Learning (from) Deep Hierarchical Structure among Features

Yu Zhang,[1] **Lei Han**[2]
[1]HKUST, [2]Tencent AI Lab
yu.zhang.ust@gmail.com, leihan.cs@gmail.com

Abstract

Data features usually can be organized in a hierarchical structure to reflect the relations among them. Most of previous studies that utilize the hierarchical structure to help improve the performance of supervised learning tasks can only handle the structure of a limited height such as 2. In this paper, we propose a Deep Hierarchical Structure (DHS) method to handle the hierarchical structure of an arbitrary height with a convex objective function. The DHS method relies on the exponents of the edge weights in the hierarchical structure but the exponents need to be given by users or set to be identical by default, which may be suboptimal. Based on the DHS method, we propose a variant to learn the exponents from data. Moreover, we consider a case where even the hierarchical structure is not available. Based on the DHS method, we propose a Learning Deep Hierarchical Structure (LDHS) method which can learn the hierarchical structure via a generalized fused-Lasso regularizer and a proposed sequential constraint. All the optimization problems are solved by proximal methods where each subproblem has an efficient solution. Experiments on synthetic and real-world datasets show the effectiveness of the proposed methods.

Introduction

Most of previous studies to utilize the hierarchical structure among features, including the group Lasso (Yuan and Lin 2006) and the Hierarchical Penalization (HP) method (Szafranski, Grandvalet, and Morizet-Mahoudeaux 2007), can only handle the hierarchical structure of a limited height up to 2. In this paper, we aim to break this assumption by utilizing the available hierarchical structure with an arbitrary height to help learn an accurate model. Moreover, in some case where the hierarchical structure is unavailable, we aim to learn such hierarchical structure among features to improve the interpretability of the resultant learner.

Specifically, given a hierarchical structure to describe relations among features, we propose a Deep Hierarchical Structure (DHS) method to utilize it. In the DHS method, each model parameter corresponding to a data feature is penalized by the product of edge weights along the path from the root to the leaf node for that feature. Interestingly, when the exponents of the edge weights along the path from the root to each leaf node are summed to 1, the proposed objective function can be proved to be convex no matter what the height of the hierarchical structure is. Moreover, when all the exponents take the same value, we can show that the proposed objective function is equivalent to a problem with a hierarchical group lasso regularization term. In order to optimize the objective function of the DHS method, we adopt the FISTA algorithm (Beck and Teboulle 2009) each of whose subproblems has an efficiently analytical solution. Moreover, in the proposed DHS method, the exponents of the edge weights need to be set based on a priori information. When this information is not available, by default we just set them to be identical. Usually this strategy works but it may be suboptimal. In order to alleviate this problem, we propose a variant of the DHS method to learn the exponents from data.

Moreover, we consider a more general case where the hierarchical structure is not available. A hierarchical structure can give us more insight about the relations among features but learning it from data is a difficult problem. To the best of our knowledge, there is no work to directly learn the hierarchical structure among features. Here we give the first try based on the DHS method by proposing a Learning Deep Hierarchical Structure (LDHS) method. Given the height of the hierarchical structure, the LDHS method assumes that each path from the root to a leaf node corresponding to a data feature does not share any node between each other, then uses a generalized fused-Lasso regularizer to enforce nodes to fuse at each height, and finally designs a sequential constraint to make the learned structure form a hierarchical structure. For optimization, we use the GIST algorithm (Gong et al. 2013) to solve the objective function of the LDHS method. By comparing with several state-of-the-art baseline methods, experiments on several synthetic and real-world datasets show the effectiveness of the proposed models.

Related Work The Composite Absolute Penalties (CAP) method (Zhao, Rocha, and Yu 2009) learns from the hierarchical structure but via the group sparsity and its objective function is different from those of the proposed methods. Moreover, the proposed LDHS method can learn the hierarchical structure but the CAP method cannot. The tree-guided group Lasso proposed in (Kim and Xing 2010) can learn from the hierarchical structure for multi-output regression, whose settings are different from ours. There are some methods (Bondell and Reich 2007; Hallac, Leskovec, and Boyd 2015; Figueiredo and Nowak 2016) which can learn the group structure among features but fail to learn the hierarchical structure, while the proposed LDHS method can do that.

Notations A hierarchical structure is said to be balanced if the paths from the root to every leaf node have the same length. For unbalanced hierarchical structure, we can easily convert it to a balanced one by adding some internal nodes as shown in Figure 1. So in this paper, the hierarchical structure mentioned is always assumed to be balanced. For a hierarchical structure of height m, the root is at height 0, the children of the root are at height 1, and the leaf nodes are at height m. The number of nodes at height i is denoted by s_i and the nodes at height i are labeled by 1 to s_i from left to right. A node denoted by $\mathcal{N}_j^i$ means that it is the jth node at height i. The set of children of a node $\mathcal{N}_j^i$ is denoted by $\mathcal{C}_j^i$ and the number of children of a node $\mathcal{N}_j^i$ is denoted by $d_j^{(i)}$. For each leaf node $\mathcal{N}_i^m$, we define $d_i^{(m)} \equiv 1$ for the ease of the notations. We define $\mathcal{F}_j^i \equiv k$ as the index of the parent node of $\mathcal{N}_j^i$, implying that $\mathcal{N}_k^{i-1}$ is the parent node of $\mathcal{N}_j^i$. The edge from a node $\mathcal{N}_j^{i-1}$ to one of its children $\mathcal{N}_k^i$ is denoted by $\mathcal{E}_{j,k}^{(i)}$ and the weight of this edge is denoted by $\sigma_{j,k}^{(i)}$, where the superscript denotes the height in which the child node lies and the subscript denotes the indices of the parent and children nodes. The path from the root to the ith leaf node $\mathcal{N}_i^m$ is denoted by a sequence of $m+1$ integers as $\mathcal{P}_i = \{i_0, \dots, i_m\}$ where $i_0 = 1$, $i_m = i$, and node $\mathcal{N}_{i_j}^j$ is on the path for $j = 0, \dots, m$, and we define $\mathcal{P}_i^j \equiv i_j$ as the index of the node at height j on the path $\mathcal{P}_i$. In the bottom figure of Figure 1, we have $\mathcal{C}_1^0 = \{1,2,3\}$, $d_1^{(0)} = 3$, $\mathcal{C}_2^1 = \{3,4,5\}$, $d_2^{(1)} = 3$, $\mathcal{F}_2^1 = 1$, and $\mathcal{F}_3^2 = 2$. The path from the root to a leaf node $\mathcal{N}_5^2$ is $\mathcal{P}_5 = \{1,2,5\}$ where $\mathcal{P}_5^0 = 1$, $\mathcal{P}_5^1 = 2$, and $\mathcal{P}_5^2 = 5$.

Learning from Deep Hierarchical Structure

Most of the existing works such as the group Lasso and the HP method (Szafranski, Grandvalet, and Morizet-Mahoudeaux 2007) can only operate on a hierarchical structure of a limited height. However, in many applications, the hierarchical structure is much more complex. To improve the applicability, we present the proposed DHS method in this section.

The Objective Function

Suppose the training dataset is denoted by $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the ith data instance and y_i is its label, and the linear learning function is denoted by $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. Suppose that the features are organized in a balanced hierarchical structure of height m where $m \geq 2$. Based on a loss function $l(\cdot,\cdot,\cdot)$, the objective function of

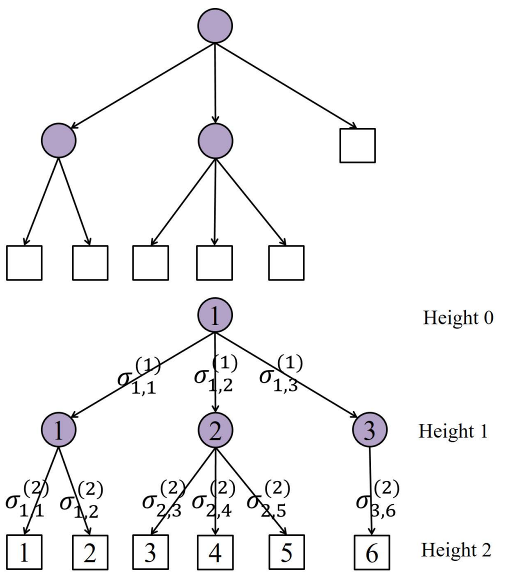


Figure 1: Illustration for the hierarchical structure. The top figure denotes a hierarchical structure and the bottom figure denotes the equivalently balanced structure.

the DHS method is formulated as

$$\min_{\mathbf{w},\boldsymbol{\sigma}} \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, y_i, \mathbf{w}) + \frac{\lambda_1}{2}\sum_{i=1}^{d} \frac{w_i^2}{\prod_{j=1}^{m}\left(\sigma^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}}\right)^{\theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}}}} + \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2$$

$$\text{s.t.} \sum_{j=1}^{s_i} d_j^{(i)} \sigma^{(i)}_{\mathcal{F}_j^i,j} = 1 \;\forall i \in [m],\; \sigma^{(i)}_{\mathcal{F}_j^i,j} \geq 0 \;\forall i,j, \qquad (1)$$

where w_i is the ith entry in $\mathbf{w}$, an edge weight $\sigma_{c,d}^{(j)}$ is defined in the previous section, $\|\cdot\|_2$ denotes the ℓ_2 norm of a vector, a/b for two scalars a and b is defined by continuation at zero as $a/0 = \infty$ if $a \neq 0$ and $0/0 = 0$, $[m]$ denotes an integer set from 1 to m, and $\theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}}$, an exponent, can be viewed as the importance for the edge weight $\sigma^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}}$. The summand in the second term of the objective function in problem (1) penalizes each w_i based on all the weights of edges on the path from the root node to the ith leaf node as well as the exponents. Hence two coefficients w_i and w_j will tend to have similar penalizations if they share many edges in the hierarchical structure. As we will see in Theorem 3, when exponents are identical to each other, this term is related to the group Lasso regularizer which enforces the group sparsity. The equality constraint in problem (1) is to restrict the scale of $\boldsymbol{\sigma}$. To preserve the convexity of problem (1) as we will see in the next section, it is required that $\theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}} \geq 0 \;\forall i,j$ and $\sum_{j=1}^{m} \theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}} = 1 \;\forall i$, which means that the sum of the nonnegative exponents of all the edges along a path from the root to each leaf node equals 1.

Properties

We first introduce a new family of convex functions with the proof in the supplementary material.

Theorem 1 $f(w, \mathbf{z}) = \frac{w^2}{\prod_{i=1}^m z_i^{\theta_i}}$ *is jointly convex with respect to* $w \in \mathbb{R}$ *and* $\mathbf{z} = (z_1, \ldots, z_m)^T \in \mathbb{R}^m$*, where* z_i*'s are required to be positive, given that* $\theta_i \geq 0$ *for* $i = 1, \ldots, m$ *and* $\sum_{i=1}^m \theta_i = 1$.

When $m = 1$, Theorem 1 asserts that $f(w, z) = w^2/z$ is jointly convex with respect to w and z when $z > 0$, which is a well-known result (pp. 72, (Boyd and Vandenberghe 2004)). When $m = 2$ and $\theta_1 = \theta_2 = 1/2$, Theorem 1 recovers Proposition 1 in (Szafranski, Grandvalet, and Morizet-Mahoudeaux 2007). Theorem 1 is more general since m can be any positive integer and different θ_i's can have different values.

Based on Theorem 1, we can prove the convexity of problem (1) in the following theorem.

Theorem 2 *Given that the loss function* $l(\mathbf{x}, y, \mathbf{w})$ *is convex with* $\mathbf{w}$*, problem (1) is jointly convex with respect to* $\mathbf{w}$ *and* $\boldsymbol{\sigma}$.

To see the effect of the regularizer in the second term of problem (1), we investigate two special cases, where m equals 2 or 3. When $m = 2$, problem (1) degenerates to the HP method (Szafranski, Grandvalet, and Morizet-Mahoudeaux 2007), which shows that when all the $\theta^{(j)}_{\mathcal{P}_i^{j-1}, \mathcal{P}_i^j}$'s equal $\frac{1}{2}$, problem (1) is equivalent to the $\ell_{\frac{4}{3},1}$-regularized group Lasso. When $m = 3$, we can derive an equivalent formulation of problem (1) as follows.

Theorem 3 *When* $m = 3$ *and all the* $\theta^{(j)}_{\mathcal{P}_i^{j-1}, \mathcal{P}_i^j}$*'s equal* $\frac{1}{3}$*, problem (1) is equivalent to this problem:*

$$\min_{\mathbf{w}} \frac{\lambda_1}{2} \left(\sum_{i=1}^{s_1} (d_i^{(1)})^{\frac{1}{6}} \left(\sum_{j \in \mathcal{C}_i^1} (d_j^{(2)})^{\frac{1}{5}} \left(\sum_{k \in \mathcal{C}_j^2} |w_k|^{\frac{3}{2}} \right)^{\frac{4}{5}} \right)^{\frac{5}{6}} \right)^2 + \frac{1}{n} \sum_{i=1}^n l(\mathbf{x}_i, y_i, \mathbf{w}) + \frac{\lambda_2}{2} \|\mathbf{w}\|_2^2. \quad (2)$$

According to Theorem 3, we can see the second term in the objective function of problem (1) can be converted to the first one of problem (2), which places the $\ell_{\frac{3}{2}}$ norm on model parameters corresponding to the leaf nodes which share the same parent node, then places the weighted $\ell_{\frac{6}{5}}$ norm on the internal nodes at height 2 which have the same parent node, and finally computes the squared weighted sum on the internal nodes at height 1. This regularizer can be viewed as a hierarchical group Lasso where at each height, the weights corresponding to nodes sharing the same parent node will be combined together via some norm. In general, for any positive integer m, when all the exponents of different $\sigma^i_{j,k}$ have the same value (i.e., $1/m$), we can always find the explicit form of the second term in the objective function of problem (1) in a similar way to the proof of Theorem 3.

Optimization

Since problem (1) is convex, we use the FISTA method (Beck and Teboulle 2009) to solve it. We use a variable $\boldsymbol{\phi}$ to denote the concatenation of $\mathbf{w}$ and $\boldsymbol{\sigma}$. We define

$$f(\boldsymbol{\phi}) = \sum_{i=1}^n \frac{l(\mathbf{x}_i, y_i, \mathbf{w})}{n} + \sum_{i=1}^d \frac{\lambda_1 w_i^2}{2 \prod_{j=1}^m \left(\sigma^{(j)}_{\mathcal{P}_i^{j-1}, \mathcal{P}_i^j} \right)^{\theta^{(j)}_{\mathcal{P}_i^{j-1}, \mathcal{P}_i^j}}}$$

and $g(\boldsymbol{\phi}) = \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2$. We define the set of constraints on $\boldsymbol{\phi}$ as

$$\mathcal{S}_\phi = \{\boldsymbol{\phi} | \sum_{j=1}^{s_i} d_j^{(i)} \sigma^{(i)}_{\mathcal{F}_j^i, j} = 1 \; \forall i \in [m], \; \sigma^{(i)}_{\mathcal{F}_j^i, j} \geq 0 \; \forall i, j\}.$$

In the FISTA algorithm, it does not minimize the original composite objective function $F(\boldsymbol{\phi}) = f(\boldsymbol{\phi}) + g(\boldsymbol{\phi})$, but instead solves a surrogate function:

$$q_r(\hat{\boldsymbol{\phi}}) = \arg \min_{\boldsymbol{\phi} \in \mathcal{S}_\phi} Q_r(\boldsymbol{\phi}, \hat{\boldsymbol{\phi}}),$$

where $Q_r(\boldsymbol{\phi}, \hat{\boldsymbol{\phi}}) = g(\boldsymbol{\phi}) + f(\hat{\boldsymbol{\phi}}) + (\boldsymbol{\phi} - \hat{\boldsymbol{\phi}})^T \bigtriangledown_{\boldsymbol{\phi}} f(\hat{\boldsymbol{\phi}}) + \frac{r}{2}\|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}\|_2^2$ and $\bigtriangledown_{\boldsymbol{\phi}} f(\hat{\boldsymbol{\phi}})$ denotes the derivative of $f(\boldsymbol{\phi})$ with respect to $\boldsymbol{\phi}$ at $\boldsymbol{\phi} = \hat{\boldsymbol{\phi}}$. Hence, in the FISTA algorithm, we just need to minimize $Q_r(\boldsymbol{\phi}, \hat{\boldsymbol{\phi}})$ with respect to $\boldsymbol{\phi} \in \mathcal{S}_\phi$. Specifically, we need to solve the following problem:

$$\min_{\mathbf{w}, \boldsymbol{\sigma}} \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2 + \frac{r}{2}\|\mathbf{w} - \tilde{\mathbf{w}}\|_2^2 + \frac{r}{2}\|\boldsymbol{\sigma} - \tilde{\boldsymbol{\sigma}}\|_2^2$$
$$\text{s.t.} \sum_{j=1}^{s_i} d_j^{(i)} \sigma^{(i)}_{\mathcal{F}_j^i, j} = 1 \; \forall i \in [m], \; \sigma^{(i)}_{\mathcal{F}_j^i, j} \geq 0 \; \forall i, j, \quad (3)$$

where r is a step size which can be determined by the FISTA algorithm, $\tilde{\mathbf{w}} = \hat{\mathbf{w}} - \frac{1}{r} \bigtriangledown_{\mathbf{w}} f(\hat{\mathbf{w}})$, and $\tilde{\boldsymbol{\sigma}} = \hat{\boldsymbol{\sigma}} - \frac{1}{r} \bigtriangledown_{\boldsymbol{\sigma}} f(\hat{\boldsymbol{\sigma}})$. It is easy to see that the solution for $\mathbf{w}$ in problem (3) is $\mathbf{w} = \frac{r}{\lambda_2 + r}\tilde{\mathbf{w}}$. For $\boldsymbol{\sigma}$ in problem (3), we can find that the corresponding problem can be decomposed into $m - 1$ subproblems with each one solving a problem with respect to $\{\sigma^{(i)}_{\mathcal{F}_j^i, j}\}_{j=1}^{s_i}$ and these subproblems have the same formulation as

$$\min_{\boldsymbol{\rho}} \|\boldsymbol{\rho} - \hat{\boldsymbol{\rho}}\|_2^2 \qquad \text{s.t. } \boldsymbol{\rho} \geq \mathbf{0}, \; \mathbf{a}^T \boldsymbol{\rho} = 1, \quad (4)$$

where $\mathbf{0}$ denotes a zero vector with appropriate size, $\geq$ denotes the elementwise inequalities between two vectors, and $\mathbf{a}$ is a constant vector with all entries positive. Problem (4) is a quadratic programming (QP) problem and we can use some off-the-shelf QP solver to solve it. To accelerate the optimization of problem (4), we propose a more efficient solution for problem (4) by solving its dual form and the detailed procedure is put in the supplementary material.

A Variant to Learn Exponents

In the DHS method, we need to manually set the exponents $\{\theta^{(j)}_{\mathcal{P}_i^{j-1}, \mathcal{P}_i^j}\}$. By default, we usually assume that all the $\theta^{(j)}_{\mathcal{P}_i^{j-1}, \mathcal{P}_i^j}$'s are equal to $\frac{1}{m}$, which satisfies the requirement to guarantee the convexity of problem (1). However, this setting may be suboptimal. In this section, we propose a DHS$_e$ method, a variant of the DHS method, to learn the exponents from data directly.

The objective function of the DHS$_e$ method is formulated as

$$
\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\sigma},\boldsymbol{\theta}} \ & \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, y_i, \mathbf{w}) + \frac{\lambda_1}{2}\sum_{i=1}^{d} \frac{w_i^2}{\prod_{j=1}^{m}\left(\sigma^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}}\right)^{\theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}}}} \\
& + \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2 \\
\text{s.t.} \ & \sum_{j=1}^{s_i} d_j^{(i)}\sigma^{(i)}_{\mathcal{F}_j^i,j} = 1\ \forall i \in [m],\ \sigma^{(i)}_{\mathcal{F}_j^i,j} \geq 0\ \forall i, j \\
& \sum_{j=1}^{m} \theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}} = 1\ \forall i \in [d],\ \theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}} \geq 0\ \forall i, j. \qquad (5)
\end{aligned}
$$

Different from problem (1) where all the $\theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}}$'s are constants, all the $\theta^{(j)}_{\mathcal{P}_i^{j-1},\mathcal{P}_i^{j}}$'s in problem (5) are variables to be optimized. The equality and inequality constraints with respect to $\boldsymbol{\theta}$ in problem (5) satisfy the requirements for the constant exponents in problem (1) and they form an $(m-1)-$dimensional simplex for each feature. Different from problem (1) which is convex, problem (5) can be proved to be non-convex with respect to all variables and hence we use the GIST algorithm (Gong et al. 2013) to solve it. Due to page limit, we put the detailed optimization procedure in the supplementary material.

Learning Deep Hierarchical Structure

In some applications, the hierarchical structure is not available. In this section, we propose the LDHS method to learn both the hierarchical structure and the model parameters from data directly.

We assume that the height of the hierarchical structure to be learned is given as m. Here we use slightly different notations to define the hierarchical structure. The weights of edges on the path from the root node to the ith leaf node corresponding to the ith feature are denoted by $\{\omega_i^{(1)}, \ldots, \omega_i^{(m)}\}$, where $\omega_i^{(j)}$ denotes the weight of an edge connecting the height $j-1$ and j on the path from the root node to the ith leaf node. At the beginning, we assume that the paths from the root node to any two different leaf nodes do not share any edge. When $\omega_j^{(i)}$ equals $\omega_k^{(i)}$ for some i, j and k, we can view it as a sign for that the two paths from the root node to the jth and kth leaf nodes become fused at height i and then in order to keep the whole structure as a hierarchical structure, it should be required that the subpaths on the two paths above height i are always fused, implying that $\omega_j^{(i')}$ will always equal $\omega_k^{(i')}$ when $i' \leq i$. So an algorithm that can learn a valid hierarchical structure should satisfy the following two requirements:

(1) It should have the ability to enforce $\omega_j^{(i)}$ to be equal to $\omega_k^{(i)}$ for some i, j and k;

(2) It should guarantee that when $\omega_j^{(i)}$ equals $\omega_k^{(i)}$, for all $i' < i$, $\omega_j^{(i')}$ will equal $\omega_k^{(i')}$.

Here we present the objective function of the LDHS method which can satisfy those two requirements:

$$
\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\omega}} \ & \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, y_i, \mathbf{w}) + \frac{\lambda_1}{2}\sum_{i=1}^{d} \frac{w_i^2}{\prod_{j=1}^{m}\left(\omega_i^{(j)}\right)^{\frac{1}{m}}} \\
& + \frac{\lambda_2}{2}\|\mathbf{w}\|_2^2 + \sum_{i=1}^{m}\eta_i \sum_{j<k} |\omega_k^{(i)} - \omega_j^{(i)}| \\
\text{s.t.} \ & \sum_{j=1}^{d} \omega_j^{(i)} = 1\ \forall i \in [m],\ \omega_j^{(i)} \geq 0\ \forall i, j \\
& |\omega_k^{(1)} - \omega_j^{(1)}| \leq \ldots \leq |\omega_k^{(m)} - \omega_j^{(m)}|\ \forall j, k, \qquad (6)
\end{aligned}
$$

where $\boldsymbol{\omega}$ is a vector containing all $\omega_i^{(j)}$'s. The last term in the objective function of problem (6), a layer-wise generalized fused-Lasso regularizer (Tibshirani et al. 2005; Hocking et al. 2011), can make $\omega_j^{(i)}$ equal to $\omega_k^{(i)}$ for some i, j and k, and so this regularizer can satisfy the first requirement. The sequential inequality constraint in problem (6) can satisfy the second requirement since when $\omega_j^{(i)}$ equals $\omega_k^{(i)}$, we can get $|\omega_j^{(i')} - \omega_k^{(i')}| \leq 0$ for any $1 \leq i' < i$, implying that $\omega_j^{(i')} = \omega_k^{(i')}$. Therefore, problem (6), which satisfies the two requirements, can learn a hierarchical structure. The exponents of all the $\omega_i^{(j)}$'s are set to be $\frac{1}{m}$. We can also set them to be other values or even learn them as the DHS$_e$ method did and this will be left as the future work. The regularization parameter η_i controls the level of fusion between $\{\omega_j^{(i)}\}_{j=1}^{d}$ at height i. A larger η_i will lead to more identical values in $\{\omega_j^{(i)}\}_{j=1}^{d}$. Therefore, it is intuitive to define an increasing order for η_i's from height m to height 0 to help construct the hierarchical structure. In practice, we set $\eta_{i-1} = \upsilon\eta_i$ for $i \geq 2$ with some constant $\upsilon > 1$. When the hierarchical structure is available or equivalently $\boldsymbol{\omega}$ is given, problem (6) becomes problem (1) and hence the LDHS method is a generalization of the DHS method to learn the hierarchical structure.

Even though the objective function of problem (6) is convex based on Theorem 1, the whole problem is non-convex due to the sequential inequality constraint and we also use the GIST method to solve it. We still use the variable $\boldsymbol{\phi}$ to denote the concatenation of $\mathbf{w}$ and $\boldsymbol{\omega}$. We define a set of constraints on $\boldsymbol{\phi}$ as $\mathcal{S}_\phi = \{\boldsymbol{\phi} | \sum_{j=1}^{d}\omega_j^{(i)} = 1\ \forall i \in [m],\ \omega_j^{(i)} \geq 0\ \forall i, j,\ |\omega_k^{(1)} - \omega_j^{(1)}| \leq \ldots \leq |\omega_k^{(m)} - \omega_j^{(m)}|\ \forall j, k.\}$. We define

$$
g(\boldsymbol{\phi}) = \begin{cases} \frac{\lambda_2\|\mathbf{w}\|_2^2}{2} + \sum_{i=1}^{m}\eta_i\sum_{j<k}|\omega_k^{(i)} - \omega_j^{(i)}| & \text{if } \boldsymbol{\phi} \in \mathcal{S}_\phi \\ +\infty, & \text{otherwise} \end{cases}
$$

$$
f(\boldsymbol{\phi}) = \frac{1}{n}\sum_{i=1}^{n} l(\mathbf{x}_i, y_i, \mathbf{w}) + \frac{\lambda_1}{2}\sum_{i=1}^{d}\frac{w_i^2}{\prod_{j=1}^{m}\left(\omega_i^{(j)}\right)^{\frac{1}{m}}}.
$$

By omitting some constant terms, the proximal problem to be solved in the GIST algorithm can be formulated as

$$\min_{\mathbf{w},\boldsymbol{\omega}} \frac{r}{2}\|\mathbf{w}-\tilde{\mathbf{w}}\|_2^2+\frac{r}{2}\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\|_2^2+\frac{\lambda_2}{2}\|\mathbf{w}\|_2^2$$
$$+\sum_{i=1}^m \eta_i \sum_{j<k} |\omega_k^{(i)}-\omega_j^{(i)}|$$
$$\text{s.t.} \sum_{j=1}^d \omega_j^{(i)}=1\ \forall i\in[m],\ \omega_j^{(i)}\geq 0\ \forall i,j$$
$$|\omega_k^{(1)}-\omega_j^{(1)}|\leq\ldots\leq|\omega_k^{(m)}-\omega_j^{(m)}|\ \forall j,k, \qquad (7)$$

where $\hat{\mathbf{w}}$ and $\hat{\boldsymbol{\omega}}$ are previous estimations for $\mathbf{w}$ and $\boldsymbol{\omega}$ respectively, $\tilde{\mathbf{w}}=\hat{\mathbf{w}}-\frac{1}{r}\bigtriangledown_{\mathbf{w}} f(\hat{\mathbf{w}})$, and $\tilde{\boldsymbol{\omega}}=\hat{\boldsymbol{\omega}}-\frac{1}{r}\bigtriangledown_{\boldsymbol{\omega}} f(\hat{\boldsymbol{\omega}})$. It is easy to see that the solution for $\mathbf{w}$ in problem (7) is $\mathbf{w}=\frac{r}{\lambda_2+r}\tilde{\mathbf{w}}$. For $\boldsymbol{\omega}$ in problem (7), its problem can be simplified as

$$\min_{\boldsymbol{\omega}} \frac{r}{2}\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\|_2^2+\sum_{i=1}^m \eta_i\|\mathbf{B}\boldsymbol{\omega}^{(i)}\|_1 \qquad (8)$$
$$\text{s.t. } \mathbf{1}^T\boldsymbol{\omega}^{(i)}=1,\ \boldsymbol{\omega}^{(i)}\geq\mathbf{0},\ |\mathbf{B}\boldsymbol{\omega}^{(1)}|\leq\ldots\leq|\mathbf{B}\boldsymbol{\omega}^{(m)}|,$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm of a vector, $\boldsymbol{\omega}^{(i)}=(\omega_1^{(i)},\ldots,\omega_d^{(i)})^T$, $\mathbf{1}$ is a vector of all ones with an appropriate size, $|\mathbf{a}|$ for a vector $\mathbf{a}$ returns a vector with each entry being the absolute value of the corresponding entry in $\mathbf{a}$, $\geq$ denotes the elementwise "no smaller than" relation between two vectors, and $\mathbf{B}$ is a $\frac{d(d-1)}{2}\times d$ matrix with each of its rows containing only two non-zero entries 1 and -1 at corresponding locations. Problem (8) seems complicated and we reformulate it as

$$\min_{\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\tau}} \frac{r}{2}\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\|_2^2+\sum_{i=1}^m \eta_i\|\boldsymbol{\tau}^{(i)}\|_1$$
$$\text{s.t. } \mathbf{1}^T\boldsymbol{\omega}^{(i)}=1, \boldsymbol{\omega}^{(i)}\geq\mathbf{0}, \boldsymbol{\mu}^{(i)}=\boldsymbol{\omega}^{(i)}, \boldsymbol{\tau}^{(i)}=\mathbf{B}\boldsymbol{\mu}^{(i)}$$
$$|\boldsymbol{\tau}^{(1)}|\leq\ldots\leq|\boldsymbol{\tau}^{(m)}|. \qquad (9)$$

Due to the existent of linear equality constraints, we use the ADMM to solve problem (9). We define the augmented Lagrangian function as

$$\mathcal{L}(\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\tau})$$
$$=\sum_{i=1}^m\left(\frac{\rho}{2}\|\boldsymbol{\tau}^{(i)}-\mathbf{B}\boldsymbol{\mu}^{(i)}\|_2^2+\mathbf{q}_i^T(\boldsymbol{\tau}^{(i)}-\mathbf{B}\boldsymbol{\mu}^{(i)})\right)$$
$$+\sum_{i=1}^m\left(\mathbf{p}_i^T(\boldsymbol{\mu}^{(i)}-\boldsymbol{\omega}^{(i)})+\frac{\rho}{2}\|\boldsymbol{\mu}^{(i)}-\boldsymbol{\omega}^{(i)}\|_2^2\right)$$
$$+\frac{r}{2}\|\boldsymbol{\omega}-\tilde{\boldsymbol{\omega}}\|_2^2+\sum_{i=1}^m\eta_i\|\boldsymbol{\tau}^{(i)}\|_1,$$

where $\{\mathbf{p}_i\}_{i=1}^m$ and $\{\mathbf{q}_i\}_{i=1}^m$ act as Lagrangian multipliers, and ρ is a penalty parameter. Then we need to solve the following problem as

$$\min_{\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\tau}} \mathcal{L}(\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\tau})$$
$$\text{s.t. } \mathbf{1}^T\boldsymbol{\omega}^{(i)}=1,\ \boldsymbol{\omega}^{(i)}\geq\mathbf{0}\ \forall i\in[m]$$
$$|\boldsymbol{\tau}^{(1)}|\leq\ldots\leq|\boldsymbol{\tau}^{(m)}|. \qquad (10)$$

In the ADMM algorithm, problem (10) can be solved alternatively with respect to $\boldsymbol{\omega}$, $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$.

With fixed $\boldsymbol{\mu}$ and $\boldsymbol{\tau}$, we need to solve the following subproblem with respect to $\boldsymbol{\omega}$ as:

$$\min_{\boldsymbol{\omega}} \sum_{i=1}^m \|\boldsymbol{\omega}^{(i)}-\mathbf{b}^{(i)}\|_2^2 \text{ s.t. } \mathbf{1}^T\boldsymbol{\omega}^{(i)}=1,\ \boldsymbol{\omega}^{(i)}\geq\mathbf{0}\ \forall i,$$

where $\mathbf{b}^{(i)}=\frac{1}{r+\rho}\left(r\tilde{\boldsymbol{\omega}}^{(i)}+\mathbf{p}_i+\rho\boldsymbol{\mu}^{(i)}\right)$. This problem can be decomposed into m subproblems where the ith subproblem with respect to $\boldsymbol{\omega}^{(i)}$ has the same formulation as problem (4), which has an efficient solution.

With fixed $\boldsymbol{\omega}$ and $\boldsymbol{\tau}$, the subproblem with respect to $\boldsymbol{\mu}$ is a QP problem. By setting the derivative of $\mathcal{L}(\boldsymbol{\omega},\boldsymbol{\mu},\boldsymbol{\tau})$ with respect to $\boldsymbol{\mu}^{(i)}$ to zero, we can obtain the analytical solution for $\boldsymbol{\mu}^{(i)}$ as

$$\boldsymbol{\mu}^{(i)}=\left(\mathbf{I}+\mathbf{B}^T\mathbf{B}\right)^{-1}\left(\mathbf{B}^T\boldsymbol{\tau}^{(i)}+\frac{1}{\rho}\mathbf{B}^T\mathbf{q}_i+\boldsymbol{\omega}^{(i)}-\frac{1}{\rho}\mathbf{p}_i\right).$$

Note that $(\mathbf{I}+\mathbf{B}^T\mathbf{B})^{-1}$ is a constant matrix. So it can be computed and stored before solving the whole problem, leading to a more efficient implementation.

With fixed $\boldsymbol{\omega}$ and $\boldsymbol{\mu}$, the subproblem with respect to $\boldsymbol{\tau}$ can be decomposed into $\frac{1}{2}d(d-1)$ subproblems with the jth one formulated as

$$\min_{\boldsymbol{\tau}^{(i)}} \quad \sum_{i=1}^m\left(\frac{\rho}{2}\left(\tau_j^{(i)}-\nu_j^{(i)}\right)^2+\eta_i|\tau_j^{(i)}|\right)$$
$$\text{s.t.} \quad |\tau_j^{(1)}|\leq\ldots\leq|\tau_j^{(m)}|, \qquad (11)$$

where $\boldsymbol{\nu}^{(i)}=\mathbf{B}\boldsymbol{\mu}^{(i)}-\frac{1}{\rho}\mathbf{q}_i$ and $\tau_j^{(i)}$, $\nu_j^{(i)}$ are the jth entries of $\boldsymbol{\tau}^{(i)}$ and $\boldsymbol{\nu}^{(i)}$, respectively. The optimal $\tau_j^{(i)}$ must have the same sign as $\nu_j^{(i)}$ since otherwise we can flip the sign of $\tau_j^{(i)}$ to achieve a lower value for the first term in the summand of the objective function in problem (11) while keeping other terms unchanged and also satisfying the constraints, which leads to a lower objective value. Then by defining new variables $\{\hat{\tau}_j^{(i)}\}$ as $\hat{\tau}_j^{(i)}=\mathrm{sgn}(\nu_j^{(i)})\tau_j^{(i)}$ where $\mathrm{sgn}(\cdot)$ gives the sign of a scalar, $\hat{\tau}_j^{(i)}$ is always nonnegative and based on problem (11), the problem with respect to $\{\hat{\tau}_j^{(i)}\}$ can be formulated as

$$\min_{\{\hat{\tau}_j^{(i)}\}} \sum_{i=1}^m\left(\hat{\tau}_j^{(i)}-\hat{\nu}_j^{(i)}\right)^2 \text{ s.t. } 0\leq\hat{\tau}_j^{(1)}\leq\ldots\leq\hat{\tau}_j^{(m)}, \qquad (12)$$

where $\hat{\nu}_j^{(i)}=|\nu_j^{(i)}|-\frac{1}{\rho}\eta_i$. This problem is similar to problem (17) in (Han and Zhang 2015b) except the requirement that all $\hat{\tau}_j^{(i)}$'s are nonnegative. We first use the algorithm with linear complexity in Section 3.2 of (Han and Zhang 2015b) to solve this problem and then make negative ones in $\{\hat{\tau}_j^{(i)}\}$ become zero to obtain the final solution.

Experiments

In this section, we conduct empirical evaluations on both synthetic and real-world problems.

We compare the proposed models (i.e., DHS, DHS$_e$ and LDHS) with state-of-the-art structured feature learning methods, including the Lasso, Group Lasso (GLasso) (Yuan and Lin 2006), the CAP family with ℓ_4 and ℓ_∞ penalties denoted by CAP$_{\ell_4}$ and CAP$_{\ell_\infty}$ (Zhao, Rocha, and Yu 2009), and the HP method (Szafranski, Grandvalet, and Morizet-Mahoudeaux 2007). Among these baseline methods, the Lasso method does not take any group or hierarchical structure into consideration and the GLasso and HP models require a hierarchical structure of height 2, while the CAP, DHS and DHS$_e$ methods need a hierarchical structure with an arbitrary height. In order to provide a fair comparison, in all the experiments we first generate a hierarchical structure on the features and then apply it to the GLasso, CAP, HP, DHS and DHS$_e$ methods, where the group structure required by the GLasso and HP methods is obtained from the two bottom-most layers of the hierarchical structure. For the LDHS method, we set its height to be that of the given hierarchical structure.

Experiments on Synthetic Data

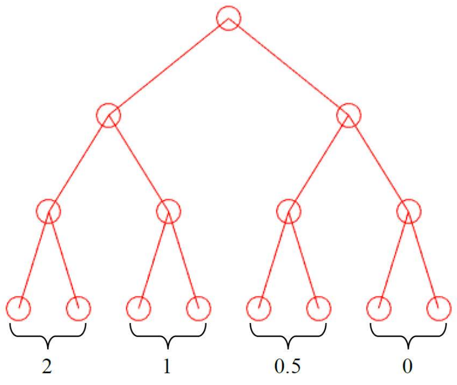
We first experiment on synthetic data. We generate three synthetic data by varying the height of the hierarchical structure as $m = 3$, 4 and 5, respectively. For simplicity, we use full binary trees and the numbers of features d under the three settings are equal to 2^3, 2^4 and 2^5, respectively. The ground truth of the feature weights $\mathbf{w}^*$ when $m = 3$ is shown in Fig. 2(a) and those for other cases are put in the supplementary material. Then, we generate data instances from a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, where $\mathbf{I}_d$ is a $d \times d$ identity matrix. We assume a linear model between data instances and labels, i.e., $\mathbf{y} = \mathbf{X}\mathbf{w}^* + \boldsymbol{\epsilon}$ with each entry ϵ_i in $\boldsymbol{\epsilon}$ $(i = 1, \cdots, n)$ following $\mathcal{N}(0, \xi^2)$. In all the settings, we generate $n = 100$ training samples and set $\xi = 2$.

Table 1: The MSE's in terms of 'mean±std' on synthetic data.

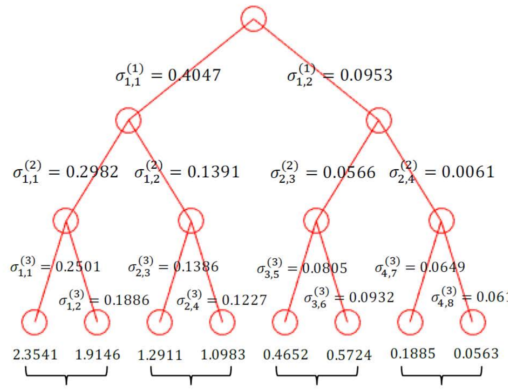
	$m = 3$	$m = 4$	$m = 5$
Lasso	0.2913±0.0399	0.8669±0.0499	1.8660±0.0549
GLasso	0.2945±0.0372	0.8597±0.0402	1.8877±0.0593
CAP$_{\ell_4}$	0.3069±0.0349	0.8597±0.0402	1.8885±0.0495
CAP$_{\ell_\infty}$	0.3069±0.0349	0.8597±0.0402	1.8886±0.0497
HP	0.2795±0.0439	0.8403±0.0314	1.5519±0.0550
DHS	0.2743±0.0337	**0.8275±0.0336**	1.4682± 0.0433
DHS$_e$	0.2748±0.0305	0.8337±0.0324	1.5596±0.0474
LDHS	**0.2370±0.0303**	0.8283±0.0354	**1.3894±0.0460**

We randomly generate 100 samples for testing and use another 100 random samples for validation to choose regularization parameters of all the methods. All of the regularization parameters in different models are chosen from a set $\{10^{-5}, 10^{-4} \cdots, 1\}$, except η_i's in the proposed LDHS method. As discussed before, we set $\eta_{i+1} = \eta_i/\upsilon$ for $i < m$, where we choose η_1 and υ from $\{10^{-5}, 10^{-4} \cdots, 1\}$ and $\{1.1, 2, 10\}$, respectively. We use the Mean Square Error (MSE) to evaluate different methods, where the MSE is defined as $\frac{1}{n}(\hat{\mathbf{w}} - \mathbf{w}^*)^\top \mathbf{X}^\top \mathbf{X}(\hat{\mathbf{w}} - \mathbf{w}^*)$ for an estimation $\hat{\mathbf{w}}$.

All the settings are repeated for 10 times to obtain the average results, which are reported in Table 1. From the results,

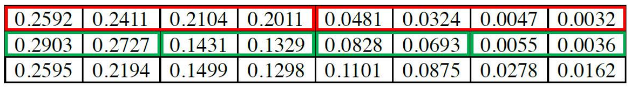


(a) The true hierarchical structure



$\sigma^{(1)}_{1,1} = 0.4047$
$\sigma^{(1)}_{1,2} = 0.0953$
$\sigma^{(2)}_{1,1} = 0.2982$
$\sigma^{(2)}_{1,2} = 0.1391$
$\sigma^{(2)}_{2,3} = 0.0566$
$\sigma^{(2)}_{2,4} = 0.0061$
$\sigma^{(3)}_{1,1} = 0.2501$
$\sigma^{(3)}_{1,2} = 0.1886$
$\sigma^{(3)}_{2,3} = 0.1386$
$\sigma^{(3)}_{2,4} = 0.1227$
$\sigma^{(3)}_{3,5} = 0.0805$
$\sigma^{(3)}_{3,6} = 0.0932$
$\sigma^{(3)}_{4,7} = 0.0649$
$\sigma^{(3)}_{4,8} = 0.0613$
2.3541
1.9146
1.2911
1.0983
0.4652
0.5724
0.1885
0.0563

(b) $\boldsymbol{\sigma}$ in DHS



0.2592	0.2411	0.2104	0.2011	0.0481	0.0324	0.0047	0.0032
0.2903	0.2727	0.1431	0.1329	0.0828	0.0693	0.0055	0.0036
0.2595	0.2194	0.1499	0.1298	0.1101	0.0875	0.0278	0.0162

(c) $\boldsymbol{\omega}$ in LDHS

Figure 2: The true hierarchical structure and estimated parameters when $m = 3$.

we observe that the proposed models with deeper hierarchical structure, i.e., the DHS, DHS$_e$ and LDHS models, generally show better performance than Lasso, GLasso, CAP$_{\ell_4}$, CAP$_{\ell_\infty}$ and HP. This demonstrates that whenever features can be organized into a deeper hierarchical structure, using the correct (deeper) hierarchical structure will allow a lower prediction error compared to models with shallow structure, e.g., groups or a hierarchical structure of height 2. The GLasso, CAP$_{\ell_4}$ and CAP$_{\ell_\infty}$ methods show comparable results since all of them belong to the CAP family and may acquire similar information from the feature groups. The DHS and DHS$_e$ methods show comparable performance, while the LDHS method, which learns the hierarchical structure automatically, achieves the lowest MSE in two of all the three settings.

Moreover, in Fig. 2, we show the estimated parameter $\boldsymbol{\sigma}$ in the DHS method and $\boldsymbol{\omega}$ in the LDHS method when $m = 3$. The estimated $\boldsymbol{\sigma}$ in DHS$_e$ is similar to that in DHS, hence we omit its result here. In Fig. 2(b), the estimated $\boldsymbol{\sigma}$

Table 2: The MSE's (in terms of 'mean±std') of different methods on the Traffic Volume data.

	Task 1	Task 2	Task 3	Task 4	Task 5
Lasso	0.1066±0.0179	0.1366±0.0279	0.1155±0.0231	0.1215±0.0132	0.1632±0.0352
GLasso	0.1042±0.0190	0.1185±0.0201	0.1030±0.0228	0.0934±0.0122	0.1725±0.0264
CAP$_{\ell_4}$	0.0939±0.0173	0.1122±0.0164	0.0944±0.0180	**0.0864±0.0109**	0.1609±0.0251
CAP$_{\ell_\infty}$	0.0977±0.0172	0.1179±0.0222	0.1029±0.0193	0.0925±0.0136	0.1800±0.0223
HP	0.0869±0.0189	0.1087±0.0270	0.0963±0.0196	0.0915±0.0133	0.1403±0.0299
DHS	0.0839±0.0187	0.1049±0.0262	0.0929±0.0187	0.0884±0.0132	0.1374±0.0284
DHS$_e$	0.0843±0.0189	0.1051±0.0262	0.0935±0.0189	0.0890±0.0128	0.1381±0.0290
LDHS	**0.0838±0.0186**	**0.1048±0.0262**	**0.0928±0.0186**	0.0882±0.0126	**0.1373±0.0284**

well matches the hierarchical structure. For example, the estimated $\boldsymbol{\sigma}$ along the path from the root to the last four features are generally small. According to the formulation of the DHS model, these σ's will give heavier penalizations on the corresponding feature weights and as a consequence, we can obtain small feature weights, which match the true values of $\mathbf{w}^*$. Similarly, for the LDHS model, the parameter $\boldsymbol{\omega}$ is an $m \times d$ matrix. We show the estimated $\boldsymbol{\omega}$ in Fig. 2(c), from which we can generally observe a hierarchical structure by comparing their values, and this hierarchical structure well matches the ground truth. This result demonstrates that the LDHS method is able to recover the hierarchical structure.

Experiments on Real-World Datasets

In this section, we experiment on three real-world datasets including the traffic volume data (Han and Zhang 2015b), the breast cancer data (Jacob, Obozinski, and Vert 2009) and the Covtype data. In these datasets, the hierarchical structure over the features is not available. By following (Kim and Xing 2010), we use a simple hierarchical k-means clustering method to generate the hierarchical structure on the features. Specifically, we perform k-means clustering to split the features into two groups, and for each group we recursively perform k-means clustering to obtain four sub-groups. Therefore, the resultant hierarchical structure has a height of $m = 3$ and hence in the LDHS method, the height of the learned hierarchical structure is also set to 3. The following experiments show that such a simple hierarchical structure is sufficient to obtain good performance for the proposed methods.

First we experiment on the traffic volume data. This dataset collects the traffic volumes from 136 entries (treated as features) and the traffic volumes through some exits (treated as response) in a highway traffic network. We choose 5 exits with the highest volumes in the network to form 5 learning tasks, each of which is to use the volumes from entries to predict the volume through a specific exit. There are totally 384 data samples. These tasks are regression problems and we use the square loss for all the methods. We randomly select 80% and 20% samples for training and testing, respectively. The regularization parameters are chosen from the same candidate set as used in the synthetic setting via 5-fold cross-validation. We use the MSE, i.e., $\frac{1}{n}\|\mathbf{y}-\hat{\mathbf{y}}\|_2^2$, to measure the performance of all the methods. The results on the 5 tasks are given in Table 2. From the results, we observe that in 4 out of the 5 tasks, the proposed DHS, DHS$_e$ and LDHS methods outperform the baseline methods and the LDHS method achieves the lowest MSE. This demonstrates again that a deeper tree structure can provide a better feature structure in this data and our proposed methods can take advantage of this structure.

Table 3: Results on the Breast Cancer (top) and Covtype (bottom) datasets.

	ACC (%)	SEN (%)	SPE (%)
Lasso	76.81±3.54	91.48±4.25	71.99±6.22
GLasso	80.69±3.08	94.42±3.67	76.24±4.77
CAP$_{\ell_4}$	81.53±3.08	**95.47±3.25**	76.96±4.97
CAP$_{\ell_\infty}$	80.97±3.33	**95.47±3.25**	76.23±5.04
HP	79.68±3.94	92.22±3.44	75.48±6.46
DHS	82.54±3.78	91.59±5.30	79.56±4.96
DHS$_e$	**83.49±3.48**	92.84±5.53	**80.41±4.32**
LDHS	82.38±3.71	91.59±7.16	79.36±3.75

	ACC (%)	SEN (%)	SPE (%)
Lasso	74.61±0.28	67.03±1.24	72.58±0.73
GLasso	74.97±0.28	70.91±1.06	79.24±0.56
CAP$_{\ell_4}$	74.81±0.11	70.33±0.19	79.51±0.13
CAP$_{\ell_\infty}$	74.80±0.10	70.30±0.18	79.52±0.14
HP	75.39±0.08	**74.31±0.18**	76.54±0.10
DHS	75.58±0.07	73.86±0.14	77.39±0.09
DHS$_e$	75.37±0.07	71.12±0.32	**79.84±0.30**
LDHS	**75.60±0.08**	74.04±0.15	77.24±0.10

Next, we conduct experiments on the breast cancer and Covtype datasets which have been studied in (Jacob, Obozinski, and Vert 2009; Han and Zhang 2015a). The breast cancer dataset contains genes in 295 tumors, among which 78 of them are metastatic while 217 are non-metastatic. Hence, the tasks here are binary classification problems. We use the square loss for all methods. It has been shown that in this dataset, some latent hierarchical structure exists. Similar to (Jacob, Obozinski, and Vert 2009; Han and Zhang 2015a), we select 300 most correlated genes to the outputs as the feature representation, and alleviate the class imbalance problem by duplicating the positive samples twice. The Covtype dataset aims to predict the forest cover type from collected cartographic variables. The problem is originally a multi-class classification problem and it is transformed into a binary classification problem in (Chang and Lin 2011). There are $n = 581,012$ examples and the feature dimensionality is $d = 54$. We evaluate all the methods on these two datasets based on three mesuares, i.e., the accuracy (ACC), sensitivity (SEN), and specificity (SPE), similar to (Yang et al. 2012; Han and Zhang 2015a). By following (Yang et al. 2012; Han and Zhang 2015a), 50%, 30% and 20% of the data are randomly chosen for training, validation, and testing, respectively. The regularization parameters are selected from the same candidate sets as described in the previous experiments. The top table in Table 3 shows the average results over 10 repetitions on the breast cancer data. According to the results, hierarchical methods with deep tree structure generally show more accurate predictions. The DHS$_e$ method achieves the best performance in this case. The CAP$_{\ell_4}$ and CAP$_{\ell_\infty}$ have better sensitivities than the proposed methods, implying that they can recover more true positive examples, while their SPE is lower. According to results reported in the bottom table of Table 3, we can see that the HP, DHS, DHS$_e$, and LDHS generally outperform the Lasso and GLasso methods on the Covtype data. Again, the LDHS method, which learns the tree structure, obtains the best accuracy.

Conclusions

In this paper, we study the problem of learning (from) deep hierarchical structure step by step. In the first step, we propose the convex DHS method to learn from hierarchical structure with an arbitrary height. Secondly, we propose the DHS$_e$ method, a variant of the DHS method, to learn the exponents from data. Finally, we propose the LDHS method to learn the hierarchical structure since it may be unavailable in some applications.

In our future work, we will learn the exponents of the LDHS method in a way similar to the DHS$_e$ method.

Acknowledgment

This research has been supported by NSFC (61473087 and 61673202).

References

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2(1):183–202.

Bondell, H., and Reich, B. 2007. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics* 64:115–123.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. New York, NY: Cambridge University Press.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):27.

Figueiredo, M. A. T., and Nowak, R. D. 2016. Ordered weighted L1 regularized regression with strongly correlated covariates: Theoretical aspects. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 930–938.

Gong, P.; Zhang, C.; Lu, Z.; Huang, J.; and Ye, J. 2013. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *Proceedings of the 30th International Conference on Machine Learning*, 37–45.

Hallac, D.; Leskovec, J.; and Boyd, S. P. 2015. Network lasso: Clustering and optimization in large graphs. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 387–396.

Han, L., and Zhang, Y. 2015a. Discriminative feature grouping. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

Han, L., and Zhang, Y. 2015b. Learning tree structure in multi-task learning. In *Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Hocking, T.; Vert, J.-P.; Bach, F.; and Joulin, A. 2011. Clusterpath: An algorithm for clustering using convex fusion penalties. In *Proceedings of the 28th International Conference on Machine Learning*, 745–752.

Jacob, L.; Obozinski, G.; and Vert, J.-P. 2009. Group lasso with overlap and graph lasso. In *Proceedings of the 26th International Conference on Machine Learning*, 433–440.

Kim, S., and Xing, E. P. 2010. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning*, 543–550.

Szafranski, M.; Grandvalet, Y.; and Morizet-Mahoudeaux, P. 2007. Hierarchical penalization. In Platt, J. C.; Koller, D.; Singer, Y.; and Roweis, S. T., eds., *Advances in Neural Information Processing Systems 20*, 1457–1464.

Tibshirani, R.; Saunders, M.; Rosset, S.; Zhu, J.; and Knight, K. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society Series B* 67(1):91–108.

Yang, S.; Yuan, L.; Lai, Y.-C.; Shen, X.; Wonka, P.; and Ye, J. 2012. Feature grouping and selection over an undirected graph. In *Proceedings of the 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 922–930.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(1):49–67.

Zhao, P.; Rocha, G.; and Yu, B. 2009. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics* 37(6A):3468–3497.