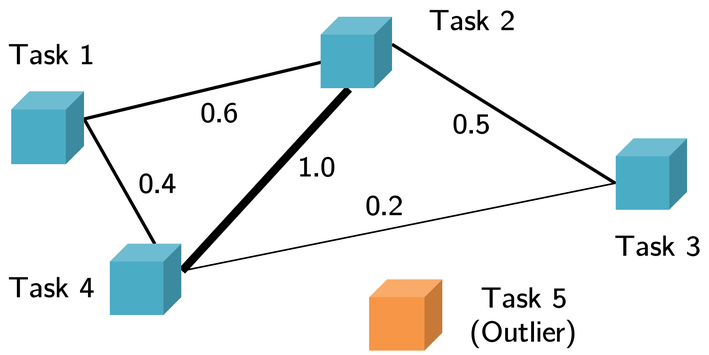
- Deep learning of multiple tasks under different distributions
  - Promote positive transfer across different tasks to tackle data scarcity
  - Circumvent negative transfer across two tasks when they are irrelevant



How to Model Task Relatedness?



- Multi-Task Learning
  - Model the **task relatedness** (task relationship)
  - Tasks have different data/features/distributions
- Multi-Label Learning
  - Model the label relatedness (label dependence)
  - Labels share the same data/features/distributions
- Multi-Class Learning
  - Learn all classes independently (one-vs-all applies)



Tensor Normal Distribution

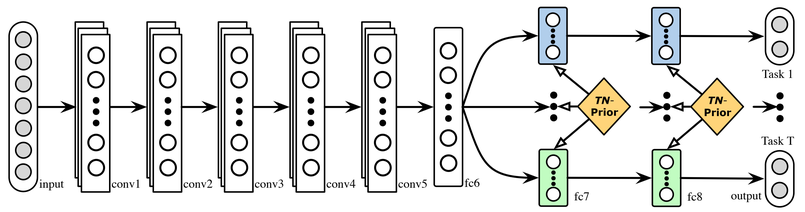
- Over order-K tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times \ldots \times d_K}$ of dimensions $(d_1, \ldots, d_K)$
- Mathematical Notation: $\mathcal{X} \sim \mathcal{TN}_{d_1 \times \ldots \times d_K}(\mathcal{M}, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K)$
- Vectorization form: $\mathrm{vec}(\mathcal{X}) \sim \mathcal{N}(\mathrm{vec}(\mathcal{M}), \mathbf{\Sigma}_1 \otimes \ldots \otimes \mathbf{\Sigma}_K)$
- Density function in vectorization form:

$$p(\mathbf{x}) = (2\pi)^{-d/2} \left(\prod_{k=1}^{K} |\mathbf{\Sigma}_k|^{-d/(2d_k)} \right) \times \exp\left(-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\top \mathbf{\Sigma}_{1:K}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right) \tag{1}$$

where $|\cdot|$ is the determinant of the square matrix, and $\mathbf{x} = \mathrm{vec}(\mathcal{X})$, $\boldsymbol{\mu} = \mathrm{vec}(\mathcal{M})$, $\mathbf{\Sigma}_{1:K} = \mathbf{\Sigma}_1 \otimes \ldots \otimes \mathbf{\Sigma}_K$, $d = \prod_{k=1}^{K} d_k$

Multilinear Relationship Network (MRN)

- Data $\{\mathcal{X}_t, \mathcal{Y}_t\}_{t=1}^{T}$ and $\mathcal{X}_t = \{\mathbf{x}_1^t, \ldots, \mathbf{x}_{N_t}^t\}$, $\mathcal{Y}_t = \{\mathbf{y}_1^t, \ldots, \mathbf{y}_{N_t}^t\}$
- T tasks, each of N_t examples and $\mathbf{x}_n^t \in \mathbb{R}^D$, $\mathbf{y}_n^t \in \{1, \ldots, C\}$
- Layer-ℓ parameter tensor $\mathcal{W}^\ell = [\mathbf{W}^{1,\ell}; \ldots; \mathbf{W}^{T,\ell}] \in \mathbb{R}^{D_1^\ell \times D_2^\ell \times T}$
  - Mode-1 (row of tensor): features (D_1^ℓ in total)
  - Mode-2 (column of tensor): classes for the classifier layer (D_2^ℓ in total)
  - Mode-3 (layer of tensor): tasks (T in total)
- Goal: model all relationships across features, classes and tasks



$$p(\mathcal{W} | \mathcal{X}, \mathcal{Y}) \propto \prod_{\ell \in \mathcal{L}} p(\mathcal{W}^\ell) \cdot \prod_{t=1}^{T} \prod_{n=1}^{N_t} p(\mathbf{y}_n^t | \mathbf{x}_n^t, \mathcal{W}^\ell) \tag{2}$$

- Tensor normal prior: $p(\mathcal{W}^\ell) = \mathcal{TN}_{D_1^\ell \times D_2^\ell \times T}(\mathbf{O}, \mathbf{\Sigma}_1^\ell, \mathbf{\Sigma}_2^\ell, \mathbf{\Sigma}_3^\ell)$, where $\mathbf{\Sigma}_1^\ell \in \mathbb{R}^{D_1^\ell \times D_1^\ell}$, $\mathbf{\Sigma}_2^\ell \in \mathbb{R}^{D_2^\ell \times D_2^\ell}$, and $\mathbf{\Sigma}_3^\ell \in \mathbb{R}^{T \times T}$ are the mode-1 (feature), mode-2 (class), mode-3 (task) covariances
- Final optimization of multilinear relationship network (MRN):

$$\min_{f_t|_{t=1}^T, \mathbf{\Sigma}_k^\ell|_{k=1}^K} \sum_{t=1}^{T} \sum_{n=1}^{N_t} J(f_t(\mathbf{x}_n^t), \mathbf{y}_n^t) + \frac{1}{2} \sum_{\ell \in \mathcal{L}} \left(\mathrm{vec}(\mathcal{W}^\ell)^\top (\mathbf{\Sigma}_{1:K}^\ell)^{-1} \mathrm{vec}(\mathcal{W}^\ell) - \sum_{k=1}^{K} \frac{D^\ell}{D_k^\ell} \ln\left(|\mathbf{\Sigma}_k^\ell|\right) \right) \tag{3}$$

Learning Algorithm

- Update $\mathcal{W}$ by the back-propagation algorithm:

$$\frac{\partial O(\mathbf{x}_n^t, \mathbf{y}_n^t)}{\partial \mathbf{W}^{t,\ell}} = \frac{\partial J(f_t(\mathbf{x}_n^t), \mathbf{y}_n^t)}{\partial \mathbf{W}^{t,\ell}} + \left[(\mathbf{\Sigma}_{1:3}^\ell)^{-1} \mathrm{vec}(\mathcal{W}^\ell) \right]_{\cdot \cdot t} \tag{4}$$

- Update $\mathbf{\Sigma}$ by the *flip-flop* algorithm:

$$\begin{aligned} \mathbf{\Sigma}_1^\ell &= \frac{1}{D_2^\ell T} (\mathcal{W}^\ell)_{(1)} (\mathbf{\Sigma}_3^\ell \otimes \mathbf{\Sigma}_2^\ell)^{-1} (\mathcal{W}^\ell)_{(1)}^\top + \epsilon \mathbf{I}_{D_1^\ell}, \\ \mathbf{\Sigma}_2^\ell &= \frac{1}{D_1^\ell T} (\mathcal{W}^\ell)_{(2)} (\mathbf{\Sigma}_3^\ell \otimes \mathbf{\Sigma}_1^\ell)^{-1} (\mathcal{W}_\ell)_{(2)}^\top + \epsilon \mathbf{I}_{D_2^\ell}, \\ \mathbf{\Sigma}_3^\ell &= \frac{1}{D_1^\ell D_2^\ell} (\mathcal{W}^\ell)_{(3)} (\mathbf{\Sigma}_2^\ell \otimes \mathbf{\Sigma}_1^\ell)^{-1} (\mathcal{W}^\ell)_{(3)}^\top + \epsilon \mathbf{I}_T. \end{aligned} \tag{5}$$

$\mathcal{W}_{(k)}$ is the mode-k matricization of tensor $\mathcal{W}$, where row i of $\mathcal{W}_{(k)}$ contains all elements of $\mathcal{W}$ with the k-th index equal to i

- Speed up computation by Kronecker property and SVD:

$$\begin{aligned} (\Sigma_3^\ell)_{ij} &= \frac{1}{D_1^\ell D_2^\ell} (\mathcal{W}^\ell)_{(3),i\cdot} (\mathbf{\Sigma}_2^\ell \otimes \mathbf{\Sigma}_1^\ell)^{-1} (\mathcal{W}^\ell)_{(3),j\cdot}^\top + \epsilon I_{ij} \\ &= \frac{1}{D_1^\ell D_2^\ell} (\mathcal{W}^\ell)_{(3),i\cdot} \mathrm{vec}\left((\mathbf{\Sigma}_1^\ell)^{-1} \mathcal{W}_{\cdot\cdot j}^\ell (\mathbf{\Sigma}_2^\ell)^{-1} \right) + \epsilon I_{ij}, \end{aligned} \tag{6}$$

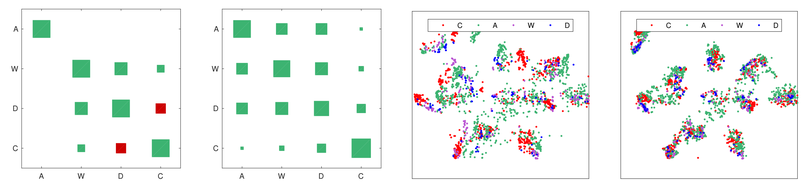
Experimental Results

Table: Accuracy on *Office-Caltech* with standard evaluation protocol (AlexNet)

Method	5%					10%					20%				
	A	W	D	C	Avg	A	W	D	C	Avg	A	W	D	C	Avg
STL (AlexNet)	88.9	73.0	80.4	88.7	82.8	92.2	80.9	88.2	88.9	87.6	91.3	83.3	93.7	**94.9**	90.8
MTFL	90.0	78.9	90.2	86.9	86.5	92.4	85.3	89.5	**89.2**	89.1	93.5	89.0	95.2	92.6	92.6
RMTL	91.3	82.3	88.8	**89.1**	87.9	92.6	85.2	93.3	87.2	89.6	94.3	87.0	96.7	93.4	92.4
MTRL	86.4	83.0	95.1	**89.1**	88.4	91.1	87.1	97.0	87.6	90.7	90.0	88.8	99.2	94.3	93.1
DMTL-TF	91.2	88.3	92.5	85.6	89.4	92.2	91.9	97.4	86.8	92.0	92.6	97.6	94.5	88.4	93.3
MRN$_8$	91.7	96.4	96.9	86.5	92.9	92.7	97.1	97.3	86.6	93.4	93.2	96.9	99.4	82.8	94.4
MRN$_t$	91.1	96.3	97.4	86.1	92.7	92.5	97.7	96.6	86.7	93.4	91.9	96.6	95.9	90.0	93.6
MRN (full)	**92.5**	**97.5**	**97.9**	87.5	**93.8**	**93.6**	**98.6**	**98.6**	87.3	**94.5**	**94.4**	**98.3**	**99.9**	89.1	**95.5**

Table: Accuracy on *Office-Home* with standard evaluation protocol (VGGnet)

Method	5%					10%					20%				
	A	C	P	R	Avg	A	C	P	R	Avg	A	C	P	R	Avg
STL (VGGnet)	35.8	31.2	67.8	62.5	49.3	51.0	40.7	75.0	68.8	58.9	56.1	54.6	80.4	71.8	65.7
MTFL	40.1	30.4	61.5	59.5	47.9	50.3	35.0	66.3	65.0	54.2	55.2	38.8	69.1	70.0	58.3
RMTL	42.3	32.8	62.3	60.6	49.5	49.7	34.6	65.9	64.6	53.7	55.2	39.2	69.6	70.5	58.6
MTRL	42.7	33.3	62.9	61.3	50.1	51.6	36.3	67.7	66.3	55.5	55.8	39.9	70.2	71.2	59.3
DMTL-TF	49.2	34.5	67.1	62.9	53.4	57.2	42.3	73.6	69.9	60.8	58.3	**56.1**	79.3	72.1	66.5
MRN$_8$	52.7	34.7	70.1	67.6	56.3	59.1	42.7	75.1	72.8	62.4	58.4	55.6	80.4	72.4	66.7
MRN$_t$	52.0	34.0	69.9	66.8	55.7	58.6	42.6	74.9	72.4	62.1	57.7	54.8	80.2	71.6	66.1
MRN (full)	**53.3**	**36.4**	**70.5**	**67.7**	**57.0**	**59.9**	**42.7**	**76.3**	**73.0**	**63.0**	**58.5**	55.6	**80.7**	**72.8**	**66.9**



(a) MTRL Relationship (b) MRN Relationship (c) DMTL-TF Features (d) MRN Features

Figure: Hinton diagram of task relationships (a)(b) and t-SNE of features (c)(d).