SUBSPACE NETWORK: DEEP MULTI-TASK CENSORED REGRESSION FOR MODELING NEURODEGENERATIVE DISEASES

Anonymous authors
Paper under double-blind review

ABSTRACT

Over the past decade a wide spectrum of machine learning models have been developed to model the neurodegenerative diseases, associating biomarkers, especially non-intrusive neuroimaging markers, with key clinical scores measuring the cognitive status of patients. Multi-task learning (MTL) has been extensively explored in these studies to address challenges associated to high dimensionality and small cohort size. However, most existing MTL approaches are based on linear models and suffer from two major limitations: 1) they cannot explicitly consider upper/lower bounds in these clinical scores; 2) they lack the capability to capture complicated non-linear effects among the variables. In this paper, we propose the *Subspace Network*, an efficient deep modeling approach for non-linear multi-task censored regression. Each layer of the subspace network performs a multi-task censored regression to improve upon the predictions from the last layer via sketching a low-dimensional subspace to perform knowledge transfer among learning tasks. Under mild assumptions, for each layer the parametric subspace can be recovered using only one pass of training data. Empirical results demonstrate that the proposed subspace network quickly picks up correct parameter subspaces, and outperforms state-of-the-arts in predicting neurodegenerative clinical scores using information in brain imaging.

1 INTRODUCTION

Recent years have witnessed increasing interests on applying machine learning (ML) techniques to analyze biomedical data. Such data-driven approaches deliver promising performance improvements in many challenging predictive problems. For example, in the field of neurodegenerative diseases such as Alzheimer's disease and Parkinson's disease, researchers have exploited ML algorithms to predict the cognitive functionality of the patients from the brain imaging scans, e.g., using the magnetic resonance imaging (MRI) as in Adeli-Mosabbeb et al. (2015); Zhang et al. (2012); Zhou et al. (2011b). As a key finding, there are typically various types of prediction targets (e.g., cognitive scores), and they can be jointly learned using multi-task learning (MTL), e.g., Caruana (1998); Evgeniou & Pontil (2004); Zhang et al. (2012), where the predictive information is shared and transferred among related models to reinforce their generalization performance.

Two challenges persist despite the progress of applying MTL in disease modeling problems. First, it is important to notice that clinical targets, different from typically regression targets, are often naturally bounded. For example, the result from Mini-Mental State Examination (MMSE) test, a key reference for deciding cognitive impairments, ranges from 0 to 30 (a healthy subject): a smaller score indicates a higher level of cognitive dysfunction (please refer to Tombaugh & McIntyre (1992)). Other cognitive scores, such as Clinical Dementia Rating Scale (CDR) Hughes et al. (1982) and Alzheimer's Disease Assessment Scale-Cog (ADAS- Cog) Rosen et al. (1984), also have specific upper and lower bounds. Most existing approaches, e.g., Zhang et al. (2012); Zhou et al. (2011b); Poulin et al. (2011), relied on linear regression without considering the range constraint, partially due to the fact that mainstream MTL models for regression, e.g., Jalali et al. (2010); Argyriou et al. (2007); Zhang et al. (2012); Zhou et al. (2011a), are developed using the least squares loss and cannot be directly extended to censored regressions. As the second challenge, a majority of MTL research focused on linear models because of computational efficiency and theoretical guarantees. However,

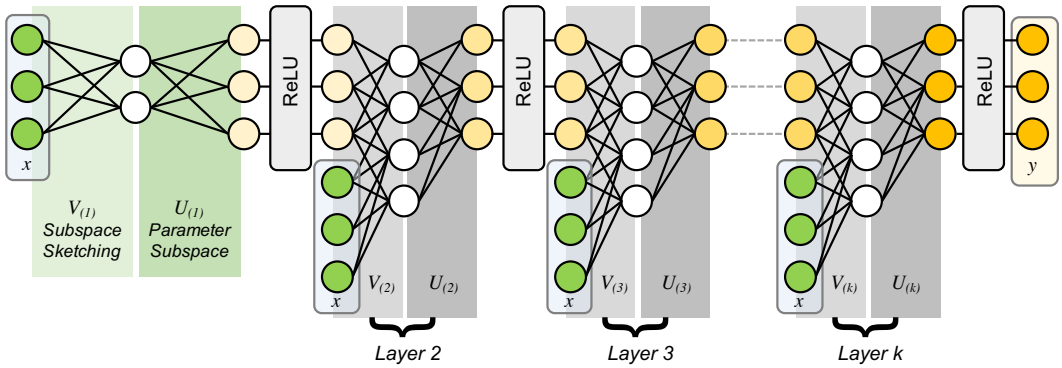


Figure 1: The proposed subspace network via hierarchical subspace sketching and refinement.

linear models cannot capture the complicated non-linear relationship between features and clinical targets. For example, Association et al. (2013) showed the early onset of Alzheimer's disease to be related to single-gene mutations on chromosomes *21*, *14*, and *1*, and the effects of such mutations on the cognitive impairment are hardly linear (please refer to Martins et al. (2005); Sweet et al. (2012)). Recent advances in multi-task deep neural networks Seltzer & Droppo (2013); Zhang et al. (2014); Wu et al. (2015) provide a promising direction, but their model complexity and high demands of training data prohibit their broader usages in clinical cohort studies.

To address the aforementioned challenges, we propose a *novel, solid, and efficient* deep modeling approach for non-linear multi-task censored regression, called *Subspace Network*, highlighting the following multi-fold technical innovations:

- It efficiently builds up a deep network in a **layer-by-layer feedforward** fashion, and in each layer considers a **censored** regression problem. The layer-wise training allows us to grow a deep model efficiently.
- It explores a **low-rank s**ubspace structure that captures task relatedness for better predictions. A critical difference on subspace decoupling between previous studies such as Mardani et al. (2015) Shen et al. (2016) and our method lies in our assumption of a low-rank structure on the **parameter space** among tasks rather than the original feature space.
- By leveraging the recent advances in online subspace sensing Mardani et al. (2015); Shen et al. (2016), we show that the parametric subspace can be recovered for each layer with feeding **only one pass** of the training data, allowing for more efficient layer-wise training.

Synthetic experiments verify the technical claims of the proposed subspace network, and it outperforms various state-of-the-arts methods in modeling neurodegenerative diseases on real datasets.

2 Multi-task censored regression via parameter subspace sketching and refinement

In censored regression, we are given a set of N observations $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ of D dimensional feature vectors $\{x_i \in \mathbb{R}^D\}$ and T corresponding outcomes $\{y_i \in \mathbb{R}_+^T\}$, where each outcome $y_{i,t} \in \mathbb{R}_+$, $t \in \{1, \cdots, T\}$, can be cognitive scores (e.g., MMSE and ADAS-Cog) or other biomarkers of interest such as proteomics[1]. For each outcome, the censored regression assumes a nonlinear relationship between the features and the outcome through a rectified linear unit (ReLU) transformation, i.e., $y_{i,t} = \mathrm{ReLU}\left(W_t^\top x_i + \epsilon\right)$ where $W_t \in \mathbb{R}^D$ is the coefficient for input features, ϵ is *i.i.d.* noise, and ReLU is defined by $\mathrm{ReLU}(z) = \max(z, 0)$. We can thus collectively represent the censored regression for multiple tasks by:

$$y_i = \mathrm{ReLU}\left(W x_i + \epsilon\right), \tag{1}$$

where $W = [W_1, \ldots, W_T]^\top \in \mathbb{R}^{T \times D}$ is the matrix including all the linear coefficients. We consider the regression problem for each outcome as a learning task, and one commonly used task relationship assumption is that the transformation matrix $W \in \mathbb{R}^{T \times D}$ belongs to a linear low-rank subspace

[1]Without loss of generality, in this paper we assume that outcomes are lower censored at 0. By using variants of Tobit models, e.g., as in Shen et al. (2016), the proposed algorithms and analysis can be extended to other censored models with minor changes in the loss function.

$\mathcal{U}$. The subspace allows us to represent W as product of two matrices, $W = UV$, where columns of $U \in \mathbb{R}^{T\times R} = [U_1, \ldots, U_T]^\top$ span the linear subspace $\mathcal{U}$, and $V \in \mathbb{R}^{R\times D}$ is the embedding coefficient. We note that the output y can be entry-wise decoupled, such that for each component $y_{i,t} = \mathrm{ReLU}(U_t^\top V x_i + \epsilon)$. By assuming a Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$, we derive the following likelihood function:

$$\Pr(y_{i,t}, x_i | U_t, V) = \phi\left(\frac{y_{i,t} - U_t^\top V x_i}{\sigma}\right) \mathbb{I}(y_{i,t} \in (0,\infty)) + \left[1 - Q\left(\frac{0 - U_t^\top V x_i}{\sigma}\right)\right] \mathbb{I}(y_{i,t} = 0),$$

where ϕ is the probabilistic density function of the standardized Gaussian $N(0,1)$ and Q is the standard Gaussian tail. The likelihood of (x_i, y_i) pair is thus given by:

$$\Pr(y_i, x_i | U, V) = \prod_{t=1}^{T} \left\{\phi\left(\frac{y_{i,t} - U_t^\top V x_i}{\sigma}\right) \mathbb{I}(y_{i,t} \in (0,\infty)) + \left[1 - Q\left(-\frac{U_t^\top V x_i}{\sigma}\right)\right] \mathbb{I}(y_{i,t} = 0)\right\}.$$

The likelihood function allows us to estimate subspace U and coefficient V from data $\mathcal{D}$. To enforce a low-rank subspace, one common approach is to impose a trace norm on UV, where trace norm of a matrix A is defined by $\|A\|_* = \sum_j s_j(A)$ and $s_j(A)$ is the jth singular value of A. Since $\|UV\|_* = \min_{U,V} \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2)$, e.g., see Srebro et al. (2005); Mardani et al. (2015), the objective function of multi-task censored regression problem is given by:

$$\min_{U,V} -\sum_{i=1}^{N} \log \Pr(y_i, x_i | U, V) + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2). \tag{2}$$

2.1 Online algorithm

We propose to solve the objective in (2) via the block coordinate descent approach which is reduced to iteratively updating the following two subproblems:

$$V^+ = \arg\min_V -\sum_{i=1}^{N} \log \Pr(y_i, x_i | U^-, V) + \frac{\lambda}{2}\|V\|_F^2 \tag{P:V}$$

$$U^+ = \arg\min_U -\sum_{i=1}^{N} \log \Pr(y_i, x_i | U, V^+) + \frac{\lambda}{2}\|U\|_F^2. \tag{P:U}$$

Define the instantaneous cost of the i-th datum:

$$g(x_i, y_i, U, V) = -\log \Pr(x_i, y_i | U, V) + \frac{\lambda}{2}\|U\|_F^2 + \frac{\lambda}{2}\|V\|_F^2, \tag{3}$$

and the online optimization form of (2) can be recast as an empirical cost minimization given below:

$$\min_{U,V} \frac{1}{N} \sum_{i=1}^{N} g(x_i, y_i, U, V).$$

According to the analysis in Section 2.2, one pass of the training data can warrant the subspace learning problem. We outline the solver for each subproblem as follows:

Problem (P:V) **sketches parameters in the current space.** We solve (P:V) using gradient descent. The parameter sketching couples all the subspace dimensions in V (not decoupled as in Shen et al. (2016)), and thus we need to solve this collectively. The update of V (V^+) can be obtained by solving the online problem given below:

$$\min_V g(x_i, y_i; U^-, V) \equiv -\sum_{t=1}^{T} \log \Pr(y_{i,t}, x | U_t^-, V) + \frac{\lambda}{2}\|V\|_F^2$$
$$= -\sum_{t=1}^{T} \log\left[\phi\left(\frac{y_{i,t} - \left(U_t^-\right)^\top V x}{\sigma}\right) \mathbb{I}(y_{i,t} \in (0,\infty)) + \left[1 - Q\left(\frac{-\left(U_t^-\right)^\top V x}{\sigma}\right)\right] \mathbb{I}(y_{i,t} = 0)\right] + \frac{\lambda}{2}\|V\|_F^2.$$

V^+ can be computed by the following gradient update:

$$V^+ = V^- - \eta \nabla_V g(x_i, y_i; U^-, V^+),$$

Algorithm 1 Single-layer parameter subspace sketching and refinement.

Require: Training data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, rank parameters λ and R,
Ensure: parameter subspace U, parameter sketch V
  Initialize U^- at random
  for $i = 1, \ldots, N$ **do**
    // 1. Sketching parameters in the current subspace

$$V^+ = \arg\min_V -\log \Pr(y_i, x_i | U^-, V) + \frac{\lambda}{2}\|V\|_F^2$$

    // 2. Parallel subspace refinement $\{U_t\}_{t=1}^T$
    for $t = 1, \ldots, T$ **do**
      $U_t^+ = \arg\min_{U_t} -\log \Pr(y_{i,t}, x_i | U_t, V^+) + \frac{\lambda}{2}\|U_t\|_2^2$
    end for
    Set $U^- = U^+, V^- = V^+$
  end for

where the gradient is given by:

$$\nabla_V g(x_i, y_i; U^-, V^+) = \lambda V + \sum_{t=1}^T \begin{cases} -\frac{y_{i,t} - \left(U_t^-\right)^\top V x_i}{\sigma^2} U_t^- x_i^\top & y_{i,t} \in (0, \infty) \\ \frac{\phi(z_t)}{\sigma[1 - Q(z_{i,t})]} U_t^- x_i^T & y_{i,t} = 0 \end{cases}$$

where $z_{i,t} = \sigma^{-1}(-\left(U_t^-\right)^\top V x)$. The Algorithm 3 for solving (P:V) is summarized in the Appendix.

Problem (P:U) **refines the subspace** U^+ **based on sketching.** We solve (P:U) using stochastic gradient descent (SGD). We note that the problem is decoupled for different subspace dimensions $t = 1, \ldots, T$ (i.e., rows of U). With careful parallel design, this procedure can be done very efficiently. Given a training data point (x_i, y_i), the problem related to the t-th subspace basis is given by:

$$\min_{U_t} g_t(x_i, y_{i,t}; U_t, V^+) \equiv -\log \Pr(y_{i,t}, x_i | U_t, V^+) + \frac{\lambda}{2}\|U_t\|_2^2$$
$$= -\log\left[\phi\left(\frac{y_{i,t} - U_t^\top V^+ x_i}{\sigma}\right)\mathbb{I}(y_{i,t} \in (0, \infty)) + \left[1 - Q\left(\frac{-U_t^\top V^+ x_i}{\sigma}\right)\right]\mathbb{I}(y_{i,t} = 0)\right] + \frac{\lambda}{2}\|U_t\|_2^2.$$

We can revise subspace by the following gradient update:

$$U_t^+ = U_t^- - \mu_t \nabla_{U_t} g_t(x_i, y_{i,t}; U_t, V^+),$$

where the gradient is given by:

$$\nabla_{U_t} g_t(x_i, y_{i,t}; U_{i,t}, V^+) = \lambda U_t + \begin{cases} -\frac{y_{i,t} - U_t^\top V^+ x}{\sigma^2} V^+ x_i & y_{i,t} \in (0, \infty) \\ \frac{\phi(z_{i,t})}{\sigma[1 - Q(z_{i,t})]} V^+ x_i & y_{i,t} = 0 \end{cases}$$

where $z_{i,t} = \sigma^{-1}(-U_t^\top V^+ x_i)$. We summarize the procedure in Algorithm 1 and show in Section 2.2 that under mild assumptions this procedure will be able to capture the underlying subspace structure in the parameter space with just one pass of the data.

2.2 Theoretical results

We establish both asymptotic and non-asymptotic convergence properties for Algorithm 1. The proof scheme is inspired by a series of previous works: Mairal et al. (2010); Kasiviswanathan et al. (2012); Shalev-Shwartz et al. (2012); Mardani et al. (2013; 2015); Shen et al. (2016). We briefly present the proof sketch, and refer readers to the literature for more details. At each iteration $i = 1, 2, ..., N$, we sample (x_i, y_i), and let U^i, V^i denote the intermediate U and V, to be differentiated from U_t, V_t which are the t-th columns of U, V. For the proof feasibility, we assume that $\{(x_i, y_i)\}_{i=1}^N$ are sampled i.i.d., and the subspace sequence $\{U^i\}_{i=1}^N$ lies in a compact set.

Asymptotic Case: To estimate U, the Stochastic Gradient Descent (SGD) iterations can be seen as minimizing the approximate cost $\frac{1}{N}\sum_{i=1}^N g'(x_i, y_i, U, V)$, where g' is a tight quadratic surrogate for

Algorithm 2 Network expansion via hierarchical parameter subspace sketching and refinement.

Require: Training data $\mathcal{D} = \{(x_i, y_i)\}$, target network depth K.
Ensure: The deep subspace network f
Set $f_{[0]}(x) = y$ and solve $f_{[0]}$ using Algorithm 1.
for $k = 1, \ldots, K-1$ **do**
// 1. Subspace sketching based on the current subspace using Algorithm 1:

$$U^*_{[k]}, V^*_{[k]} = \underset{U_{[k]}, V_{[k]}}{\arg\min}\, \mathbb{E}_{(x,y)\sim\mathcal{D}} \left\{ \ell(y, \mathrm{ReLU}\left(U_{[k]} V_{[k]} f_{[k-1]}(x)\right)) \right\},$$

// 2. Expand the layer using the refined subspace as our new network:

$$f_{[k]}(x) = \mathrm{ReLU}\left(U^*_{[k]} V^*_{[k]} f_{[k-1]}(x)\right)$$

end for
return $f = f_{[K]}$

g based on the second-order Taylor approximation around U^{N-1}. Furthermore, g can be shown to be smooth, by bounding its first-order and second-order gradients w.r.t. each U_t (similar to Appendix 1 of Shen et al. (2016)).

Following Mairal et al. (2010); Mardani et al. (2015), it can then be established that, as $N \to \infty$, *the subspace sequence* $\{U^i\}_{i=1}^N$ *asymptotically converges to a stationary-point of the batch estimator, under a few mild conditions*. We can sequentially show: 1) $\sum_{i=1}^N g'(x_i, y_i, U^i, V^i)$ asymptotically converges to $\sum_{i=1}^N g(x_i, y_i, U^i, V^i)$, according to the quasi-martingale property in the almost sure sense, owing to the tightness of g'; 2) the first point implies convergence of the associated gradient sequence, due to the regularity of g; 3) $g_t(x_i, y_i, U, V)$ is bi-convex for the block variables U_t and V.

Non-Asymptotic Case: When N is finite, Mardani et al. (2013) asserts that *the distance between successive subspace estimates will vanish as fast as* $o(1/i)$*:* $\|U^i - U^{i-1}\|_F \leq \frac{B}{i}$*, for some constant* B *that is independent of* i *and* N. Following Shen et al. (2016) to exploit the unsupervised formulation of regret analysis as in Kasiviswanathan et al. (2012); Shalev-Shwartz et al. (2012), we can similarly obtain a tight regret bound that will again vanish if $N \to \infty$.

3 SUBSPACE NETWORK VIA HIERARCHICAL SKETCHING AND REFINEMENT

The single layer model in (1) has very limited power to capture the highly nonlinear regression relationships, as the parameters are linearly linked to the subspace except for a ReLU operation. However, the single-layer procedure in Algorithm 1 has provided a building block, based on which we can develop an efficient algorithm to train a deep *subspace network* in a greedy fashion. We thus propose a network expansion procedure to overcome such limitation.

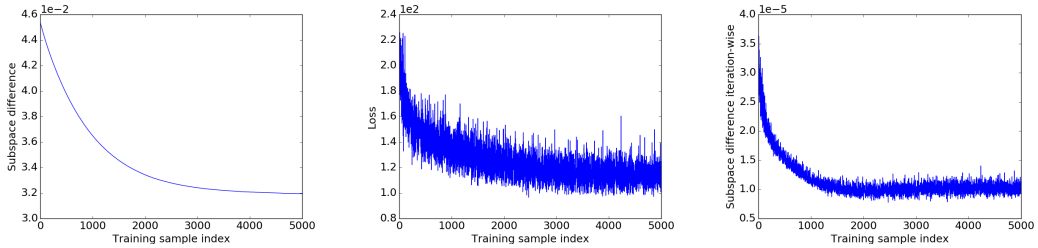
After we obtain the parameter subspace U and sketch V for the single-layer case (1), we project the data points by $\bar{x} = \mathrm{ReLU}(UVx)$. A straightforward idea of the expansion is to use $(\bar{x}, y)$ as the new samples to train another layer. Let $f_{[k-1]}$ denote the network structure we obtained before the k-th expansion starts, $k = 1, 2, ..., K-1$, the expansion can recursively stack more ReLU layers:

$$f_{[k]}(x) = \mathrm{ReLU}\left(U_{[k]} V_{[k]} f_{[k-1]}(x) + \epsilon\right), \tag{4}$$

However, we observe that simply stacking layers by repeating (4) many times can cause substantial information loss and degrade the generalization performance, especially since our training is layer-by-layer without "looking back" (i.e., top-down joint tuning). Inspired by deep residual networks by He et al. (2016) that exploit "skip connections" to pass lower-level data and features to higher levels, we concatenate the original samples with the newly transformed, censored outputs after each time of expansion, i.e., reformulating $\bar{x} = [\mathrm{ReLU}(UVx); x]$ (similar manners could be found in Zhou & Feng (2017)).The new form after the expansion is given below:

$$f_{[k]}(x) = \mathrm{ReLU}\left(U_{[k]} V_{[k]} [f_{[k-1]}(x); x] + \epsilon\right),$$

We summarize the network expansion process in Algorithm 2. The architecture of the resulting *subspace network* (SN) is illustrated in Figure 1. Compared to the single layer model (1), SN



(a) (b) (c)

Figure 2: (a) Subspace differences, w.r.t. the index i; (b) Convergence of Algorithm 1, w.r.t. the index i; (c) Iteration-wise subspace differences, w.r.t. the index i.

gradually refines the parameter subspaces by multiple stacked nonlinear projections. It is expected to achieve superior performance due to the higher learning capacity, and can also be viewed as a gradient boosting method. Meanwhile, the layer-wise low-rank subspace structural prior would improve generalization further compared to the naive multi-layer networks.

4 Experiment

4.1 Simulations on Synthetic Data

Subspace recovery in a single layer model. We first evaluate the recovery of the original subspace by using the proposed Algorithm 1 on synthetic data. We generated $X \in \mathbb{R}^{N\times d}$, $U \in \mathbb{R}^{T\times r}$ and $V \in \mathbb{R}^{r\times d}$, all as i.i.d. random Gaussian matrices. The target matrix $Y \in \mathbb{R}^{N\times T}$ was then synthesized using (1). We set $N = 5,000$, $d = 200$, $T = 100$ $r = 10$, and ϵ as a $N(0, 3^2)$ random noise.

Figure 2a displays the plot of *subspace difference* between the ground-truth U and the learned subspace U^i throughout the iterations, i.e., $\|U - U^i\|/\|U\|$ w.r.t. i. This result verifies that Algorithm 1 is able to correctly find and smoothly converge to the underlying low-rank subspace of the synthetic data. The objective values throughout the online training process of Algorithm 1 are plotted in Figure 2b. We further show the plot of *iteration-wise subspace differences*, defined as $\|U_i - U_{i-1}\|_F/\|U\|$, in Figure 2c, which complies with the $o(1/t)$ result in our non-asymptotic analysis.

On the other hand, after one-pass of the entire data, we would also like to verify whether the model weights can be reliably recovered by multiplying the subspace and projection matrix. We randomly pick two tasks and plot recovered weights versus ground truth. Correlation between recovered weights and true weights for all 100 tasks are also computed. As can be checked from Figure 4 in the Appendix, we display the distribution of computed correlations and scatter plot for two chosen tasks, and see that by passing data only once, our algorithm is able to recover the weights highly accurately (the majority of the predicted weights value having correlations with ground truth of above 0.9).

Multi-layer subspace network We re-generated synthetic data by repeatedly applying (1) for three times, each time followed the same setting as the single-layer model. A three-layer SN was then learned using Algorithm 2. As one simple baseline, a *multi-layer perceptron* (MLP) is trained, whose three hidden layers have the same dimensions as the three ReLU layers of the SN. Inspired by Xue et al. (2013); Sainath et al. (2013); Wang et al. (2015), we then applied low-rank matrix factorization to each layer of MLP, with the same desired rank r, creating the *factorized MLP* (f-MLP) baseline that has the identical architecture (including both ReLU hidden layers and linear bottleneck layers) to SN. We further re-trained f-MLP on the same data from end to end, leading to another *retrained factorized MLP* (rf-MLP) baseline. We adopt the same definition of subspace difference as in the single-layer case for evaluation. Table 1 compares the U subspace differences of the learned weight matrices, for each layer of SN, f-MLP, and rf-MLP, where SN achieves mostly the lowest differences.

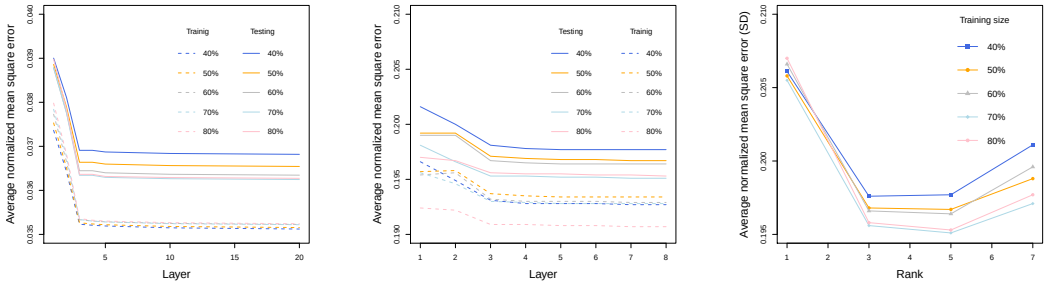
Table 1: Comparison of subspace differences for each layer of SN, f-MLP, and rf-MLP.

SN	Subspace Difference	f-MLP	Subspace Difference	rf-MLP	Subspace Difference
Layer 1	**0.0322**	Layer 1	0.0327	Layer 1	0.0324
Layer 2	0.0322	Layer 2	0.0319	Layer 2	**0.0318**
Layer 3	**0.0320**	Layer 3	0.0320	Layer 3	0.0323

Benefits of Going Deep. We re-generated synthetic data again in the same way as the first experiment; yet different from the first one, we now aim to show that a deep SN will boost performance over

Table 2: Average normalized mean square error (SD) under different approaches for synthetic data. Standard deviation of 10 trials is given in parenthesis. (SN Layer = 20)

SplitPercent	least squares	LS+ℓ_2	LS+ℓ_1	Multi-trace	Multi-$\ell_2 1$	Censor	SN
40%	.6416 (.0135)	.6416 (.0135)	.1850 (.0211)	.1289 (.0148)	.1351 (.0194)	.0428 (.0003)	**.0369** (.0002)
50%	.6407 (.0098)	.6407 (.0098)	.1795 (.0168)	.1350 (.0196)	.1407 (.0441)	.0408 (.0004)	**.0366** (.0003)
60%	.6397 (.0076)	.6397 (.0076)	.1754 (.0127)	.1659 (.0146)	.1914 (.0815)	.0395 (.0003)	**.0364** (.0003)
70%	.6416 (.0044)	.6416 (.0044)	.1766 (.0076)	.1534 (.0096)	.1515 (.0185)	.0388 (.0004)	**.0363** (.0003)
80%	.6401 (.0030)	.6401 (.0030)	.1725 (.0054)	.1653 (.0404)	.1934 (.0818)	.0383 (.0006)	**.0364** (.0005)



(a) (b) (c)

Figure 3: (a) Average normalized mean square error per layer for synthetic data; (b) Average normalized mean square error per layer for real data (with a fixed rank of 5); (c) Average normalized mean square error under different rank estimations for real data.

single-layer subspace recovery, even in this simplest synthetic setting. We will also compare SN with state-of-art approaches single task models and multi-task models. Least square is treated as a naive baseline, while ridge (LS + ℓ_2) and lasso (LS + ℓ_1) regressions are considered for shrinkage or variables selection purpose; Censor regression (Censor), also known as Tobit model, predicts bounded targets with higher accuracy, e.g., Berberidis et al. (2016). Multi-task models with regularizations on trace norm (Multi-trace) and $\ell_{2,1}$ norm (Multi-$\ell_{2,1}$) has been demonstrated to be successful on simultaneous structured/sparse learning, e.g., Yang et al. (2010); Zhang et al. (2013).[2].

We performed 10-fold random-sampling validation on the same dataset, i.e., randomly splitting into training and validation data 10 times. For each split, we fit model on training data and evaluated performance on validation data. Mean of normalized mean square error (MNMSE) across all tasks is obtained as the overall performance for each split. For methods without hyper parameters (least square and censor regression), an average of MNMSE (AMNMSE) for 10 splits is regarded as the final performance; for methods with tunable parameters, e.g., λ in lasso, we perform a grid search on λ values and choose the optimal one. We consider different splitting size with training samples containing [40%, 50%, 60%, 70%, 80%] of all the samples.

Table 3: Running time and platforms for different methods.

Method	Time (h)	Platform
Least Square	0.3	Matlab
LS+ℓ_2	1.6	Matlab
LS+ℓ_1	2.5	Matlab
Multi-trace	18	Matlab
Multi-$\ell_2 1$	17	Matlab
Censor	32	Matlab
SN (20 layers)	2.3	Python

We grow the number of layers in SN from 2 to 20, and plot the errors in Figure 3a. SN will steadily improves its performance with more layers, until reaching a plateau (here at around 5 layers, because of the simple underlying data distribution). The observation is quite consistent among different splits, and also in more experiments. Table 2 further compares SN (with 20 layers), the performance for all approaches, from which we can confirm that: (1) the censored model significantly reduces the bias compared to uncensored models; (2) the multi-task model captures task relatedness and outperforms single task models; (3) by combining the best of both worlds, SN outperforms all competitors.

Computation speed All experiments are run on the same machine (1 x Six-core Intel Xeon E5-1650 v3 [3.50GHz], 12 logic core, 128 GB RAM), without GPU acceleration. The running time for completing a 10-fold validation for all splitting percentages for synthetic data are given in Table3. SN improves generalization performance without significant loss of training speed, which is another advantage over other competitors.

[2]Least square, ridge, lasso, and censor regression are implemented by Matlab optimization toolbox. MTLs are implemented using default settings in MALSAR Zhou et al. (2011a) with parameters carefully tuned.

4.2 EXPERIMENTS ON REAL DATA

We evaluated SN in a real clinical setting with the goal to build models to predict important clinical scores representing the subject's cognitive status and signaling the progression of Alzheimer's disease (AD), from structural Magnetic Resonance Imaging (sMRI) data The ADNI phase 1 cohort[3] is used. In the experiments, we worked with 1.5 Tesla structural MRI collected from the subjects at baseline, and performed cortical reconstruction and volumetric segmentations with the FreeSurfer following the procotol in Jack et al. (2008). For each MRI image, we extracted 138 features representing the cortical thickness and surface areas of region-of-interests (ROIs) using the Desikan-Killiany cortical atlas Desikan et al. (2006). After preprocessing, we obtained a dataset containing 670 samples and 138 feature dimensions. These imaging features were used to predict a set of 30 clinical scores including ADAS scores Rosen et al. (1984) at baseline and future (6 months from baseline), baseline Logical Memory from Wechsler Memory Scale IV Scale—Fourth (2009), Neurobattery scores (i.e. immediate recall total score and Rey Auditory Verbal Learning Test scores), and the Neuropsychiatric Inventory (NPI) Cummings (1997) at baseline and future. A 6-layer SN with rank 5 per layer is employed by default.

Table 4: Average normalized mean square error (SD) for non-calibrated vs. calibrated SN for real data (6 layers). The standard deviation of 10 trials is given in parenthesis.

SplitPercent	Non-calibrate	Calibrate
40%	0.1993 (0.0034)	**0.1977** (0.0031)
50%	0.1987 (0.0043)	**0.1967** (0.0036)
60%	0.1991 (0.0044)	**0.1964** (0.0039)
70%	0.1982 (0.0042)	**0.1951** (0.0038)
80%	0.1984 (0.0041)	**0.1954** (0.0039)

σ^2 **Calibration** In synthetic formulation we assume that noise variance σ^2 is the same across all tasks, which is unnecessarily true in real cases. To deal with heterogeneous σ^2 among tasks, we add a *calibration* step in our optimization process, where we estimate task-specific $\hat{\sigma_t^2}$ using $\|y - \hat{y}\|_2^2/N$ before ReLu transformation, as the input for next layer and repeatomh layer-wise. We compare performance of both non-calibrated and calibrated methods in next subsection.

Performance We adopt the same metrics as the synthetic experiments to evaluate performance on real data. Different from synthetic data where the low-rank structure is predefined, for real data, we try different tuning parameters as well as different low-rank structures. Table 4 compares the performances between σ^2 non-calibrated versus calibrated models. We observe a clear improvement by assuming different σ^2 across tasks. Table 5 shows the results for all comparison methods, with SN outperforming all else. Figure 3b shows the SN performance growth with increasing the number of layers. Figure 3c further reveals the SN performance using varying rank estimations, when the groundtruth rank is unavailable in real data. As expected, the U-shape curve suggests that an overly low rank may not be informative enough to recover the original weight space, while a high rank structure cannot enforce as strong a structural prior.

Table 5: Average normalized mean square error (SD) under different approaches for real data. Standard deviation of 10 trials is given in parenthesis.

SplitPercent	least squares	Censor	LS+ℓ_2	LS+ℓ_1	Multi-trace	Multi-$\ell_2$1	SN
40%	.3874 (.0203)	.3870 (.0306)	.2632 (.0036)	.2393 (.0056)	.2572 (.0156)	.2006 (.0099)	**.1977** (.0031)
50%	.3119 (.0124)	.3072 (.0144)	.2444 (.0048)	.2202 (.0049)	.2406 (.0175)	.2003 (.0132)	**.1967** (.0035)
60%	.2779 (.0123)	.2719 (.0114)	.2338 (.005)	.2112 (.0055)	.2596 (.0233)	.2072 (.0204)	**.1964** (.0038)
70%	.2563 (.0108)	.2516 (.0108)	.2234 (.0058)	.2037 (.0042)	.2368 (.0362)	.2017 (.0116)	**.1951** (.0038)
80%	.2422 (.0112)	.2384 (.0099)	.2177 (.0062)	.2005 (.0054)	.2176 (.0171)	.2009 (.0050)	**.1953** (.0039)

5 CONCLUSIONS AND FUTURE WORK

In this paper we proposed a *Subspace Network*, an efficient deep modeling approach for non-linear multi-task censored regression, where each layer of the subspace network performs a multi-task censored regression to improve upon the predictions from the last layer via sketching a low-dimensional subspace to perform knowledge transfer among learning tasks. We show that under mild assumptions, for each layer we can recover the parametric subspace using only one pass of training data. We demonstrate empirically that the subspace network can quickly capture correct parameter subspaces, and outperforms state-of-the-arts in predicting neurodegenerative clinical scores from brain imaging. Based on similar formulations, the proposed method can be easily extended to cases where the targets have nonzero bounds, or both lower and upper bounds.

[3] `http://adni.loni.usc.edu/`

REFERENCES

Ehsan Adeli-Mosabbeb, Kim-Han Thung, Le An, Feng Shi, and Dinggang Shen. Robust feature-sample linear discriminant analysis for brain disorders diagnosis. In *NIPS*, pp. 658–666, 2015.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. *NIPS*, 19:41, 2007.

Alzheimer's Association et al. 2013 alzheimer's disease facts and figures. *Alzheimer's & dementia*, 9 (2):208–245, 2013.

Dimitris Berberidis, Vassilis Kekatos, and Georgios B Giannakis. Online censoring for large-scale regressions with application to streaming big data. *TSP*, 64(15):3854–3867, 2016.

Rich Caruana. Multitask learning. In *Learning to learn*, pp. 95–133. Springer, 1998.

Jeffrey L Cummings. The neuropsychiatric inventory assessing psychopathology in dementia patients. *Neurology*, 48(5 Suppl 6):10S–16S, 1997.

Rahul S Desikan, Florent Ségonne, Bruce Fischl, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3): 968–980, 2006.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Charles P Hughes, Leonard Berg, Warren L Danziger, et al. A new clinical scale for the staging of dementia. *The British journal of psychiatry*, 140(6):566–572, 1982.

Clifford R Jack, Matt A Bernstein, Nick C Fox, Paul Thompson, et al. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *J. of mag. res. imag.*, 27(4):685–691, 2008.

Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K Ravikumar. A dirty model for multi-task learning. In *NIPS*, pp. 964–972, 2010.

Shiva P Kasiviswanathan, Huahua Wang, Arindam Banerjee, and Prem Melville. Online l1-dictionary learning with application to novel document detection. In *NIPS*, pp. 2258–2266, 2012.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11(Jan):19–60, 2010.

Morteza Mardani, Gonzalo Mateos, and Georgios B Giannakis. Dynamic anomalography: Tracking network anomalies via sparsity and low rank. *J. of Sel. To. in Sig. Proc.*, 7(1):50–66, 2013.

Morteza Mardani, Gonzalo Mateos, and Georgios B Giannakis. Subspace learning and imputation for streaming big data matrices and tensors. *TSP*, 63(10):2663–2677, 2015.

CAR Martins, A Oulhaj, CA De Jager, and JH Williams. Apoe alleles predict the rate of cognitive decline in alzheimer disease a nonlinear model. *Neurology*, 65(12):1888–1893, 2005.

Stéphane P Poulin, Rebecca Dautoff, John C Morris, et al. Amygdala atrophy is prominent in early alzheimer's disease and relates to symptom severity. *Psy. Res.: Neur.*, 194(1):7–13, 2011.

Wilma G Rosen, Richard C Mohs, and Kenneth L Davis. A new rating scale for alzheimer's disease. *The American journal of psychiatry*, 1984.

Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, et al. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *ICASSP*, pp. 6655–6659. IEEE, 2013.

Wechsler D Wechsler Memory Scale—Fourth. Edition (wms-iv). *New York: Psychological Corporation*, 2009.

Michael L Seltzer and Jasha Droppo. Multi-task learning in deep neural networks for improved phoneme recognition. In *ICASSP*, pp. 6965–6969. IEEE, 2013.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Yanning Shen, Morteza Mardani, and Georgios B Giannakis. Online categorical subspace learning for sketching big data with misses. *arXiv preprint arXiv:1609.08235*, 2016.

Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *NIPS*, pp. 1329–1336, 2005.

Robert A Sweet, Howard Seltman, James E Emanuel, et al. Effect of alzheimer's disease risk genes on trajectories of cognitive function in the cardiovascular health study. *Ame. J. of Psyc.*, 169(9): 954–962, 2012.

Tom N Tombaugh and Nancy J McIntyre. The mini-mental state examination: a comprehensive review. *Journal of the American Geriatrics Society*, 40(9):922–935, 1992.

Zhangyang Wang, Jianchao Yang, Hailin Jin, et al. Deepfont: Identify your font from an image. In *MM*, pp. 451–459. ACM, 2015.

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

Jian Xue, Jinyu Li, and Yifan Gong. Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*, pp. 2365–2369, 2013.

Haiqin Yang, Irwin King, and Michael R Lyu. Online learning for multi-task feature selection. In *CIKM*, pp. 1693–1696. ACM, 2010.

Daoqiang Zhang, Dinggang Shen, Alzheimer's Disease Neuroimaging Initiative, et al. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease. *NeuroImage*, 59(2):895–907, 2012.

Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via structured multi-task sparse learning. *IJCV*, 101(2):367–383, 2013.

Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, pp. 94–108. Springer, 2014.

Jiayu Zhou, Jianhui Chen, and Jieping Ye. Malsar: Multi-task learning via structural regularization. *Arizona State University*, 21, 2011a.

Jiayu Zhou, Lei Yuan, Jun Liu, and Jieping Ye. A multi-task learning formulation for predicting disease progression. In *SIGKDD*, pp. 814–822. ACM, 2011b.

Zhi-Hua Zhou and Ji Feng. Deep forest: Towards an alternative to deep neural networks. *arXiv preprint arXiv:1702.08835*, 2017.

Appendix

Algorithm 3 Gradient descent algorithm for problem P:V.

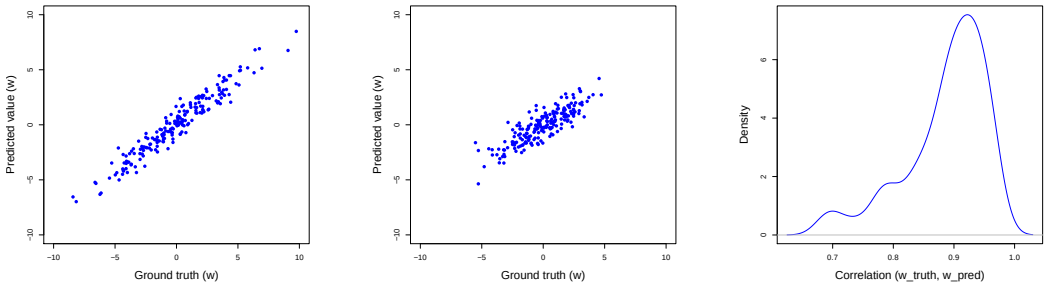
Require: Training data (x_i, y_i), U^-, step size η,
Ensure: sketch V
Initialize V^- at random
// 1. Perform gradient step and update the current solution of V.
for $t = 1, \ldots, T$ **do**
Compute $z_{i,t} = \sigma^{-1}(-\left(U_t^-\right)^\top V x_i)$
Compute the gradient for y_t:

$$\nabla g_t(x_i, y_{i,t}; U^-, V^+) = \begin{cases} -\frac{y_{i,t} - \left(U_t^-\right)^\top V x_i}{\sigma^2} U_t^- x_i^\top & y_{i,t} \in (0, \infty) \\ \frac{\phi(z_{i,t})}{\sigma[1 - Q(z_{i,t})]} U_t^- x_i^\top & y_{i,t} = 0 \end{cases}$$

end for
// 2. Update the current sketch V^-

$$V^+ = V^- - \eta \left[\sum_{t=1}^{T} \nabla g_t(x, y_t; U^-, V^+) + \lambda V \right]$$

Set $V^- = V^+$



(a) (b) (c)

Figure 4: (a) Predicted weight vs true weight for task 1; (b) Predicted weight vs true weight for task 2; (c) Distribution of correlation between predicted weight and true weight for all tasks