Review of Graph Sparsity Methods and Sparsity in Convolutional Neural Networks

Abstract

In this paper we present a review of algorithms and consistency results for sparsity methods when structured information is known about the covariates. Results discussed include a generalization of the consistency of group lasso to the case of overlapping groups, a connection between general sparsity methods and information theory, and a way to combine group and graph sparsity regularizers. Finally, we discuss some recent work on the use of group lasso for regularization in convolutional neural networks.

1 Introduction

Over the last decade a variety of algorithms and theoretical results related to minimizing an empirical risk $L(\beta)$ as a function of some $\beta \in \mathbb{R}^d$ when d is large but β is sparse have emerged. Two popular methods for performing this minimization are the lasso method [7], where a regularization term of $\|\beta\|_1 = \sum_{j=1}^{d} |\beta_j|$ is used, and basis pursuit [6], where nonzero elements of β are chosen greedily.

In many applications, however, further information is known about the covariates β: for instance, it may be known that the elements of β are clustered into a few select groups, such as groups of genes, and the goal is to determine which of these groups of genes are important in relating an output vector Y to a feature matrix X through $Y \approx X\beta$. In this case, a generalization of the lasso known as the group lasso can be used [9]: the group lasso penalty of $\beta \in \mathbb{R}^d$ is the sum of the l_2 norms of the restriction of β to each of the groups. In this paper we provide a review of techniques introduced to minimize empirical risk $\|Y - X\beta\|_2^2$ when structured information about β beyond the simple existence of groups is known. For instance, the d covariates in β may lie on some graph such that an edge between two covariates means that the covariates are more likely to be related.

In Sections 2 through 4, we review prior results which propose a variant of group lasso for overlapping groups, give an information theoretical framework for general sparsity methods, and combine the ideas of group sparsity and graph sparsity. In Section 5, we review some recent work using group sparsity to regularize weights in convolutional neural networks and present a few experiments.

Notation. We use the following notation: the number of training examples is denoted by n, and the index set of coefficients is denoted by $\mathcal{I}_d = \{1, \ldots, d\}$. Then we normally have $X \in \mathbb{R}^{n \times d}, Y \in \mathbb{R}^n$. For a vector $\beta \in \mathbb{R}^d$, we let $\text{supp}(\beta)$ denote the subset of $\mathcal{I}_d$ corresponding to the nonzero covariates of β. For a set A, $\mathcal{P}(A)$ denotes the power set (set of all subsets) of A. Given a set of groups $\mathcal{G} \subset \mathcal{P}(\mathcal{I}_d)$, a group $g \in \mathcal{G}$, and a vector $\beta \in \mathbb{R}^d$, we let $\beta_g \in \mathbb{R}^d$ denote the vector whose components are all 0 apart from those in the group g, which are equal to the corresponding coefficients of β. For a graph G with vertex set V and edge set E, we write $G = (V, E)$, and $V = V(G), E = E(G)$. Unless otherwise stated, $\|\beta\|$ refers to the l_2 norm $\|\beta\|_2$ of the vector $\beta \in \mathbb{R}^d$. We use standard complexity notation: $O(\cdot), \Omega(\cdot), \Theta(\cdot)$, etc.

2 Overlapping group sparsity

Original guarantees on group sparsity were proved with the assumption that the groups do not overlap. Given a set of groups $\mathcal{G} \subset \mathcal{P}(\mathcal{I}_d)$, the group lasso norm, namely $R_{group}(\beta) = \sum_{g\in\mathcal{G}} \|\beta_g\|$, can also be considered when groups in $\mathcal{G}$ overlap. The resulting norm still has the property that it is not differentiable at points β when some of the β_g are 0. The solution to an empirical risk minimization problem regularized with the norm R_{group} tends to converge to such a solution β with many singularities, meaning that the support of β is a subset of the complement of the union of several groups in $\mathcal{G}$. Jacob et al. [5] introduced a new norm $R_{overlap}$ such that regularizing with $R_{overlap}$ produces β with support that is a union of a few groups in $\mathcal{G}$. To define $R_{overlap}$, let $\mathcal{V}_\mathcal{G} \subset \mathbb{R}^{d\times|\mathcal{G}|}$ denote the set of $|\mathcal{G}|$-tuples of vectors $\mathbf{v} = (v_g)_{g\in\mathcal{G}}$, such that for each $g \in \mathcal{G}$, $v_g \in \mathbb{R}^d$ satisfies $\text{supp}(v_g) \subset g$. Then define:

$$R_{overlap}(\beta) = \inf_{\mathbf{v}\in\mathcal{V}_\mathcal{G}, \sum_{g\in\mathcal{G}} v_g = \beta} \sum_{g\in\mathcal{G}} \|v_g\|. \tag{1}$$

Note that the optimization problem (1) is convex and the objective $\sum_{g\in\mathcal{G}} \|v_g\|$ is coercive, so there is some $\beta \in \mathbb{R}^d$ that achieves the minimum. Let $V(\beta) \subset \mathcal{V}_\mathcal{G}$ denote the set of all $|\mathcal{G}|$-tuples of vectors $\mathbf{v}$ which reach the minimum in (1). If $V(\beta)$ contains a single element $\mathbf{v}$, then $\cup_{g:v_g\neq 0} g$ is said to be the *group-support* of $\bar{\beta}$. For some $\mathbf{v}$ that achieves the minimum in the above definition, the support of β is then in the union of the supports of each $v_g \neq 0$.

Given $Y = X\bar{\beta} + \epsilon$, for $Y, \epsilon \in \mathbb{R}^n, X \in \mathbb{R}^{n\times d}$, and ϵ a random vector whose elements are independent with bounded variance, the main theorem in [5] states necessary and sufficient conditions for an algorithm that solves $\min_{\beta\in\mathbb{R}^d} \|X\beta - Y\|_2^2 + \lambda R_{overlap}(\beta)$ to learn the support of $\bar{\beta}$ with high probability. To state these conditions, a lemma giving the dual of the optimization problem in (1) is needed:

Lemma 1 ([5]) *We have:*

1. *For $\beta \in \mathbb{R}^d$,*

$$R_{overlap}(\beta) = \sup_{\alpha\in\mathbb{R}^d : \forall g\in\mathcal{G} \|\alpha_g\|\leq 1} \alpha^T\beta. \tag{2}$$

   Also define $\mathcal{G}_1 = \{g \in \mathcal{G} : \exists \mathbf{v} = (v_g)_g \in V(\beta), v_g \neq 0\}$ as the set of groups g such that some group-decomposition $\mathbf{v}$ of β has $v_g \neq 0$. Let $J_1 = \cup_{g\in\mathcal{G}_1} g$. Then for any optimum α in (2), $\alpha_{J_1} \in \mathbb{R}^d$ is uniquely defined; we write $\alpha_{J_1}(\beta)$ to denote this vector.

2. *Some $\alpha \in \mathbb{R}^d$ solves (2) if and only if there is $\mathbf{v} = (v_g)_{g\in\mathcal{G}} \in V(\beta)$ such that for all $g \in \mathcal{G}$, if $v_g \neq 0$, then $\alpha_g = v_g/\|v_g\|_2$, and otherwise $\|\alpha_g\|_2 \leq 1$.*

Condition 2 of the above lemma states that α solving (2) is essentially a "bounded aggregation" of any tuple of vectors $\mathbf{v} \in V(\beta)$, and condition 1 states that the restriction of α to J_1 is uniquely defined. The conditions that are sufficient for the solution of $\min_{\beta\in\mathbb{R}^d} \|X\beta - Y\|_2^2 + \lambda R_{overlap}(\beta)$ to learn $\text{supp}(\bar{\beta})$ with high probability are:

1. $\Sigma := \frac{1}{n} X^T X \succ 0$.
2. There is some neighborhood of $\bar{\beta}$ in which (1) has a unique solution.
3. Define, for $S, T \subset \mathcal{I}_d$, $X_S \in \mathbb{R}^{n\times|S|}$ as the submatrix of X whose columns correspond to the elements of S, and $\Sigma_{ST} = \frac{1}{n} X_S^T S_T$; moreover let $\mathcal{G}_1 = \{g : v_g \neq 0\}$, where $\mathbf{v} = (v_g)_{g\in\mathcal{G}}$ is the unique solution of (1) for the given $\bar{\beta}$, and $J_1 = \cup_{g\in\mathcal{G}_1} g$ be the group-support of $\bar{\beta}$. Then the condition is: for all $g \in \mathcal{I}_d \backslash \mathcal{G}_1$,

$$\|\Sigma_{gJ_1}\Sigma_{J_1J_1}^{-1}\alpha_{J_1}(\bar{\beta})\|_2 \leq 1. \tag{3}$$

The third condition above essentially states that the columns of X corresponding to groups in $\mathcal{G} - J_1$ are not too correlated with the columns of X corresponding to groups in J_1, when weighted by $\alpha_{J_1}(\bar{\beta})$. Note that (3) involves a slight abuse of notation, since technically $\alpha_{J_1}(\bar{\beta}) \in \mathbb{R}^d$; however, in (3), we condense it to a vector in $\mathbb{R}^{|J_1|}$ by ignoring coefficients of $\alpha_{J_1}(\bar{\beta})$ corresponding to $\mathcal{I}_d \backslash J_1$ (which must be 0 anyway). We can now formally state the consistency result of [5]:

Theorem 2 *If conditions 1-3 above hold, $\lambda_n \to 0$, and $\lambda_n\sqrt{n} \to \infty$, then the solution $\hat{\beta} = \arg\min_{\beta\in\mathbb{R}^d} \|X\beta - Y\|_2^2 + \lambda_n R_{overlap}(\beta)$ has the same group-support J_1 as $\bar{\beta}$ with high probability (as $n \to \infty$), and $\|\bar{\beta} - \hat{\beta}\|_2^2 \to 0$ in probability as $n \to \infty$.*

As a specific case, the overlapping group norm $R_{overlap}(\beta)$ can be used to enforce sparsity on the variables $\mathcal{I}_d$ when a graph structure $G = (\mathcal{I}_d, E)$ is given. In particular, $\mathcal{G}$ can be chosen as the set of all paths in G of some fixed length $\ell \geq 1$. As ℓ is increased, we expect the resulting β to have larger connected components. As a special case we have $\ell = 1$, so that the resulting β will tend to have pairs of covariates that are connected by an edge in E set to non-zero values. Of course, care must be taken to ensure that $\cup_{g\in\mathcal{G}} = \mathcal{I}_d$, or else some covariates are entirely ignored. We will further discuss such notions of graph sparsity in later sections.

3 Sparsity and coding complexity

In this section we discuss an approach of studying general structured sparsity from a coding complexity perspective [4] that generalizes the notion of group sparsity discussed in the previous section. We begin with a definition of coding complexity:

Definition 3 ([4]) *A function $cl(S)$ defined on $S \subset \mathcal{I}_d$ is a coding length if $\sum_{S\subset\mathcal{I}_d} 2^{-cl(S)} \leq 1$. We define $cl(\emptyset) = 0$. The coding complexity of S corresponding to cl is defined by $c(S) = |S| + cl(S)$. Given a coding complexity $c : \mathcal{P}(\mathcal{I}_d) \to \mathbb{R}^{\geq 0}$, the structured sparse coding complexity of a coefficient vector $\beta \in \mathbb{R}^d$ is $c(\beta) = \min\{c(S) : supp(\beta) \subset S\}$.*

It is well known [2] that a function cl is a coding length if and only if there is a prefix-free coding scheme that encodes any $S \subset \mathcal{I}_d$ with at most $\mathrm{cl}(S)$ bits. Thus to define cl we will often construct codings of S. Corresponding to a coding complexity c is the following natural extension of L_0 regularization [4]: given an $n \times d$ data matrix X and an output vector $Y \in \mathbb{R}^d$, we want to find

$$\hat{\beta} = \arg\min_{\beta\in\mathbb{R}^d} \|X\beta - Y\|^2, \text{ subject to } c(\beta) \leq k. \tag{4}$$

Alternatively, the Lagrangian formulation $\hat{\beta} = \arg\min_{\beta\in\mathbb{R}^d}(\|X\beta - Y\|^2 + \lambda c(\beta))$ can be considered.

Huang et al. [4] introduced a greedy algorithm with provable guarantees to solve this coding complexity regularization problem. Before stating their results, we first explain how specific choices of the regularization term $c(\beta)$ generalize traditional notions of sparsity. To construct such choices of c, a subset $\mathcal{B} \subset \mathcal{P}(\mathcal{I}_d)$ is chosen such that $\mathcal{I}_d = \cup_{B\in\mathcal{B}} B$ and for each $j \in \mathcal{I}_d$, $\{j\} \in \mathcal{B}$; such a $\mathcal{B}$ is called a block set, and its elements are called blocks. Then, if cl_0 is a coding length on $\mathcal{B}$, defining $\mathrm{cl}_{\mathcal{B}}(B) = \mathrm{cl}_0(B) + 1$ for $B \in \mathcal{B}$ leads the function $\mathrm{cl}_{\mathcal{B}}(S) := \min\left\{\sum_{j=1}^b \mathrm{cl}(B_j) : S = \cup_{j=1}^b B_j, B_j \in \mathcal{B}\right\}$ to be a well-defined coding length [4]. The greedy algorithm mentioned above works by iterating through the elements of some block set $\mathcal{B}$ at each step and selecting the "best" block to include in the support of the learned vector.

If we choose $\mathcal{B}_1 = \{\{1\}, \ldots, \{d\}\}$, then $\mathrm{cl}_0(\{j\}) = \log_2 d$ is a coding length on $\mathcal{B}_1$; this coding length induces the coding length $\mathrm{cl}_{\mathcal{B}_1}(S) = |S|\log_2(2d)$ on $\mathcal{I}_d$. The coding complexity of S corresponding to cl is now given by $c(S) = |S|(1 + \log_2(2d))$; the corresponding coding complexity of a vector $\beta \in \mathbb{R}^d$ is proportional to the l_0 norm of β. Now suppose we are instead given a partition $\mathcal{I}_d = \cup_{j=1}^q G_j$ into q non-overlapping groups. We let $\mathcal{B}_G = \{G_1, \ldots, G_q\}$, and choose $\mathcal{B} = \mathcal{B}_G \cup \mathcal{B}_1$, and define a coding length cl_0 on $\mathcal{B}$ by letting $\mathrm{cl}_0(\{j\}) = \infty$ and $\mathrm{cl}_0(G_j) = \log_2 q$. Then the induced coding length on $\mathcal{P}(\mathcal{I}_d)$ is given by $\mathrm{cl}_{\mathcal{B}}(B) = g\log_2(2q)$ if B is the union of g distinct groups, and $\mathrm{cl}(B) = \infty$ otherwise. This coding length corresponds to the notion of group sparsity.

Finally, the notion of coding length can be used to define sparsity on a graph: suppose we are given a graph G such that $V(G)$ contains as a subset each element of $\mathcal{I}_d$ (G may also contain supplementary vertices). The idea of graph sparsity is that nodes which induce connected subgraphs of G are more likely to correspond to covariates that are grouped together than nodes that are "far apart" on G. Formally, we have:

Lemma 4 ([4]) *For a given graph G, there is some constant C_G such that for any probability distribution p on $V(G)$, $cl(S) = C_G|S| + \sum_{j=1}^q \min_{v\in S_j} \log_2(1/p(v))$ is a coding length on G.*

Here we write $S = S_1 \cup \cdots \cup S_q$ to denote the connected components of the induced subgraph of S in G.

If we take $V(G) = \mathcal{I}_d$, $p(v) = 1/d$ for all $v \in V(G)$, then sets S which induce subgraphs with fewer connected components of G clearly have smaller coding length. The constant C_G can be taken as $\Delta(G) + 1$ for any G, where $\Delta(G)$ denotes the maximum degree of G [4].

3.1 Error bounds

In this section we describe error bounds regarding the relation between the solution $\hat{\beta}$ to (4) and a target vector $\bar{\beta}$. These bounds operate under the assumption that the elements Y_i of the vector Y are independent sub-Gaussians; that is, there is $\sigma \geq 0$ such that for all i, and for all $t \in \mathbb{R}$, $E[e^{t(Y_i - E[Y_i])}] = e^{\sigma^2 t^2/2}$. We also define, for $S \subset \mathcal{I}_d$,

$$\rho_-(S) = \inf \left\{ \frac{1}{n} \|X\beta\|_2^2 / \|\beta\|_2^2 \ : \ \text{supp}(\beta) \subset S \right\}, \quad \rho_-(k) = \inf\{\rho_-(S) \ : \ S \subset \mathcal{I}_d,\ c(S) \leq k\}.$$

$\rho_+(S)$ and $\rho_+(k)$ are defined analogously, with the $\inf$ replaced by $\sup$. These definitions are similar to that of the restricted isometry property of the matrix X, except they are weaker since the ratio $\|X\beta\|_2^2/\|\beta\|_2^2$ is constrained only for certain β (those whose support is in some set with small coding complexity). The below theorem gives bounds on the quality of the solution of (4) under the assumption of independent, sub-Gaussian noise:

Theorem 5 ([4]) *Fix $\bar{\beta} \in \mathbb{R}^d$. Then with probability at least $1 - \eta$, for all $\epsilon \geq 0$ and $\hat{\beta} \in \mathbb{R}^d$ with $\|X\hat{\beta} - Y\|^2 \leq \|X\bar{\beta} - Y\|^2 + \epsilon$, we have*

$$\|X\hat{\beta} - E[Y]\|_2 \leq \|X\bar{\beta} - E[Y]\|_2 + \sigma O\left(\sqrt{\log(1/\eta)} + \sqrt{c(\hat{\beta})}\right) + O(\sqrt{\epsilon}) \tag{5}$$

$$\|\hat{\beta} - \bar{\beta}\|_2^2 \leq \frac{O\left(\|X\bar{\beta} - E[Y]\|^2 + \sigma^2 c(\hat{\beta}) + \sigma^2 \log(1/\eta)\right)}{n\rho_-(c(\hat{\beta}) + c(\bar{\beta}))}. \tag{6}$$

Recall that k is the coding complexity, so that we are given $c(\bar{\beta}) \leq k$ and solve for $c(\hat{\beta}) \leq k$ in (4). Dividing both sides of (5) by $\sqrt{n}$ gives that as $n \to \infty$ and for fixed k (in fact, for any sequence of k such that $k/n \to 0$), the root mean squared error for $\hat{\beta}$, namely $\|X\hat{\beta} - E[Y]\|_2/\sqrt{n}$ approaches the root mean squared error for the optimal $\bar{\beta}$, namely $\|X\bar{\beta} - E[Y]\|_2/\sqrt{n}$. Therefore, the above theorem gives a general condition for an upper bound on the number of training examples n needed to outperform methods such as standard sparsity. In particular, whenever $c(\bar{\beta}) = O(|\text{supp}(\bar{\beta})| + \text{cl}(\text{supp}(\bar{\beta})) \ll |\text{supp}(\bar{\beta})| \ln d$, then the number of examples n needed to force $\|X\hat{\beta} - E[Y]\|_2 - \|X\bar{\beta} - E[Y]\|_2 < \delta$ for a given δ is much smaller than the number of examples $|\text{supp}(\bar{\beta})| \ln d$ from standard sparsity.

Equation (6) has similar consequences in the realm of compressed sensing; in particular, for compressed sensing, we take $\sigma = 0$ (so there is no noise), which implies that $X\bar{\beta} = E[Y]$. Then (6) becomes: $\|\bar{\beta} - \hat{\beta}\|_2 = 0$ as long as $\rho_-(c(\bar{\beta}) + c(\hat{\beta})) > 0$. The below theorem gives a sufficient condition for this to be the case:

Theorem 6 ([4]) *Suppose the elements of $X \in \mathbb{R}^{n \times p}$ are iid Gaussian $\mathcal{N}(0, 1)$. Given $t, k > 0$ and $\delta \in (0, 1)$, let $n \geq \frac{8}{\delta^2}(\ln 3 + t + k \ln(1 + 8/\delta))$. Then with probability at least $1 - e^{-t}$, $\rho_-(k) \geq 1 - \delta$ and $\rho_+(k) \leq 1 + \delta$.*

In order to have $\rho_-(c(\bar{\beta}) + c(\hat{\beta})) > 0$ with probability 0.99, the above theorem shows that it suffices to take $n = \Omega(c(\bar{\beta}) + c(\hat{\beta}))$ which corresponds to the previous result (5) which also states that n must be on the order of the coding complexity.

3.2 Optimization with coding complexity regularization

As the optimization problem (4) is generally computationally hard, Huang et al. [4] proposed a greedy algorithm which generalizes orthogonal matching pursuit (OMP). The algorithm is shown

in Algorithm 3.2; it takes as input a block set $\mathcal{B} \subset \mathcal{P}(\mathcal{I}_d)$ which it uses to build up the set $S^{(t)}$ (as a union of blocks of $\mathcal{B}$) greedily. At each step, it maximizes the variance of the residual when projected onto columns of X that have not yet been chosen; for $S \subset \mathcal{I}_d$, this projection is computed by $P_S = X_S(X_S^T X_S)^{-1} X_S^T$, where X_S denotes the sub-matrix of X whose columns correspond to the elements of S. Note that it is not imperative that the actual coding complexity $c(\beta)$ used in (4) be the coding complexity $c_{\mathcal{B}}(\beta)$ induced by $\mathcal{B}$. However, Huang et al. showed that if the coding complexity induced by $\mathcal{B}$ approximates c, then their algorithm will achieve an approximate solution. To state their result formally, given $\mathcal{B} \subset \mathcal{P}(\mathcal{I}_d)$, we let $\rho_0(\mathcal{B}) = \max_{B \in \mathcal{B}} \rho_+(B)$ and $c_0(\mathcal{B}) = \max_{B \in \mathcal{B}} c(B)$. Then we have:

Theorem 7 ([4]) *Suppose the coding function c in the empirical risk minimization problem (4) is sub-additive. Let $\bar{\beta}, \epsilon$ be such that*

$$\epsilon \in (0, \|Y\|_2^2 - \|X\bar{\beta} - Y\|_2^2], \qquad k \geq \frac{\rho_0(\mathcal{B}) c_{\mathcal{B}}(\bar{\beta})}{\rho_-(k + c(\bar{\beta}))} \ln \frac{\|Y\|_2^2 - \|X\bar{\beta} - Y\|_2^2}{\epsilon}.$$

Then when Algorithm 3.2 stops, we have $\|X\hat{\beta} - Y\|_2^2 \leq \|X\bar{\beta} - Y\|_2^2 + \epsilon$.

Suppose $\frac{\rho_0(\mathcal{B})}{\rho_-(k + c(\bar{\beta}))} = \Theta(1)$ (as is often the case in practice), $c(\bar{\beta}) = \Theta(c_{\mathcal{B}}(\bar{\beta}))$ (meaning that $c_{\mathcal{B}}$ is a good approximation of c) and ϵ is chosen a factor of 2^a smaller than $\|Y\|_2^2 - \|X\bar{\beta} - Y\|_2^2$. Then the above theorem implies that we can obtain an error $\|X\hat{\beta} - Y\|_2^2$ of only $\|Y\|_2^2/2^a$ more than the minimum error $\|X\bar{\beta} - Y\|_2^2$ yet can ensure that k (which is approximately the coding complexity of $\hat{\beta}$) is greater than $c(\bar{\beta})$ by at most a times a constant factor. Combined with (5) from Theorem 5, which requires an upper bound on $c(\hat{\beta})$, this gives a bound on when StructOMP is guaranteed to converge to the optimal solution.

Algorithm 1 StructOMP (Huang et al., 2011)

1: Input: $(X, Y), \mathcal{B} \subset \mathcal{P}(\mathcal{I}_d), k > 0$. Initialize: $S^{(0)} = 0, \beta^{(0)} = 0$.
2: **for** $t = 1, 2, \ldots$ **do**
3: Choose $B^{(t)} \in \mathcal{B}$ by $B^{(t)} = \arg\min_{B \in \mathcal{B}} \frac{\|P_{B \setminus S^{(t-1)}}(X\beta^{(t-1)} - Y)\|_2^2}{c(B \cup S^{(t-1)}) - c(S^{(t-1)})}$
4: Set $S^{(t)} = S^{(t-1)} \cup B^{(t)}$.
5: Set $\beta^{(t)} = \arg\min_{\beta \in \mathbb{R}^d} \|X\beta - Y\|_2^2$ subject to $\text{supp}(\beta) \subset S^{(t)}$.
6: If $c(\beta^{(t)}) > k$ break and return $S^{(t)}, \beta^{(t)}$.
7: **end for**

4 Group-Graph regularization

Dai et al. [3] combined the ideas of graph sparsity and group sparsity to create a regularization term adapted to the case in which the covariates $\mathcal{I}_d$ are grouped into a set of non-overlapping groups, and that there is some graph structure on the individual groups (rather than on the covariates $\mathcal{I}_d$). Formally, suppose that we have a partition $\mathcal{I}_d = B_1 \cup \cdots \cup B_q$. Let $\mathcal{B} = \{B_1, \ldots, B_q\}$ and suppose that we are given some graph $H = (\mathcal{B}, E)$. Now let $\mathcal{G}$ denote the set of all paths P on H; each path $P \in \mathcal{G}$ is given some positive weight $\eta_P > 0$. Then the quantity

$$R_{g^2}^0(\beta) = \min_{\mathcal{G}_1 \subset \mathcal{G}} \left\{ \sum_{P \in \mathcal{G}_1} \eta_P \text{ s.t. } \text{supp}((\|\beta\|_{B_1}, \ldots, \|\beta\|_{B_q})) \subset \bigcup_{P \in \mathcal{G}_1} P \right\} \tag{7}$$

is similar to the norm $R_{overlap}$ defined in Section 2 in that it tends to favor vectors β whose nonzero variables belong to groups that lie on a path in H (note the fact that here the vertices of H are groups in $\mathcal{B}$, whereas in Section 2 the vertices of H are elements of $\mathcal{I}_d$). Following the notation of [3], we write $\sigma(\beta) = (\|\beta_{B_1}\|, \ldots, \|\beta_{B_q}\|) \in \mathbb{R}^q$. Note that $R_{g^2}^0$ is non-convex; to optimize an objective with a regularization term of $R_{g^2}^0$, Dai el al. took the approach of using a convex relaxation; note that this approach is different from the one described in Section 3, but similar to the one described in Section 2 (when the norm $R_{overlap}$ was already convex). To describe this convex relaxation, we enumerate all paths in $\mathcal{G}$ by $\mathcal{G} = \{P_1, \ldots, P_{|\mathcal{G}|}\}$, and let $N \in \{0, 1\}^{q \times |\mathcal{G}|}$ be a $q \times |\mathcal{G}|$ matrix with $N_{ij} = 1$ if

vertex $B_i \in \mathcal{B}$ belongs to path P_j, and $N_{ij} = 0$ otherwise. We also let $\eta = (\eta_{P_j})_{P_j \in \mathcal{G}} \in \mathbb{R}_+^{|\mathcal{G}|}$ be the vector of path weights. Moreover, for a vector $\sigma \in \mathbb{R}^q$, we let $H(\sigma) \in \mathbb{R}^q$ be such that $H(\sigma)_j = 0$ if $\sigma_j = 0$, and $H(\sigma_j) = 1$ otherwise. Then (7) can be rewritten as the following integer linear programming problem:

$$R^0_{g^2}(\beta) = \min_{x \in \{0,1\}^{|\mathcal{G}|}} \{\eta^T x \text{ such that } Nx \geq H(\sigma(\beta))\}. \tag{8}$$

Note that $H(\sigma(\beta))_j = 1$ if and only if $\|\beta_{B_j}\| > 0$. The convex relaxation $R_{g^2}(\beta)$ is then defined by: $R_{g^2}(\beta) = \min_{x \in \mathbb{R}^{|\mathcal{G}|}} \{\eta^T x \text{ such that } Nx \geq \sigma(\beta)\}$.

4.1 Proximal algorithm

The regularization term $R_{g^2}(\beta)$ is then used to solve the following problem: $\hat{\beta} = \min_{\beta \in \mathbb{R}^d} L(\beta) + \lambda R_{g^2}(\beta)$, where L is a loss function (for instance, $L(\beta) = \|X\beta - Y\|_2^2$ for given X, Y). Since R_{g^2} is convex and coercive, as long as the proximal operator $\text{prox}_{R_{g^2}}$ can be computed quickly, then the fast iterative-shrinkage thresholding algorithm [1] can be used to solve for $\hat{\beta}$. Dai et al. showed that computation of $\text{prox}_{R_{g^2}}$ can be reduced to solving a flow problem on the graph H, which can be done in time polynomial in $|\mathcal{B}| = q$ and $|E|$. In order for this reduction to hold, the weights η_P must be chosen as follows: given the graph $H = (\mathcal{B}, E)$ as above, let $H' = (\mathcal{B} \cup \{s,t\}, E')$, where $E' = E \cup \{(s,u) : u \in \mathcal{B}\} \cup \{(v,t) : v \in \mathcal{B}\}$. Each edge $uv \in E'$, for $u, v \in V(H')$, is given some cost c_{uv}. Then for a path $P = (v_1, \ldots, v_\ell)$, with $v_1, \ldots, v_\ell \in \mathcal{B}$, we let $\eta_P = c_{sv_1} + \sum_{i=1}^{\ell-1} c_{v_i v_{i+1}} + c_{v_\ell t}$. Now let $\mathcal{F} = \{f : f \text{ is a network flow on } H'\}$, and for a flow f, $1 \leq j \leq q$, let $s_j(f)$ be the amount of flow through vertex $B_j \in \mathcal{B}$. The below theorem provides an efficient way of computing $\text{prox}_{R_{g^2}}(\beta)$:

Theorem 8 ([3]) *For $1 \leq j \leq q$, $i \in B_j$, $w^* = prox_{R_{g^2}}(\beta) = \arg\min_{w \in \mathbb{R}^d} \frac{1}{2}\|\beta - w\|^2 + R_{g^2}(w)$ is given by*

$$w_i^* = \min\left\{\beta_i, \frac{s_j(f^*)}{\|\beta_{B_j}\|}\beta_i\right\}, \tag{9}$$

with

$$f^* \in \arg\min_{f \in \mathcal{F}} \sum_{(u,v) \in E(H')} \left\{ f_{uv} c_{uv} + \frac{1}{2} \sum_{j=1}^{q} \max\left(\|\beta_{B_j}\| - s_j(f), 0\right)^2 \right\}.$$

f^* can be approximated to within a factor of ϵ in time poly$(|V(H)|, |E(H)|, \log(\max_j \|\beta_{B_j}\|_2/\epsilon))$.

In the case that $E = \emptyset$, note that $\mathcal{G}$ is simply the set of vertices of H, and $\eta_P = 1$ for each $P \in \mathcal{G}$, so that $R_{g^2}(\beta) = \sum_{j=1}^q \|\beta_{B_j}\|_2$ is equal to the group lasso norm of β. In this case if we take $c_{su} = c_{ut} = 1/2$ for each $u \in \mathcal{G}$, it is not too difficult to see that the proximal operator in (9) is equal to the soft group thresholding operator on β.

5 Group lasso in CNNs

In this section we describe recent work of Wen et al. [8] that uses a group lasso regularization term to train a convolutional neural network (CNN). Their primary motivation was to speed up the computation of convolutional layers, but as we shall see below, the group lasso also improves interpretability in this case. Suppose a CNN has L convolutional layers, with weights $W^{(1)}, \ldots, W^{(L)}$ in each layer. Following the notation of [8], we write $W^{(l)} \in \mathbb{R}^{N_l \times C_l \times M_l \times K_l}$, where N_l, C_l, M_l, and K_l denote the number of filters, input channels, filter height, and filter width, respectively, in layer l. The loss function in this setting can be written as $\mathcal{E}(W) = \mathcal{E}_D(W) + \lambda R(W) + R_{struct}(W)$, where $\mathcal{E}_D$ is the loss on the data, $R(W)$ is a non-structural regularizer on W (for instance, an l_2 norm of W), and R_{struct} is a group lasso regularizer that enforces structured sparsity on the weights W.

If the groups are chosen as $\{W^{(l)}_{n_l,:,:,:}\}_{1 \leq l \leq L, 1 \leq n_l \leq N_l}$, namely all weights corresponding to each filter in each layer, then we have $R_{struct}(W) = \lambda_n \sum_{l=1}^{l} \sum_{n_l=1}^{N_l} \|W^{(l)}_{n_l,:,:,:}\|_2$. Regularizing with this norm tends to zero out some of the filters in each layer, so the corresponding channels in the next layer are useless. Therefore, it makes sense to combine group lasso of filters and channels [8], as in $R_{struct}(W) = \lambda_n \sum_{l=1}^{l} \sum_{n_l=1}^{N_l} \|W^{(l)}_{n_l,:,:,:}\|_2 + \lambda_c \sum_{l=1}^{l} \sum_{n_l=1}^{N_l} \|W^{(l)}_{:,c_l,:,:}\|_2$. Note that for each

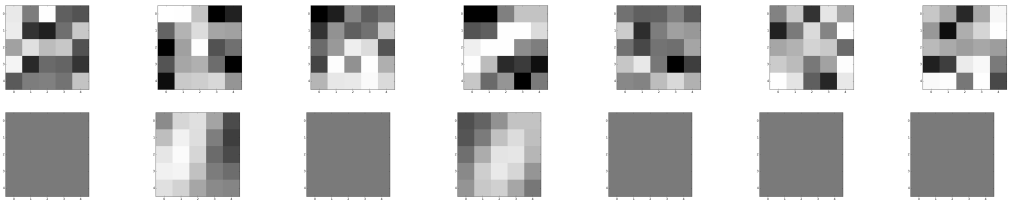

Figure 1: Sample of 7 of 32 layer 1 filters for MNIST. Top: no sparsity regularization. Bottom: filter/channel group lasso regularization

layer l, the weights $W_{n_l,:,:,:}$ and $W_{:,c_l,:,:}$ overlap (namely in the filter $W_{n_l,c_l,:,:}$). As the standard group lasso regularizer is used for $R_{struct}(W)$, the support of the learned W will tend to be in the complement of the union of several groups $W_{n_l,:,:,:}$ and $W_{:,c_l,:,:}$; it could be interesting to see if similar results are obtained if a framework similar to that in Section 2 for overlapping groups is used instead (since we can no longer write the prediction vector as $X\hat{\beta}$, some substantial changes would be needed to the algorithm in [5]). Wen et al. [8] also considered the regularization term given by group lasso with groups $W_{:,c_l,m_l,k_l}$ (where l, c_l, m_l, k_l collectively define a group).

In Figure 1 we present the results of some experiments involving the channel/filter regularizer implemented in LeNet with TensorFlow for MNIST handwriting data. The figure displays a sample of the 32 layer 1 convolutional weights; note that many of the filters are set to 0 when group lasso is used. Also note that the remaining weights are smoother and correspond more closely with patterns likely to be seen in images of handwritten digits; this is exactly consistent with those of [8]. With $\lambda_n = \lambda_c = 0.5$, and running the optimization on two iterations on the data, the test error was 1.4% without group lasso regularization and 7.9% with group lasso regularization. This much larger error can likely be decreased by training for more iterations or changing the optimization algorithm (TensorFlow's momentum optimizer was used). Indeed, Wen et al. [8] reported a test error of 0.9% without group lasso and between 0.8%-1.0% with filter/channel group lasso.

Conclusion In this review we have discussed several methods for learning a covariate vector $\bar{\beta}$ when we know structured relationships between the individual covariates, such as through a graph. We have also discussed a recent extension of traditional algorithms that solve, for instance, a least squares problem $\|X\beta - Y\|_2^2 + R(\beta)$ to the non-linear, non-convex case of neural networks. There is much interesting work to be done towards determining extensions of some of the bounds and consistency results in this review to the case of deep networks.

References

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

[2] Thomas Cover and Joy Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[3] Xin-Yu Dai, Jian-Bing Zhang, Shu-Jian Huang, Jia-Jun Chen, and Zhi-Hua Zhou. Structured sparsity with group-graph regularization. *Advancement of Artificial Intelligence*, 2015.

[4] Junchou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *Journal of Machine Learning Research*, 12:3371–3412, 2011.

[5] Laurent Jacob, Guillame Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. *Proceedings of the 26th International Conference on Machine Learning*, 2009.

[6] Michael Saunders Scott Chen, David Donoho. Atomic decomposition by basis pursuit. *SIAM Journal of the Science of Computing*, 20:33–61, 1998.

[7] Rob Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[8] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *arXiv*, page 1608.03665, 2016.

[9] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.