TOMATO: A Topic-Wise Multi-Task Sparsity Model

Jason (Jiasheng) Zhang
The Pennsylvania State University, USA
jpz5181@psu.edu

Dongwon Lee
The Pennsylvania State University, USA
dongwon@psu.edu

ABSTRACT

The *Multi-Task Learning (MTL)* leverages the inter-relationship across tasks and is useful for applications with limited data. Existing works articulate different task relationship assumptions, whose validity is vital to successful multi-task training. We observe that, in many scenarios, the inter-relationship across tasks varies across different groups of data (i.e., topic), which we call *within-topic task relationship hypothesis*. In this case, current MTL models with homogeneous task relationship assumption cannot fully exploit different task relationships among different groups of data. Based on this observation, in this paper, we propose a generalized topic-wise multi-task architecture, to capture the within-topic task relationship, which can be combined with any existing MTL designs. Further, we propose a new specialized MTL design, topic-task-sparsity, along with two different types of sparsity constraints. The architecture, combined with the topic-task-sparsity design, constructs our proposed TOMATO model. The experiments on both synthetic and 4 real-world datasets show that our proposed models consistently outperform 6 state-of-the-art models and 2 baselines with improvement from 5% to 46% in terms of task-wise comparison, demonstrating the validity of the proposed within-topic task relationship hypothesis. We release the source codes and datasets of TOMATO at: https://github.com/JasonLC506/MTSEM.

CCS CONCEPTS

• **Computing methodologies** → **Multi-task learning**.

KEYWORDS

multi-task; topic-wise; sparsity

ACM Reference Format:
Jason (Jiasheng) Zhang and Dongwon Lee. 2020. TOMATO: A Topic-Wise Multi-Task Sparsity Model . In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3340531.3411972

1 INTRODUCTION

The development of advanced machine learning techniques (e.g., deep learning) often requires a large amount of labeled samples to train a good model. However, this requirement is hard to meet for many applications due to the prohibitive cost of data collection and labeling. To mitigate this problem, the *Multi-Task Learning (MTL)* approach takes an advantage of multiple related tasks to facilitate the training of some or all of the tasks that have limited training samples [21]. It has been successfully applied to many learning problems in domains such as computer vision [22, 23, 25, 27, 34, 36] and natural language processing [5, 9, 19, 29, 30].

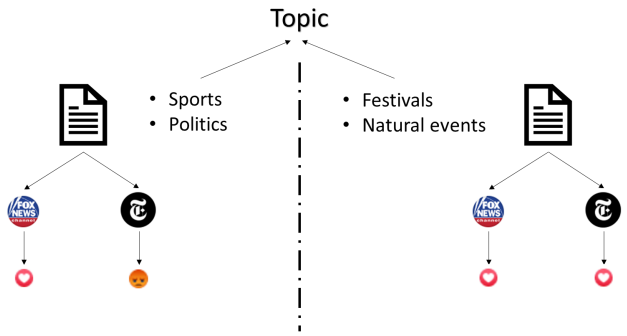


Figure 1: Illustration of EXAMPLE 1, different news topics result in different relationships between readers' reactions to them in different news channels. Icons are adopted from the Fox News channel www.facebook.com/FoxNews/ and the New York Times channel www.facebook.com/nytimes in Facebook.

The principle of MTL is to leverage the relationship assumptions among tasks through a model design–e.g., commonalities across tasks. Some well-known MTL design categories are feature selection [10, 12, 26], where tasks shared a feature-wise sparsity structure, task structure [8, 11, 16, 17, 37, 38], where model parameters of different tasks share common structures, and the low rank structure of model parameters of tasks [3, 13, 14] in linear models [35], and parameter sharing [7, 18, 32] and information sharing [20, 22, 23] in neural network models [24]. Each of the above designs corresponds to an assumption of the task relationship. The validity of the task-relationship assumption in these models is vital to achieve successful learning.

However, we observe that, such a task relationship used in previous methods does not always hold. More specifically, a task relationship can often hold only *within topics*–i.e., commonalities across tasks hold only for certain topics (or groups) of data. Consider the following two motivating examples.

Example 1 (Predicting User Emotions, Fig. 1). *Consider the problem to accurately predict news readers' reactions (e.g., LIKE, ThumbsDown) toward news posts from different news channels (e.g., NYT, Wapo, Fox). To overcome insufficient data per new channel, one models the problem as MTL (i.e., news channels as tasks), assuming that readers' reactions across tasks be similar. However, in practice,*

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3411972

such an assumption on the task relationship may not hold. For instance, readers' reactions can be highly consistent across different news channels for news on the topics of natural events and festivals; however, different standing points of different channels often result in exactly opposite readers' reactions for the news on the topics of sports and political news.

Example 2 (Searching Relevant Products). *In e-commerce applications, consider a problem of searching products for different user groups. For instance, both male and female users (i.e., different user groups as tasks) may have similar taste for products related to food (i.e., topic), but different taste for books or music (i.e., topic). In this case, considering the same task relationship across all products will either miss the similarity (i.e., treating two tasks as independent) or cause negative knowledge transfer (i.e., treating two tasks the same).*

Based on these observations, therefore, we propose a "within-topic" task relationship hypothesis to reveal the data-dependent task relationship. This hypothesis assumes that task relationship may appear different within data if from different topics. The topics are determined by input features of data (clusters of data), different from task groups in within-group clustering design [13]. Compared with the recent works [20, 23] on data-dependent task relationship, with the clear notion of topics, the data dependency and task relationship can be "decoupled" here, which enables the application of any existing task relationship designs to reveal within-topic task relationship. In this work, therefore, we propose a topic-wise multi-task architecture using a topic module to distribute data from different topics to different modules, so that different task relationship can be learned. Within each topic, we propose two topic-task-sparsity constraints to enforce a multi-task sparsity structure for task relationship, where only a few tasks are allowed to deviate from a global structure shared by all other tasks. This multi-task sparsity structure is consistent with the aforementioned example, where only a few news channels are different from the others per topic.

Our contributions can be summarized as follows:

(1) We propose the within-topic task relationship hypothesis for the MTL problem;
(2) we propose a topic-wise multi-task architecture based on the hypothesis;
(3) we propose two types of topic-task sparsity constraints, topic-task-element and topic-task-exclusive and the optimization algorithms with proof;
(4) the proposed topic-wise multi-task sparsity model consistently outperforms state-of-the-art MTL models in experiments on both synthetic and real world datasets.

The remaining of the paper is organized as following: we first introduce related works. The problem and the hypothesis are described next. After that, We present the detail of the model design and its optimization. Finally, the experiment results are presented, followed by the conclusion.

2 RELATED WORK

In this section, we review related works on linear MTL models, MTL neural networks and sparsity constraints used in neural networks.

We summarize the existing MTL works both based on linear models and neural network and the proposed TOMATO for comparison in Table. 1.

There are a lot of works on linear MTL models. Interested readers are referred to [35] for a comprehensive survey. Those models are designed based on different assumptions of task relationships. More specifically, [10, 12, 26] assume different tasks share similar sparse feature selection pattern. [8, 11, 16, 17, 37, 38] assume that the weight vectors. With similar spirit of above task structure assumption,[3, 13, 14] directly assume that the weight matrix should be low-rank, which enforce different tasks to share the same low-dimension feature transformation. Though the simplicity of the linear structure provides such flourishing of MTL designs, it is less flexible compared with neural network models.

The neural network MTL models are based on two designs, parameter sharing and information sharing. The most common shared-bottom model is similar to the feature selection design in linear MTL models. Built upon the shared-bottom design, [7, 18, 32] propose further constraints on parameter sharing. Unique for neural network MTL models is information sharing [22], where cross-stitch structures are used to enable information flows from one task to another. Though neural network provides more flexibility of model design, as the information sharing, task relationship still relies only on design assumptions but not further information.

There are two recent works [20, 23], whose task-specific gates can be considered as data-dependent task relationship design. The distribution of weights given to different experts by different tasks are determined by the inputs. When such distributions of two tasks given a group of input samples are similar, those two tasks are related and vice versa. However, both data-dependency and task relationship are modeled by the weights of different tasks, which excludes the application of more flexible task relationship designs. Moreover, it can be seen later that MMoE [20] can be seen as a special instantiation of our proposed architecture.

Many task relationships in linear MTL models are achieved by constraints over weight matrix, especially sparsity constraints (e.g., $l_{1,q}$ penalty). Within neural network models, the sparsity constraints are recently applied to model compression [1, 4, 28, 33]. For example, [1, 28] use group sparsity ($l_{1,q}$) loss to zero-out the entire neurons to learn a sparse model for both memory and computation efficiency. [33] combines both group sparsity ($l_{2,1}$) and exclusive penalty ($l_{1,2}$). In this work, we adopt group sparsity as topic-task-element penalty ($l_{1,1,2}$) and propose group exclusive penalty as topic-task-exclusive penalty ($l_{2,1,2}$), together with its optimization algorithm.

3 PROBLEM DEFINITION

We formally define the multi-task learning (MTL) problem.

Definition 1 (Multi-Task Learning Problem). *Given T tasks, for each task $t \in [T]$, there are N_t samples (X_t, Y_t), with each $x_t \in \mathbf{R}^{d_t}$ as input feature and $y_t \in \mathbf{R}^{p_t}$ as labels. Here in this work, we take homogeneous MTL setting, where the dimensions and types of the features and labels for different tasks are the same, respectively, that is, for $\forall t \in [T]$, $d_t = d$ and $p_t = p$. Then, the MTL problem is to*

Model	Task Relationship Assumption (MTL design)	Data Dependency
Linear Models [35]	feature selection	N.A.
	task structure	N.A.
	low rank	N.A.
Shared-bottom	shared feature extraction	N.A.
Inter-task-l_2 [7]	weights similarity by l_2 constraint	N.A.
DMTRL [32], MRN [18]	low rank	N.A.
Cross-stitch [22]	cross-task communication	N.A.
MMoE [20], Routing [23]	weights of shared experts	weights of shared experts
TOMATO	topic-task-sparsity	topic-wise multi-task architecture

Table 1: Summary of existing MTL works and the proposed TOMATO.

find a mapping $f : \mathbf{R}^d \times [T] \mapsto \mathbf{R}^p$*, such that the overall cost*

$$\mathcal{L} = \sum_{t \in [T]} \frac{1}{N_t} \sum_{n_t \in [N_t]} L(f(x_t, t), y_t)$$

is minimized.

The proposed within-topic task relationship hypothesis can be formally defined as follows:

Definition 2 (Within-Topic Task Relationship Hypothesis). *Given each sample input* x*, there is a topic* $h(x)$ *given by* $h : \mathbf{R}^d \mapsto [K]$*, where* K *is the number of topics. The prediction function* $f : \mathbf{R}^d \times [T] \mapsto \mathbf{R}^p$ *can be decomposed as* $f(x, t) = g(h(x), x, t)$*. Within each topic* $k \in [K]$*,* $g(k, ., .)$ *shows the task relationship between each* $g(k, ., t_1)$ *and* $g(k, ., t_2)$ *with* $t_1 \neq t_2$*.*

4 TOPIC-WISE MULTI-TASK SPARSITY MODEL

In this section, we describe the proposed topic-wise multi-task sparsity model. First, the topic-wise multi-task architecture is described as the overview of the model, which can be combined with any existing MTL design as within-topic task relationship. Second, the two sparsity constraints are introduced for within-topic task relationship. Third, the optimization algorithm is described.

4.1 Topic-Wise Multi-Task Architecture

The topic-wise multi-task architecture is designed based on the within-topic task relationship hypothesis. Specifically, given input x, it is cast by a set of topic-task-specific functions $\{g(k, x, t) \| k \in [K]\}$ into the topic-task-specific hidden layers, and the task-specific layer afterward is obtained by aggregating topic-task-specific layers over different topics weighted by topic distribution $h(x)$ such that $\sum_k h(x)_k = 1$, which can be formulated as

$$f(x, t) = \sum_k h(x)_k g(k, x, t). \tag{1}$$

When a task relationship is enforced in topic-task-specific functions $\{g(k, x, t)\}$ within each topic k, the topic-wise multi-task architecture reveals Definition. 2. Compared with the existing shared-bottom architecture (Fig. 2), the topic module $h(x)$ distributes data samples to different within-topic task relationship, rather than all data with the same task relationship. This clearer task relationship within each topic leads to more compact structure of $g(k, x, t)$ (i.e., low-rank structure, parameter sharing), compensating the redundancy by the extra topic dimension and boosts the performance.

We compare the proposed architecture to the recent MMoE work [20], that models data-dependent task relationship. From its viewpoint, our work decouples the data-dependent task relationship into data-dependence ($h(x)$) and within-topic task relationship ($g(k, x, t)$), which enables the application of all existing task relationship designs for the latter. To see this, if we choose the factorization structure (DMTRL) [32] for within-topic task relationship, $g(k, x, t) = \sum_e p(k, t)_e q(x, e)$, Eq. 1 becomes $f(x, t) = \sum_k \sum_e h(x)_k p(k, t)_e q(x, e)$. Compared with Eq. 7 in [20], MMoE can be seen as a special instantiation of the proposed architecture by setting $gate(x, t)_e = \sum_k h(x)_k p(k, t)_e$.

4.2 Topic-Task Sparsity

In this subsection, we describe a new MTL design, called *topic-task sparsity*, to capture task relationship with the help of the proposed topic-wise multi-task architecture.

We assume that, within each topic, only a few tasks (news channels) may deviate from the majority. For example, within political topic, the readers' reactions to similar posts under extreme conservative or liberal news channels are usually different from those under the majority milder channels. We proposed the topic-task-sparsity design that

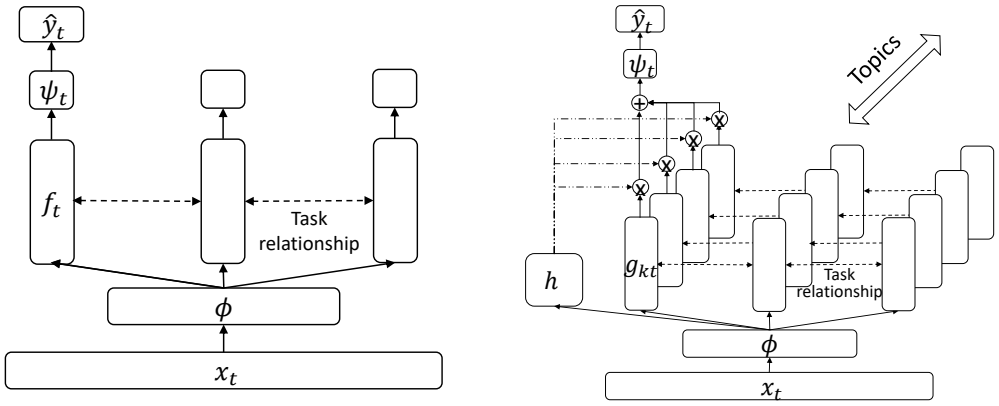
$$\theta_{k,t} = \theta^0 + \theta^s_{k,t}, \tag{2}$$

where $\theta_{k,t}$ is the vector of the parameters of topic-task-specific function $g(k, x, t) = g(x|\theta_{k,t})$, θ^0 is the global parameters that shared by different topics k and tasks t, and $\theta^s_{k,t}$ is the topic-task-sparse part of the parameters. We note Θ^s as the tensor combining $\theta^s_{k,t}$ for all topics and tasks.

To enforce sparsity structure in Θ^s, we proposed two types of topic-task-sparsity constraints $\Omega(\Theta^s)$. First, an element-wise sparsity structure is assumed for Θ^s, which is enforced by topic-task-element constraint defined as

$$\Omega^{el}(\Theta^s) = \sum_k \sum_t ||\theta^s_{k,t}||_2, \tag{3}$$

where $||.||_q$ is the l_q norm. The entire topic-task-element constraint $\Omega^{el}()$ is a $l_{2,1,1}$ norm, which is also known as group sparsity constraints. It is used in [1, 28] to zero out entire neurons for compression. Here, similar property is used to enforce certain topic-task-specific parameters $\theta_{k,t}$ to be the same as the global ones θ^0. The



(a) Existing neural network MTL architectures

(b) Topic-wise multi-task architecture

Figure 2: Neural Network MTL architectures. Compared with existing neural network MTL architectures (a), our proposed topic-wise multi-task architecture (b) expand the task-specific modules f_t to topic-task-specific modules g_{kt}, and a topic module h decides which topic a given sample belongs to and the corresponding topic-task-specific models g_{kt} to apply. This new architecture allows data from different topics to enjoy different task relationships.

effect of the additional topic dimension in the above topic-task-element constraint lies in its element-wise sparsity. Without topics, it is reduced to task-wise sparsity, often a too-strong assumption for task relationship.

Next, we consider another topic-task-sparsity constraint that more explicitly takes advantage of the topic dimension. It is called topic-task-exclusive constraint, defined as

$$\Omega^{ex}(\Theta^s) = \frac{1}{2}\sum_k(\sum_t ||\theta^s_{k,t}||_2)^2. \tag{4}$$

The entire topic-task-exclusive constraint Ω^{ex} is the square of a $l_{2,1,2}$ norm. The l_1 norm for the task dimension still enforces the entire $\theta^s_{k,t}$ parameters to zero for certain topics k and task t. The l_2 norm for the topic dimension however, tends to balance the deviation of topic-task-specific parameters $\theta_{k,t}$ from the global θ^0 to be similar. In other words, the competition is now across tasks within each topic rather than among topic-task pairs under topic-task-element constraint. This norm is first applied to sparsity constraint, to our best knowledge. The usage of similar exclusive sparsity constraint, the square of $l_{1,2}$ norm in [33] shows its effect to find sparse feature selection structure for each neuron. The topic-task-sparsity designs given two proposed constraints are visualized as the norms of the learned topic-task-sparse parameters in Fig. 3.

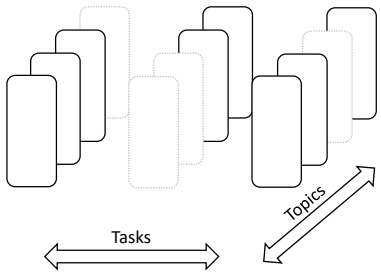


Figure 3: Topic-task-sparsity parameters under sparsity constraints. Gray dash blocks represent topic-task-sparsity parameters $\theta^s_{t,k}$ that are zero.

4.3 Topic-Wise Multi-Task Sparsity Model

The TOpic-wise Multi-tAsk sparsiTy mOdel (TOMATO) is the combination of the topic-wise multi-task architecture and either of the topic-task-element or topic-task-exclusive constraint. A typical model implementation, as used in the experiments of this work, is described as following from bottom to top, as shown in Fig. 2. First, the shared bottom module $\phi(x)$ can be any feature extraction modules (e.g., multi-layer perceptron, convolutional neural network or recurrent neural network), from which the multi-layer perceptron is used. More specifically,

$$\phi(x) = a(w^\phi_1 a(w^\phi_0 x + b^\phi_0) + b^\phi_1), \tag{5}$$

where w^ϕ_0 (w^ϕ_1) is the weight for first (second) hidden layers of the shared bottom module $\phi(x)$, similar for bias b^ϕ_0 (b^ϕ_1), and $a()$ is the activation function. The topic module upon that $h(\phi(x))$ is simple linear transformation with a softmax activation

$$h(\phi(x)) = softmax(w^h\phi(x) + b^h). \tag{6}$$

At the same time, the topic-task-specific function $g(k, \phi(x), t)$ for each topic k and each task t is typically set as multi-layer perceptron of m layers as

$$g(k, \phi(x), t) = a(w^g_{k,t,m-1}...a(w^g_{k,t,0}\phi(x)+b^g_{k,t,0})...+b^g_{k,t,m-1}), \tag{7}$$

where the sets of all weights and biases $\{w^g_{k,t,j-1}, b^g_{k,t,j-1}|j \in [m]\}$ is flattened and concatenated into the topic-task-specific parameter $\theta_{k,t}$. Finally, the topic-weighted combined task-specific linear layer $f(\phi(x), t) = \sum(h(\phi(x))_k g(k, \phi(x), t))$ optionally goes through a

final task-specific linear layer $\hat{y} = \psi(f(\phi(x), t), t)$. When this final task-specific layer is used, some task-wise difference can be conserved no matter the topic-task-sparsity.

The overall loss function $\mathcal{L}$ is defined as

$$\mathcal{L} = \sum_{t\in[T]} \frac{1}{N_t} \sum_{n_t\in[N_t]} L(\hat{y}, y_t) + \lambda\Omega(\Theta^s), \tag{8}$$

where Ω can either be Ω^{el} or Ω^{ex}, and λ controls the strength of the sparsity penalty. When $\lambda \to +\infty$, the topic-task-specific functions $g(x, \theta_{k,t})$ reduce to a global function $g(x, \theta^0)$, leading to the closest task relationship, and vice versa.

4.4 Optimization

Both topic-task-element and topic-task-exclusive constraints are non-smooth functions, which exclude the usage of the conventional stochastic gradient descent (SGD) method to minimize Eq. 8. Alternatively, therefore, we use the stochastic proximal gradient method. At each iteration j, it calculates an intermediate parameters $\bar{\Theta}^s$ using the conventional SGD step and optimizes the solution or proximal operator as

$$\Theta^{s,j+1} = \arg\min_{\Theta^s} \frac{1}{2\lambda r}||\Theta^s - \bar{\Theta}^s||_2^2 + \Omega(\Theta^s), \tag{9}$$

where r is the learning rate of the current iteration.

As for the topic-task-element constraint $\Omega^{el}(\Theta^s)$ in Eq. 3, the proximal operator, the proximal operator from Eq. 9 for each topic k and each task t can be calculated independently. Therefore, the proximal operator can be easily derived as

$$prox_{el}(\theta^s_{k,t}) = (1 - \frac{\lambda r}{||\bar{\theta}^s_{k,t}||_2})_+ \bar{\theta}^s_{k,t}, \tag{10}$$

where $()_+$ is the clip function $max(, 0)$.

The proximal operator for the topic-task-exclusive constraint $\Omega^{ex}(\Theta^s)$ is more complicated because the parameters for different tasks are coupled by the l_2 norm at the topic-dimension.

Lemma 1. *The solution for Eq. 9 with $\Omega = \Omega^{ex}$, is*

$$prox_{ex}(\theta^s_{k,t}) = (1 - \frac{A_k}{||\bar{\theta}^s_{k,t}||_2})_+ \bar{\theta}^s_{k,t}, \tag{11}$$

where A_k is maximum of the "diluted" average $A_{\mathcal{T}'}$ of $\{||\theta^s_{k,t}||_2 \,|t \in \mathcal{T}'\}$, $\forall \mathcal{T}' \subset [T]$,

$$\begin{aligned} A_k &= \max_{\mathcal{T}'\subset[T]} A_{k,\mathcal{T}'} \\ s.t., A_{k,\mathcal{T}'} &= \frac{1}{\frac{1}{\lambda r} + |\mathcal{T}'|} \sum_{t'\in\mathcal{T}'} ||\bar{\theta}^s_{k,t'}||_2. \end{aligned} \tag{12}$$

And we denote $\mathcal{T}_k = \arg\max_{\mathcal{T}'} A_{k,\mathcal{T}'}$.

The key of the proof is to notice that the maximum "diluted" average A_k is a threshold that divide the tasks into two sets, $\mathcal{T}_k$ and $[T]\backslash\mathcal{T}_k$, where $\mathcal{T}_k = \arg\max_{\mathcal{T}'} A_{k,\mathcal{T}'}$.

Lemma 2. *If $t \in \mathcal{T}_k$, then $||\bar{\theta}^s_{k,t}||_2 \geq A_k$ and if $t \in [T]\backslash\mathcal{T}_k$, $||\bar{\theta}^s_{k,t}||_2 \leq A_k$.*

Proof. First, assume otherwise $||\bar{\theta}^s_{k,t}||_2 < A_k$ for some $t \in \mathcal{T}_k$. It is equivalent as

$$\begin{aligned} &||\bar{\theta}^s_{k,t}||_2 < A_k \\ \Leftrightarrow\ &(\frac{1}{\lambda r} + |\mathcal{T}_k|)||\bar{\theta}^s_{k,t}||_2 < \sum_{t'\in\mathcal{T}_k} ||\bar{\theta}^s_{k,t'}||_2 \\ \Leftrightarrow\ &(\frac{1}{\lambda r} + |\mathcal{T}_k\backslash\{t\}|)||\bar{\theta}^s_{k,t}||_2 < \sum_{t'\in\mathcal{T}_k\backslash\{t\}} ||\bar{\theta}^s_{k,t'}||_2 \\ \Leftrightarrow\ &||\bar{\theta}^s_{k,t}||_2 < A_{\mathcal{T}_k\backslash\{t\}}. \end{aligned} \tag{13}$$

On the other hand,

$$\begin{aligned} &A_k - A_{\mathcal{T}_k\backslash\{t\}} \\ =&\delta[(\frac{1}{\lambda r} + |\mathcal{T}_k\backslash\{t\}|) \sum_{t'\in\mathcal{T}_k} ||\bar{\theta}^s_{k,t'}||_2 - (\frac{1}{\lambda r} + |\mathcal{T}_k|) \sum_{t'\in\mathcal{T}_k\backslash\{t\}} ||\bar{\theta}^s_{k,t'}||_2] \\ =&\delta(\frac{1}{\lambda r} + |\mathcal{T}_k\backslash\{t\}|)\ [\ ||\bar{\theta}^s_{k,t}||_2 - A_{\mathcal{T}_k\backslash\{t\}}] \\ <&0, \end{aligned} \tag{14}$$

where $\delta = \frac{1}{(\frac{1}{\lambda r}+|\mathcal{T}_k|)(\frac{1}{\lambda r}+|\mathcal{T}_k\backslash\{t\}|)} > 0$. It contradicts with the condition that A_k is the maximum "diluted" average. Second, for the second statement, because Eq. 13 and Eq. 14 only involve equivalence relationship, it is straightforward to prove with $\mathcal{T}\backslash\{t\}$ replaced by $\mathcal{T}\bigcup\{t\}$. □

With Lemma. 2, the proximal operator in Eq. 11 can be rewritten as

$$prox_{ex}(\theta^s_{k,t}) = \begin{cases} (1 - \frac{A_k}{||\bar{\theta}^s_{k,t}||_2})\bar{\theta}^s_{k,t} & t \in \mathcal{T}_k \\ 0 & t \in [T]\backslash\mathcal{T}_k \end{cases},$$

which can be easily proved being the sub-differential calculus solution of Eq. 9 , with $\Omega = \Omega^{ex}$. It therefore proves Lemma. 1.

Algorithm 1 Greedy Calculation A_k

Input: $||\bar{\theta}^s_{k,t}||_2$ for $t \in [T]$, λ, r
output: A_k
1: Sort $||\bar{\theta}^s_{k,t}||_2$, and denote $a_i = ||\bar{\theta}^s_{k,t_i}||_2$ for $i \in [T]$ s.t., $a_1 \geq a_2 \geq ... \geq a_T \geq 0$;
2: $S_0 \leftarrow 0$, $S_i \leftarrow S_{i-1} + a_i$ for $i = 1, 2, ..., T$;
3: $S_i \leftarrow \frac{1}{\frac{1}{\lambda r}+i} S_i$ for $i \in [T]$;
4: Return $A_k = \max_i S_i$.

The remaining challenge is to efficiently calculate A_k in Eq. 11. We prove that it can be obtained by a simple greedy algorithm, as Algorithm. 1.

The time complexity in terms of number of tasks T is $Tlog(T)$. The correctness of Algorithm. 1 is proved below.

Proof. Using Lemma. 2, we can prove that $\mathcal{T}_k$ must be the subset of the largest $|\mathcal{T}_k|$ norms.

From Lemma. 2, $\forall t \in \mathcal{T}_k$, $t' \in [T]\backslash\mathcal{T}_k$,

$$\begin{aligned} &||\bar{\theta}^s_{k,t}||_2 \geq A_k, \\ &||\bar{\theta}^s_{k,t'}||_2 \leq A_k \\ \Rightarrow &||\bar{\theta}^s_{k,t}||_2 \geq ||\bar{\theta}^s_{k,t'}||_2. \end{aligned} \tag{15}$$

Dataset	Synthetic	MNIST-MTL	AwA2	School	MCSEM
input dim d	64	64	500	28	756
output dim p	5	2	2	3	5
# tasks T	12	10	50	139	12
# samples/task N_t	2,000	1,800	100	111	3,523

Table 2: Data statistics summary

Therefore the greedy Algorithm. 1 will not miss A_k. □

The name "diluted" average is from the observation that for the given subset $\mathcal{T}' \subset [T]$, the $A_{k,\mathcal{T}'}$ from Eq. 12, can be treated as the average of the union of $\frac{1}{\lambda r}$ zeros and $\{||\bar{\theta}^s_{k,t'}||_2 \,|t' \in \mathcal{T}'\}$. When $\lambda \to \infty$, A_k reduces to the maximum of all subset averages, which is the largest norm $a_1 = \max_{t\in[T]} ||\bar{\theta}^s_{k,t}||_2$. Going back to Eq. 11, in this case, the resulting proximal operators are zero for all the tasks, as expected. On the other hand, when $\lambda < \infty$, $a_1 > A_k$. Back to Eq. 11, this means that with finite strength of the sparsity penalty λ, the topic-task-exclusive constraint Ω^{ex} will not zero out the parameters for all tasks within each topic.

5 EXPERIMENTS

In this section, we apply the proposed models against one synthetic and four real-world datasets to validate the proposal. All codes and datasets used in the experiments are available [1].

5.1 Datasets

We evaluate our proposal against one synthetic dataset and four real datasets.

(1) **Synthetic Dataset**: We generate a synthetic MTL dataset following the within-topic task relationship hypothesis.
- Input feature: K topic cores $e_k \in \mathbf{R}^d$ are sampled from normal distributions $\mathcal{N}(0, \sigma_e^2)$ for each $k \in [K]$. Input features are generated as $x_t \in \mathbf{R}^d$ from normal distributions $\mathcal{N}(0, \sigma_x^2)$. The topic distribution $h(x)$ of an input x is determined as $h(x)_k \propto \exp(||x - e_k||_2^2/\sigma^2_{topic})$.
- Parameters: a global linear weight $w_0 \in \mathbf{R}^{d\times p}$ is element-wise sampled from $\mathcal{N}(0, \sigma^2_{w_0})$. Topic-task-sparsity weight $w^s_{k,t}$ are generated in the way that within each topic, $z \ll T$ tasks $\mathcal{T}_k \subset [T]$ are randomly sampled and their topic-task-sparsity weight are assigned random values from $\mathcal{N}(0, \sigma^2_{w^s})$, while the rest are assigned 0. The topic-task-specific weight is $w_{k,t} = w_0 + w^s_{k,t}$.
- Label: The linear output $\bar{y}_t$ for input x_t is generated by $\bar{y}_t = softmax(\sum_{i=1}^d x_{t,i} \sum_{k=1}^K h(x_t)_k w_{k,t,i})$. We add non-linearity to the final label $y_t = \beta(\bar{y}_t)$, with β the non-linear function used in [20].

(2) **MNIST-MTL Dataset**: We use the multi-task version of the MNIST data (MNIST-MTL) [15]. Each task is a binary classification problem that distinguish one digit from the others. For each of the $T = 10$ tasks, we sample 900 positive samples and 900 negative samples with 100 samples for each of the other digits. We adopt the feature extraction method used in linear methods [12] to get input of dimension $d = 64$.

(3) **AwA2 Dataset**: AwA2 is a benchmark dataset containing 37,322 images of 50 animals [31]. Each task is a binary classification problem similar to MNIST-MTL data. For each of the $T = 50$ tasks, we sample 50 positive samples and 1 negative samples for each of the other animals. We use the pre-trained features [31] and reduce the dimension to 500 with PCA.

(4) **School Dataset**: School data is a benchmark dataset containing performance of 15362 students from 139 schools [2]. The score performance is partitioned to 3 segments, $[0, 10)$, $[10, 20)$ and $[20, 71)$. Each task is to classify the performance of students from a school. It is a challenging dataset due to relatively smaller relevance between features and labels as indicated by the regression performance reported in [2].

(5) **MCSEM Dataset**: The multi-channel social emotion mining (MCSEM) data is crawled from public posts from 12 news channels on Facebook, together with their public users' emotional reactions (i.e., clicks on the emoticon buttons, love, angry, wow, happy, and sad). We used the pre-trained BERT model [6] to obtain the document embeddings as the input with $d = 756$. The emotional reactions for each post are normalized to label distributions over the five emoticon labels. Each task is to predict the label distribution given the posts of each channel.

The data statistics is summarized in Table. 2. We use sample weighting to ensure that the sums of all sample weights for different tasks are the same, following $\mathcal{L}$ in Definition 1. The task-wise data imbalance problem is beyond the scope of this work. For each dataset, 20% samples are used for testing the remaining 80% as training. The results reported are averages from 10 iterations of random splits. The split uses stage-wise sampling with tasks as stages to avoid random imbalance across tasks.

5.2 Competing Models

We compare the proposed models with a list of baselines and state-of-the-art MTL neural network models.

(1) **Separate**: It learns each task with separate neural network modules that do not correlate. It is a baseline model to test the necessity of the use of MTL framework.

(2) **Shared-bottom**: It is a broadly used MTL model where all tasks share bottom feature extraction module and own their own top modules.

(3) **Single**: It learns all tasks with a single neural network module. It is a baseline model to test the necessity of the use of MTL framework.

(4) **Inter-task-l_2** [7]: Based on **Shared-bottom** model, the $l2$ penalty is assigned to the difference of task-specific module parameters of each pair of tasks. This model therefore assumes that the parameters of different tasks should be similar.

(5) **DMTRL** [32]: Based on **Shared-bottom** model, the tensor consisting of task-specific parameters of all tasks are assumed of a low-rank structure, modeled by tensor factorization. In our comparison, we adopt the Tucker decomposition as it shows the most reliable results in [32].

(6) **MRN** [18]: Based on **Shared-bottom** model, the tensor consisting of task-specific parameters of all tasks are assumed

[1] https://github.com/JasonLC506/MTSEM

with a fully-decomposed tensor normal distribution, whose parameters are jointly learnt with the model parameters. The task relationship is assumed as the shared prior distributions for corresponding parameters of modules for different tasks.

(7) **Cross-stitch** [22]: Based on **Shared-bottom** model, the task-specific modules are assumed able to communicate with each other by stitches connection between each pair of them. The task relationship is assumed as information sharing.

(8) **MMoE** [20]: This model consists of multiple expert modules and task-specific expert distribution modules to combine the output of experts for each task. For fair comparison, we also add shared-bottom feature extraction modules as the most bottom layers.

(9) **TOMATO-el**: It is the proposed topic-wise multi-task sparsity model with topic-task-element constraint Ω^{el} from Eq. 3.

(10) **TOMATO-ex**: It is the proposed topic-wise multi-task sparsity model with topic-task-exclusive constraint Ω^{ex} from Eq. 4.

Here in this work, we only focus on their capability to capture task relationship in MTL problems. Therefore, we implement a unified architecture for all models. They share the same shared-bottom module structure as MLP (multi-layer perceptron) of one layer except **Separate** and other modules of different models are MLP. All models are trained using stochastic gradient descent (SGD) with learning rate at iteration i, $r_i = r_0 \gamma^{i/\eta}$, where r_0 is the initial learning rate, γ is the decay rate and η is the decay steps. Random dropout for certain layers and l_2 regularization are used to avoid overfitting.

5.3 Hyperparameter Tuning

The list of tunable hyperparameters for different models and their choice ranges are provided as following.

- common hyperparameters
  - initial learning rate r_0: $\{0.001, 0.01, 0.1, 1.0\}$,
  - decay rate γ: $\{0.8, 0.9\}$,
  - decay steps η: $\{100, 1000\}$,
  - input dropout rate dr_0: $\{0.0, 0.1, 0.2\}$
  - bottom hidden dropout rate dr_1: $\{0.0, 0.1, 0.2\}$,
  - bottom hidden layer dimension hd_{bottom}: $\{32, 64\}$,
  - top-most hidden layer dimension hd_{top}: $\{4, 8, 16, 32, 64\}$,[2]
  - l_2 regularization for all parameters λ_0: $\{0.0, 0.01, 0.1, 1.0\}$;
- model-specific hypeparameters
  - **Inter-task-l_2**
    - inter-task parameter difference l_2 regularization: $\{0.0, 0.0001, 0.001, 0.01, 0.1, 1.0\}$,
  - **MRN**
    - multi-linear prior norm regularization: $\{0.0, 0.0001, 0.001, 0.01, 0.1, 1.0\}$,
    - prior update frequency: $\{30, 100\}$[3],
  - **MMoE**
    - number of experts: $\{1, 2, 4, 8\}$,
  - **TOMATO-el** and **TOMATO-ex**
    - number of topics: $\{1, 2, 4, 8, 16\}$,
    - sparsity penalty λ: $\{0.0, 0.00001, 0.0001, 0.001, 0.01, 0.1\}$.

We use 20% training data as validation set to find the best hyperparameters for each model on each dataset using grid search over the union of common and model-specific hyperparameters. In order to decrease the time cost of hyperparameter tuning and also minimize impact of feature extraction on model performance, for each dataset, we first find the best shared-bottom hyperparameters dr_0, dr_1 and hd_{bottom}, and fix them for all other models on that data.

For **TOMATO-el** and **TOMATO-ex**, the number of layers m of the topic-task-specific modules are chosen from $\{1, 2\}$. In order to make fair comparison with other models, the top task-specific linear layer is added when $m = 1$ while omitted when $m = 2$. When $m = 2$, the output dimensions of the topic-task-specific modules should be the dimension p of the label.

5.4 Evaluation Measure

As the evaluation measures, we use the popularly-used *misclassification rate* for the experiments on Synthetic, MNIST-MTL, AwA2 and School datasets, and the *cross-entropy* for the experiments on MCSEM dataset where label distribution rather than a single true label is given for each data sample.

5.5 Results

5.5.1 ***Q1: Are the proposed models able to capture the within-topic task relationship?*** We show the l_2 norm of topic-task-sparsity parameters learned from the synthetic data in Fig. 4. Compared to Fig. 4.(a), the ground truth parameters, both the TOMATO-el (c) and TOMATO-ex (d) models can exactly capture the sparsity structure. We also test the case without topic by TOMATO-el (Similar results can be obtained by TOMATO-ex) (b), which cannot find the similarity between the majority tasks and task 2 and 7 (task 0 and 9) in data from topic 0 (1), but only treat all of them different from other tasks. This shows the effect of the topic-wise multi-task architecture.

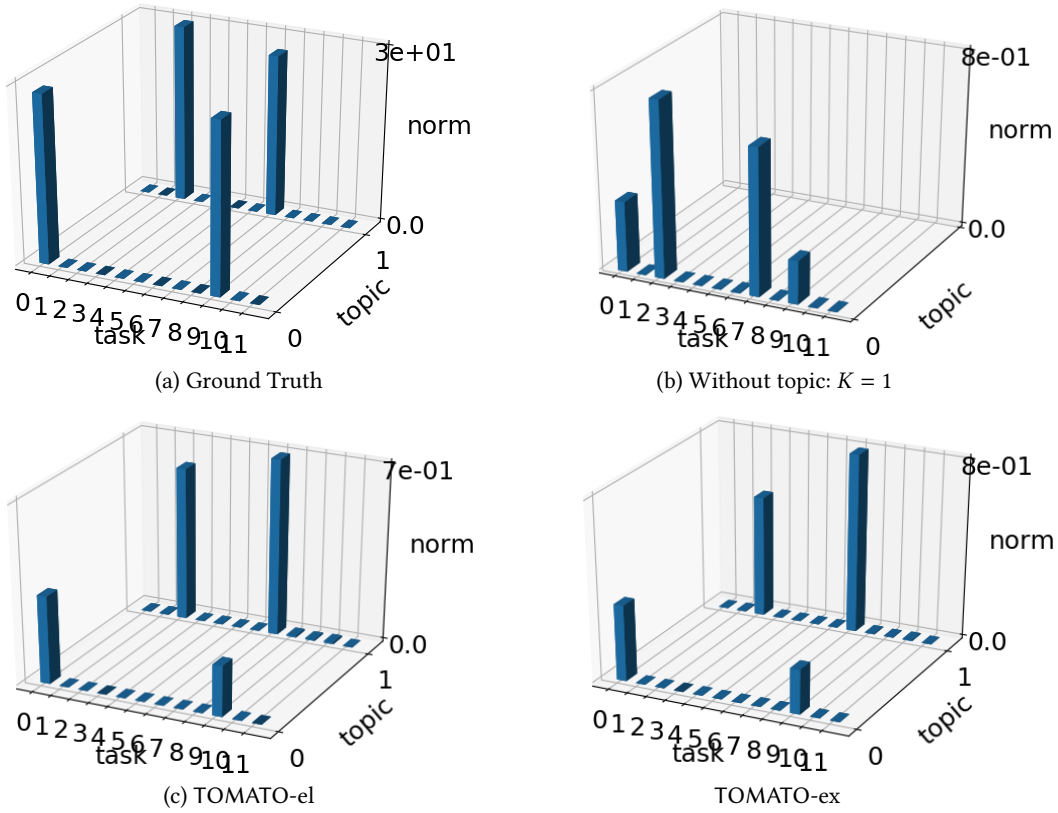
5.5.2 ***Q2: How do the proposed models perform?*** The overall performance results are presented in Table. 3. First, the proposed models **TOMATO-el** and **TOMATO-ex** consistently outperforms all the competing models. This validates the superiority of the proposed topic-wise multi-task architecture and also the proposed two topic-task-sparsity MTL designs. Second, the task relationship varies a lot across different datasets. On the one hand, comparing **Single** and **Separate**, which are the two extreme cases in MTL, their performance difference in different datasets varies. Therefore, some of the datasets (e.g., MNIST-MTL) have task relationship that is hard to catch, while others (e.g., synthetic) make it more beneficial to risk negative transfer for more data. On the other hand, the performance of different models, which are different assumptions of task-relationship, vary across different datasets. For example, **Cross-stitch** performs good on MCSEM data, but even worse than **Separate** baseline on synthetic data. This shows that the topic-wise multi-task architecture is more flexible in leveraging different task relationship.

[2]For **Shared-bottom, Inter-task-l_2, DMTRL** and **MRN**, it is the dimension of the hidden layer of the task-specific top module; for **TOMATO-el** and **TOMATO-ex**, it is the dimension of the hidden layer of the topic-task-specific top module; for **MMoE**, it is the dimension of the output layer of each expert module.

[3]100 is the default value used in [18], we add 30 for an alternative as different datasets are used

Dataset	synthetic (4800)	MNIST-MTL (3600)	AwA2 (1000)	School (3086)	MCSEM (8455)
Separate	16.55 (794)	2.59 (93)	7.49 (75)	50.95 (1,572)	1.371
Shared-bottom	14.51 (696)	2.68 (96)	4.92 (49)	50.01 (1,543)	1.326
Single	14.24 (684)	49.98 (1,799)	16.75 (168)	51.80 (1,599)	1.342
Inter-task-l_2	14.21 (682)	2.49 (90)	4.92 (49)	50.41 (1,556)	<u>1.322</u>
DMTRL	15.08 (724)	2.60 (94)	4.75 (48)	49.34 (1,523)	1.333
MRN	14.47 (695)	2.68 (96)	9.81 (98)	51.06 (1,576)	1.329
Cross-stitch	14.69 (705)	2.68 (96)	<u>4.63 (46)</u>	50.22 (1,550)	1.327
MMoE	14.40 (691)	2.59 (93)	13.60 (136)	48.18 (1,487)	1.337
TOMATO-el	**14.09 (676)**	<u>2.44 (88)</u>	**4.16 (42)**	**45.00 (1,389)**	<u>1.322</u>
TOMATO-ex	<u>14.10 (677)</u>	**2.32 (84)**	5.50 (55)	<u>46.72 (1,442)</u>	**1.321**

Table 3: Overall performance in misclassification rate in percentage or cross-entropy for MCSEM data. The numbers in parenthesis besides misclassification rate indicate the numbers of the misclassified samples and those besides dataset names are the size of the corresponding test set. Bold (underlined) are the best (second) for each data set.



(a) Ground Truth (b) Without topic: $K = 1$

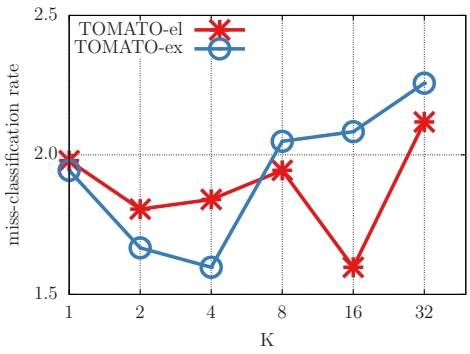
(c) TOMATO-el TOMATO-ex

Figure 4: The topic-wise multi-task structures as the l_2 norm of topic-task-sparsity parameters from the synthetic data: (a) ground truth as the weight $w^s_{k,t}$ used to generate the data; (b) without topic as learned by TOMATO-el with $K = 1$; (c) TOMATO-el, as learned by TOMATO-el with $K = 2$; (d) TOMATO-ex, as learned by TOMATO-ex with $K = 2$.

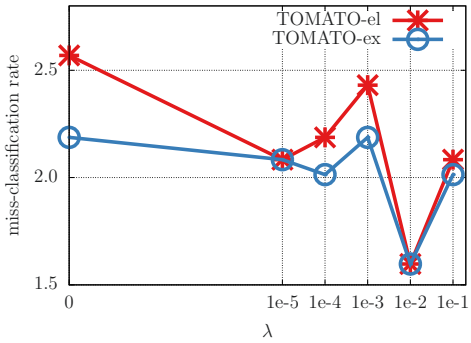
5.5.3 ***Q3: How is the trade-off between positive and negative transfer?*** The topic-wise multi-task architecture is proposed to capture more subtle task relationship so that achieve better trade-off between positive and negative transfer. Table. 3 gives an overall view of the answer, that the proposed architecture does perform better. Further more, here we take a detail view. In Table. 4, the average task-wise improvement is presented. For each model on each dataset, we calculate its performance on each task t as $perf_{model,t}$ of the dataset. After that, for each task, we calculate the relative improvement over the **Separate** model $100 * (perf_{\textbf{Separate},t} - perf_{model,t})/perf_{\textbf{Separate},t}$. This task-wise improvement provides

Dataset	synthetic	MNIST-MTL	AwA2	School	MCSEM
Separate	0.00	0.00	0.00	0.00	0.00
Shared-bottom	2.44	-0.09	3.05	3.18	3.08
Single	2.75	-48.66	-9.70	-2.04	2.20
Inter-task-l_2	2.79	0.10	3.05	1.96	3.39
DMTRL	1.75	-0.01	3.22	5.56	2.68
MRN	2.49	-0.08	-2.34	-0.05	3.00
Cross-stitch	2.23	-0.08	3.35	2.76	2.98
MMoE	2.57	0.00	-6.29	8.96	2.39
TOMATO-el	**2.94**	0.16	**3.88**	**16.74**	3.38
TOMATO-ex	2.92	**0.28**	2.41	12.93	**3.62**

Table 4: Average task-wise improvement percentage. Bold (underlined) are the best (second) for each data set.



(a) Performance over different number of topics K



(b) Performance over different sparsity constraint strength λ

Figure 5: Ablation Study in MNIST-MTL data

the judgement of positive or negative transfer. When this improvement is negative, for the specific task, there is no benefit to take into account other tasks, which is negative transfer, and vice versa. For each entry in Table. 4, we report the average task-wise improvement percentage over all tasks of a datasets from a model. We observe that the proposed models give consistent and better improvement over all datasets (i.e., from 5% to 46% better than competing models in different datasets). Therefore, it shows that the proposed methods do achieve better trade-off between positive and negative transfer.

5.5.4 ***Q4. Ablation study: Are the proposed architecture and MTL design nontrivial in real world data?*** We notice that there are two hyperparameters that distinguish the proposed models from the existing ones. Here we show ablation study results in MNIST-MTL data. Similar results can be obtained in other datasets. First when the number of topics $K = 1$, the proposed topic-wise multi-task architecture reduces to conventional MTL architecture. From the ablation study with different K values, Fig. 5(a), the performance of TOMATO-el (TOMATO-ex) with $K = 16$ ($K = 4$) is better than that of trivial model with $K = 1$. This validates the topic-wise multi-task architecture. Second when the sparsity penalty strength vanishes, $\lambda = 0$, the sparsity constraints are disabled. From the ablation study with different λ values, Fig. 5(b), the best performance is achieved with $\lambda = 1e - 2$. It validates the topic-task-sparsity MTL design.

Further, we show the l_2 norm of the learnt topic-task-sparsity parameter $\theta^s_{t,k}$ norm in Fig. 6 which manifests the design visualized in Fig. 3. Both TOMATO-el and TOMATO-ex show the proposed topic-wise multi-task sparsity structures, where within different topics, task relationship is different. For example, in Fig. 6(b), only topic 0 shows clear sparsity structure among tasks, which indicates strong task relationship among data from topic 0. Such data-dependent task relationship helps TOMATO explore task relationship within certain groups of data and avoid negative transfer effect from data without task relationship.

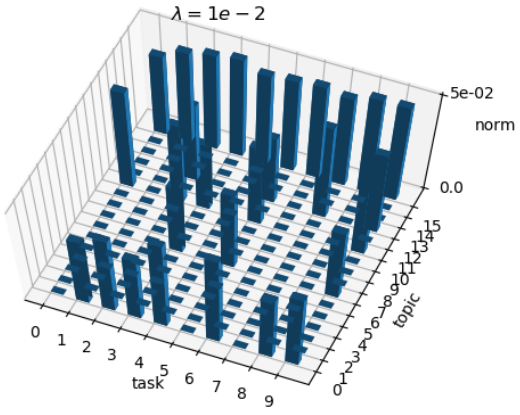
Moreover, the topic-task-element constraints result in topics with all topic-task-sparsity parameters zero-out, as shown in Fig. 6(a), while the topic-task-exclusive constraints will not, in Fig. 6(b), as expected from the analysis in Section. 4.4. More specifically, we notice there are more than one topics with all topic-task-sparsity parameters zero-out for TOMATO-el, as shown in Fig. 6(a). From the viewpoint of task relationship, those topics should be degenerated as the task relationship among them are the same (i.e., all tasks are the same). However, if we use smaller number of topics $K = 8$, which is sufficient for the optimal parameters state shown in Fig. 6(a) for $K = 16$, the performance gets worse, as shown in Fig. 5(a). It may be attributed to the effect of optimization process on the final local optimal parameter state.

6 CONCLUSION

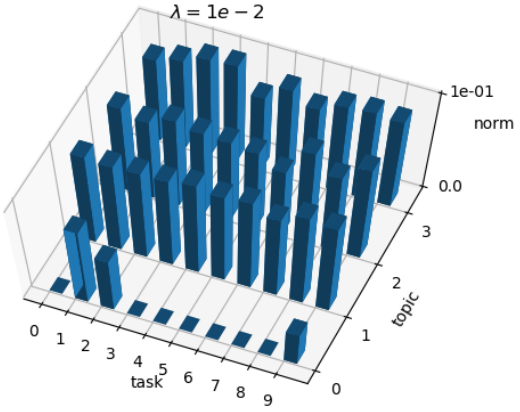
In this work, from a closer look into the validity of task relationship, we propose a within-topic task relationship hypothesis and develop a topic-wise multi-task architecture which is general enough to be combined with existing MTL designs. Further, we propose the topic-task-sparsity MTL design, specially designed for the topic-wise multi-task architecture, along with two types of sparsity constraints. The architecture, combined with the topic-task-sparsity design, constructs our proposed TOpic-wise Multi-tAsk sparsiTy mOdel (TOMATO). The experiments on both synthetic and real-world datasets show that the proposed models consistently outperform existing state-of-the-art models, which supports the validity of the within-topic-task relationship hypothesis.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. This work was in part



(a) TOMATO-el



(b) TOMATO-ex

Figure 6: Topic-task-sparsity parameter $\theta^s_{t,k}$ norms for each topic and task learnt in MNIST-MTL data

supported by NSF awards #1742702, #1820609, #1909702, #1915801, and #1934782.

REFERENCES

[1] Jose M Alvarez and Mathieu Salzmann. 2016. Learning the number of neurons in deep networks. In *NeurIPS*. 2270–2278.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2007. Multi-task feature learning. In *NeurIPS*. 41–48.

[3] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2008. Convex multi-task feature learning. *Machine Learning* 73, 3 (2008), 243–272.

[4] Maxwell D Collins and Pushmeet Kohli. 2014. Memory bounded deep convolutional networks. *arXiv preprint arXiv:1412.1442* (2014).

[5] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*. ACM, 160–167.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[7] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *ACL*, Vol. 2. 845–850.

[8] Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *SIGKDD*. ACM, 109–117.

[9] Lin Gong, Benjamin Haines, and Hongning Wang. 2017. Clustered model adaption for personalized sentiment analysis. In *WWW*. 937–946.

[10] Daniel Hernández-Lobato, José Miguel Hernández-Lobato, and Zoubin Ghahramani. 2015. A probabilistic model for dirty multi-task feature selection. In *ICML*. 1073–1082.

[11] Laurent Jacob, Jean-philippe Vert, and Francis R Bach. 2009. Clustered multi-task learning: A convex formulation. In *NeurIPS*. 745–752.

[12] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K. Ravikumar. 2010. A Dirty Model for Multi-task Learning. In *NeurIPS*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.). 964–972.

[13] Zhuoliang Kang, Kristen Grauman, and Fei Sha. 2011. Learning with Whom to Share in Multi-task Feature Learning.. In *ICML*, Vol. 2. 4.

[14] Abhishek Kumar and Hal Daume III. 2012. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417* (2012).

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[16] Giwoong Lee, Eunho Yang, and Sung Hwang. 2016. Asymmetric multi-task learning based on task relatedness and loss. In *ICML*. 230–238.

[17] Sulin Liu and Sinno Jialin Pan. 2017. Adaptive Group Sparse Multi-task Learning via Trace Lasso.. In *IJCAI*. 2358–2364.

[18] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and S Yu Philip. 2017. Learning multiple tasks with multilinear relationship networks. In *NeurIPS*. 1594–1603.

[19] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114* (2015).

[20] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *SIGKDD*. ACM, 1930–1939.

[21] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. 2016. The benefit of multitask representation learning. *JMLR* 17, 1 (2016), 2853–2884.

[22] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. 2016. Cross-stitch networks for multi-task learning. In *CVPR*. 3994–4003.

[23] Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. 2018. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *ICLR* (2018).

[24] Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

[25] Chi Su, Fan Yang, Shiliang Zhang, Qi Tian, Larry S Davis, and Wen Gao. 2015. Multi-task learning with low rank attribute embedding for person re-identification. In *ICCV*. 3739–3747.

[26] Xin Wang, Jinbo Bi, Shipeng Yu, and Jiangwen Sun. 2014. On multiplicative multitask feature learning. In *NeurIPS*. 2411–2419.

[27] Xiaogang Wang, Cha Zhang, and Zhengyou Zhang. 2009. Boosted multi-task learning for face verification with applications to web image and video search. In *CVPR*. IEEE, 142–149.

[28] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. In *NeurIPS*. 2074–2082.

[29] Fangzhao Wu and Yongfeng Huang. 2015. Collaborative multi-domain sentiment classification. In *ICDM*. IEEE, 459–468.

[30] Zhizheng Wu, Cassia Valentini-Botinhao, Oliver Watts, and Simon King. 2015. Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *ICASSP*. IEEE, 4460–4464.

[31] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. 2018. Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *TPAMI* (2018).

[32] Yongxin Yang and Timothy Hospedales. 2016. Deep multi-task representation learning: A tensor factorisation approach. *arXiv preprint arXiv:1605.06391* (2016).

[33] Jaehong Yoon and Sung Ju Hwang. 2017. Combined group and exclusive sparsity for deep neural networks. In *ICML*. JMLR. org, 3958–3966.

[34] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. 2012. Robust visual tracking via multi-task sparse learning. In *CVPR*. IEEE, 2042–2049.

[35] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).

[36] Yu Zhang and Dit-Yan Yeung. 2010. Multi-task warped gaussian process for personalized age estimation. In *CVPR*. IEEE, 2622–2629.

[37] Jiayu Zhou, Jianhui Chen, and Jieping Ye. 2011. Clustered multi-task learning via alternating structure optimization. In *NeurIPS*. 702–710.

[38] Qiang Zhou and Qi Zhao. 2015. Flexible clustered multi-task learning by learning representative tasks. *PAMI* 38, 2 (2015), 266–278.