Nonoverlap-Promoting Variable Selection

Pengtao Xie [1 2] **Hongbao Zhang** [1] **Yichen Zhu** [3] **Eric P. Xing** [1]

Abstract

Variable selection is a classic problem in machine learning (ML), widely used to find important explanatory factors, and improve generalization performance and interpretability of ML models. In this paper, we consider variable selection for models where multiple responses are to be predicted based on the same set of covariates. Since each response is relevant to a unique subset of covariates, we desire the selected variables for different responses have small overlap. We propose a regularizer that simultaneously encourage orthogonality and sparsity, which jointly brings in an effect of reducing overlap. We apply this regularizer to four model instances and develop efficient algorithms to solve the regularized problems. We provide a formal analysis on why the proposed regularizer can reduce generalization error. Experiments on both simulation studies and real-world datasets demonstrate the effectiveness of the proposed regularizer in selecting less-overlapped variables and improving generalization performance.

1. Introduction

Among the many criteria of evaluating model quality, two are typically considered: (1) accuracy of prediction on unseen data; (2) interpretation of the model. For (2), scientists prefer a simpler model because it puts more light on the relationship between the response and covariates. Parsimony is especially an important issue when the number of predictors is large. With a large number of predictors, we often would like to determine a smaller subset that exhibits the strongest effects.

To produce accurate prediction while selecting a subset of important factors, regularization-based variable-selection methods have been widely studied. The most notable one is ℓ_1-regularization (Tibshirani, 1996), which encourages the model coefficients to be sparse. Its variants including ℓ_1/ℓ_2-norm (Yuan & Lin, 2006) that brings in a group sparsity effect and elastic net (Zou & Hastie, 2005) which encourages strongly correlated predictors to be in or out of the model together, among many others.

In many ML problems, multiple responses are to be predicted based on the same set of covariates. For example, in multi-task classification, the classifiers of m classes are built on top of a shared feature set and each classifier has a class-specific coefficient vector. In topic modeling (Blei et al., 2003), multiple topics are learned over the same vocabulary and each topic has a unique multinomial distribution on the words. Different responses are relevant to different subsets of covariates. For example, an education topic is relevant to words like student, university, professor while a political topic is relevant to words like government, president, election, etc. To account for the difference between different responses when performing variable selection, we desire the selected variables for different responses to be less-overlapped.

The problem is formally formulated as follows. Consider m responses sharing d covariates. Each response has a specific d-dimensional weight vector $\mathbf{w}$ where each dimension corresponds to a covariate. Let $s(\mathbf{w}) = \{k|w_k \neq 0\}$ – the support of $\mathbf{w}$ – index the selected variables for a response. For any two responses i and j, we desire their selected variables $s(\mathbf{w}_i)$ and $s(\mathbf{w}_j)$ are less overlapped, where the overlapness is measured by $\frac{|s(\mathbf{w}_i)\cap s(\mathbf{w}_j)|}{|s(\mathbf{w}_i)\cup s(\mathbf{w}_j)|}$. To achieve this effect, we propose a regularizer that simultaneously encourages different weight vectors to be close to being orthogonal and each vector to be sparse, which jointly encourage vectors' supports to have small overlap. Empirically, we verify that minimizing this regularizer reduces overlap among selected variables.

The major contributions of this work include:

- We propose a new type of regularization approach which encourages a nonoverlap effect in variable selection.
- We apply the proposed regularizer to four models: multiclass logistic regression, distance metric learning, sparse

[1]Petuum Inc [2]School of Computer Science, Carnegie Mellon University [3]School of Mathematical Sciences, Peking University. Correspondence to: Pengtao Xie <pengtao.xie@petuum.com>, Eric P. Xing <eric.xing@petuum.com>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

coding, and deep neural networks.

- We derive efficient algorithms to solve these regularized problems. In particular, we develop an algorithm based on ADMM and coordinate descent for regularized sparse coding.
- We analyze why the proposed regularizer improves generalization performance.
- In experiments, we demonstrate the empirical effectiveness of this regularizer.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 presents the methods. Section 4 provides generalization error analysis and Section 5 gives experimental results. Section 6 concludes the paper.

2. Related Works

Variable selection based on regularization has been widely studied. Lasso (Tibshirani, 1996) uses ℓ_1-norm to encourage the coefficient vector of the linear regression model to be sparse. The lasso is able to recover the exact support of a sparse model from data generated by this model if the covariates are not too correlated (Zhao & Yu, 2006). Elastic net (Zou & Hastie, 2005) uses the weighted sum of the ℓ_1 and ℓ_2 norm to encourage strongly-correlated variables to be co-selected. Group lasso (Yuan & Lin, 2006) uses the ℓ_1/ℓ_2 penalty, which is defined as the sum of the ℓ_2 norms of sub-weight-vectors corresponding to predefined groups, to select groups of variables. It recovers the support of a model if the support is a union of groups and if covariates of different groups are not too correlated. Zhao et al. (2009) proposed composite absolute penalties for hierarchical selection of covariates, e.g., when one has a hierarchy over the covariates and wants to select covariates only if their ancestors in the hierarchy are also selected. Graphical lasso (Friedman et al., 2008) uses matrix ℓ_1 norm for covariance (or neighborhood) selection.

In the context of group variable selection, several works (Bach, 2009; Jacob et al., 2009; Zhao et al., 2009) consider the cases where variables from different groups are overlapping or nonoverlapping. Their problem settings are different from ours. In their problems, the (non)overlapping structure is with respect to groups and is known as a prior while in our problem it is with respect to different responses and is unknown.

3. Methods

In this section, we propose a nonoverlap-promoting regularizer and apply it to four ML models.

3.1. Nonoverlap-Promoting Regularization

We assume the model has m responses and each is parameterized by a weight vector. For a vector $\mathbf{w}$, its *support* $s(\mathbf{w})$ is defined as $\{i|w_i \neq 0\}$ – the indices of nonzero entries in $\mathbf{w}$. And the support contains indexes of the selected variables. We first define a score $\tilde{o}(\mathbf{w}_i, \mathbf{w}_j)$ to measure the overlap between selected variables of two responses:

$$\tilde{o}(\mathbf{w}_i, \mathbf{w}_j) = \frac{|s(\mathbf{w}_i) \cap s(\mathbf{w}_j)|}{|s(\mathbf{w}_i) \cup s(\mathbf{w}_j)|}, \tag{1}$$

which is the Jaccard index of the supports. The smaller $\tilde{o}(\mathbf{w}_i, \mathbf{w}_j)$ is, the less overlapped the two sets of selected variables are. For m variable sets, the overlap score is defined as the sum of pairwise scores

$$o(\{\mathbf{w}_i\}_{i=1}^m) = \frac{1}{m(m-1)} \sum_{i \neq j}^{m} \tilde{o}(\mathbf{w}_i, \mathbf{w}_j). \tag{2}$$

This score function is not smooth, which will result in great difficulty for optimization if used as a regularizer. Instead, we propose a smooth function that is motivated from $\tilde{o}(\mathbf{w}_i, \mathbf{w}_j)$ and can achieve a similar effect as $o(\mathcal{W})$. The basic idea is: to encourage small overlap, we can encourage (1) each vector has a small number of non-zero entries and (2) the intersection of supports among vectors is small. To realize (1), we use an L1 regularizer to encourage the vectors to be sparse. To realize (2), we encourage the vectors to be close to being orthogonal. For two sparse vectors, if they are close to orthogonal, then their supports are landed on different positions. As a result, the intersection of supports is small.

We follow the method proposed by (Xie et al., 2017b) to promote orthogonality. To encourage two vectors $\mathbf{w}_i$ and $\mathbf{w}_j$ to be close to being orthogonal, one can make their ℓ_2 norm $\|\mathbf{w}_i\|_2$, $\|\mathbf{w}_j\|_2$ close to one and their inner product $\mathbf{w}_i^\top \mathbf{w}_j$ close to zero. Based on this, one can promote orthogonality among a set of vectors by encouraging the Gram matrix $\mathbf{G}$ ($G_{ij} = \mathbf{w}_i^\top \mathbf{w}_j$) of these vectors to be close to an identity matrix $\mathbf{I}$. Since $\mathbf{w}_i^\top \mathbf{w}_j$ and zero are off the diagonal of $\mathbf{G}$ and $\mathbf{I}$ respectively, and $\|\mathbf{w}_i\|_2^2$ and one are on the diagonal of $\mathbf{G}$ and $\mathbf{I}$ respectively, encouraging $\mathbf{G}$ close to $\mathbf{I}$ essentially makes $\mathbf{w}_i^\top \mathbf{w}_j$ close to zero and $\|\mathbf{w}_i\|_2$ close to one. As a result, $\mathbf{w}_i$ and $\mathbf{w}_j$ are encouraged to be close to being orthogonal. In (Xie et al., 2017b), one way proposed to measure the "closeness" between two matrices is to use the log-determinant divergence (LDD) (Kulis et al., 2009). The LDD between two $m \times m$ positive definite matrices $\mathbf{X}$ and $\mathbf{Y}$ is defined as $D(\mathbf{X}, \mathbf{Y}) = \text{tr}(\mathbf{X}\mathbf{Y}^{-1}) - \log\det(\mathbf{X}\mathbf{Y}^{-1}) - m$ where $\text{tr}(\cdot)$ denotes matrix trace. The closeness between $\mathbf{G}$ and $\mathbf{I}$ can be achieved by encouraging their LDD $D(\mathbf{G}, \mathbf{I}) = \text{tr}(\mathbf{G}) - \log\det(\mathbf{G}) - m$ to be small.

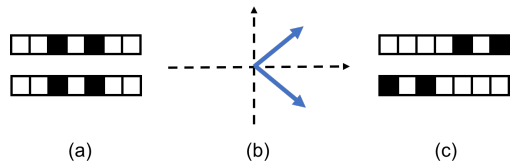

Figure 1. (a) Under L1 regularization, the vectors are sparse, but their supports are overlapped; (b) Under LDD regularization, the vectors are orthogonal, but their supports are overlapped; (c) Under LDD-L1 regularization, the vectors are sparse and mutually orthogonal and their supports are not overlapped.

Combining the orthogonality-promoting LDD regularizer with the sparsity-promoting L1 regularizer together, we obtain the following LDD-L1 regularizer

$$\Omega(\mathcal{W}) = \text{tr}(\mathbf{G}) - \log\det(\mathbf{G}) + \gamma \sum_{i=1}^{m} |\mathbf{w}_i|_1, \quad (3)$$

where γ is a tradeoff parameter between these two regularizers. As verified in experiments, this regularizer can effectively promote nonoverlap. The formal analysis of the relationship between Eq.(3) and Eq.(2) will be left for future study. It is worth noting that either L1 or LDD alone is not sufficient to reduce overlap. As illustrated in Figure 1(a) where only L1 is applied, though the two vectors are sparse, their supports are completely overlapped. In Figure 1(b) where the LDD regularizer is applied, though the two vectors are very close to orthogonal, their supports are completely overlapped since they are dense. In Figure 1(c) where the LDD-L1 regularizer is used, the two vectors are sparse and are close to being orthogonal. As a result, their supports are not overlapped.

3.2. Case Studies

We apply the LDD-L1 regularizer to four ML models.

Multiclass Logistic Regression (MLR) aims at classifying a data example $\mathbf{x} \in \mathbb{R}^d$ (whose features are treated as covariates) into one of m classes (treated as responses). It is parameterized by an coefficient matrix $\mathbf{W} \in \mathbb{R}^{d\times m}$ where the i-th column is the coefficient vector of class i and d is the feature dimension of $\mathbf{x}$. In inference, MLR calculates $\mathbf{p} = \text{softmax}(\mathbf{W}^\top\mathbf{x}+\mathbf{b}) \in \mathbb{R}^m$ where p_i denotes the probability that $\mathbf{x}$ belongs to class i and $\mathbf{b} \in \mathbb{R}^m$ is a bias vector. $\mathbf{x}$ is assigned to the class yielding the largest probability. Given N training examples $\{\mathbf{x}_n, y_n\}_{n=1}^N$, MLR learns $\mathbf{W}$ by minimizing the cross-entropy loss between $\mathbf{p}_n$ and the ground-truth class label y_n. The LDD-L1 regularizer can be applied to encourage the coefficient vectors of different classes to have less-overlapped supports.

Distance Metric Learning (DML) has wide applications in classification, clustering and information retrieval (Xing et al., 2002; Davis et al., 2007; Guillaumin et al., 2009). Given data pairs labeled as similar or dissimilar, DML aims at learning a distance metric such that similar pairs would be placed close to each other and dissimilar pairs are separated apart. Following (Weinberger et al., 2005), we define the distance metric between $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ as $\|\mathbf{W}^\top\mathbf{x} - \mathbf{W}^\top\mathbf{y}\|_2^2$ where $\mathbf{W} \in \mathbb{R}^{d\times m}$ contain m projection vectors which are treated as responses. The features in a data example are treated as covariates. Given N training examples, $\{\mathbf{x}_n, \mathbf{y}_n, t_n\}_{n=1}^N$, where $\mathbf{x}_n$ and $\mathbf{y}_n$ are similar if the label t_n equals to 1 and dissimilar if $t_n = 0$, following (Guillaumin et al., 2009), we learn the distance metric by minimizing $\sum_{n=1}^N \log(1 + \exp((2t_n - 1)\|\mathbf{W}^\top\mathbf{x} - \mathbf{W}^\top\mathbf{y}\|_2^2))$. Using LDD-L1 to promote nonoverlap among the projection vectors in $\mathbf{W}$, we obtain the LDD-L1 regularized DML problem:

$$\min_{\mathcal{F}} \sum_{n=1}^{N} \log(1 + \exp((2t_n - 1)\|\mathbf{W}^\top(\mathbf{x} - \mathbf{y})\|_2^2)) + \lambda\Omega(\mathbf{W}). \quad (4)$$

Sparse Coding (SC) Given n data samples $\mathbf{X} \in \mathbb{R}^{d\times n}$ where d is the number of features (treated as covariates), SC (Olshausen & Field, 1997) aims at using a dictionary of basis vectors (treated as responses) $\mathbf{W} \in \mathbb{R}^{d\times m}$ to reconstruct $\mathbf{X}$, where m is the number of basis vectors. Each data sample $\mathbf{x}$ is reconstructed by taking a sparse linear combination of the basis vectors $\mathbf{x} \approx \sum_{j=1}^m \alpha_j\mathbf{w}_j$, where $\{\alpha_j\}_{j=1}^m$ are the linear coefficients and most of them are zero. The reconstruction error is measured using the squared L2 norm $\|\mathbf{x} - \sum_{j=1}^m \alpha_j\mathbf{w}_j\|_2^2$. To achieve sparsity among the codes, L1 regularization is utilized: $\sum_{j=1}^m |\alpha_j|_1$. To avoid the degenerated case where most coefficients are zero and the basis vectors are of large magnitude, L2 regularization is applied to the basis vectors: $\|\mathbf{w}_j\|_2^2$. We apply the LDD-L1 regularizer to encourage the supports of the basis vectors to have small overlap. Putting these pieces together, we obtain the LDD-L1 regularized SC (LDD-L1-SC) problem

$$\min_{\mathbf{W},\mathbf{A}} \frac{1}{2}\|\mathbf{X} - \mathbf{W}\mathbf{A}\|_F^2 + \lambda_1|\mathbf{A}|_1 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda_3}{2}(\text{tr}(\mathbf{W}^\top\mathbf{W}) - \log\det(\mathbf{W}^\top\mathbf{W})) + \lambda_4|\mathbf{W}|_1 \quad (5)$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ denotes all the linear coefficients.

Deep Neural Networks (DNNs) In a DNN with L hidden layers, each hidden layer l is equipped with $m^{(l)}$ units and each unit i is connected with all units in layer $l-1$. We treat units in layer l as a group of *responses* and units in the layer $l-1$ as the corresponding *covariates*. Hidden unit i at layer l is parameterized by a weight vector $\mathbf{w}_i^{(l)}$. For the

$m^{(l)}$ weight vectors $\mathcal{W}^{(l)} = \{\mathbf{w}_i^{(l)}\}_{i=1}^{m^{(l)}}$ in each layer l, we apply the LDD-L1 regularizer to encourage them to have less-overlapped supports. An LDD-L1 regularized DNN problem can be defined in the following way:

$$\min_{\{\mathcal{W}^{(l)}\}_{l=1}^{L}} \quad \mathcal{L}(\{\mathcal{W}^{(l)}\}_{l=1}^{L}) + \lambda \sum_{l=1}^{L} \Omega(\mathcal{W}^{(l)})$$

where $\mathcal{L}(\{\mathcal{W}^{(l)}\}_{l=1}^{L})$ is the objective function of this DNN. In the experiments, we study two popular instances of DNNs: long short-term memory (LSTM) network (Hochreiter & Schmidhuber, 1997) and convolutional neural network (CNN).

3.3. Algorithm

For LDD-L1-regularized MLR, NN and DML problems, we solve them using proximal gradient descent (Parikh & Boyd, 2014). The proximal operation is with respect to the L1 regularizer in LDD-L1. The algorithm iteratively performs the following three steps until convergence: (1) calculate gradient of $\mathcal{L}(\mathbf{W}) + \lambda(\text{tr}(\mathbf{W}^\top\mathbf{W}) - \text{logdet}(\mathbf{W}^\top\mathbf{W}))$ where $\mathcal{L}(\mathbf{W})$ is the loss function of the unregularized ML model and $\text{tr}(\mathbf{W}^\top\mathbf{W}) - \text{logdet}(\mathbf{W}^\top\mathbf{W})$ is the LDD regularizer in LDD-L1; (2) perform gradient descent update of $\mathbf{W}$; (3) apply the proximal operator of the L1 regularizer to $\mathbf{W}$.

For LDD-L1-SC, we solve it by alternating between $\mathbf{A}$ and $\mathbf{W}$: (1) updating $\mathbf{A}$ with $\mathbf{W}$ fixed; (2) updating $\mathbf{W}$ with $\mathbf{A}$ fixed. These two steps alternate until convergence. With $\mathbf{W}$ fixed, the sub-problem defined over $\mathbf{A}$ is $\min_{\mathbf{A}} \ \frac{1}{2}\|\mathbf{X} - \mathbf{W}\mathbf{A}\|_F^2 + \lambda_1|\mathbf{A}|_1$, which can be decomposed into n Lasso problems (**P1**): for $i = 1, \cdots, n$, $\min_{\mathbf{a}_i} \ \frac{1}{2}\|\mathbf{x}_i - \mathbf{W}\mathbf{a}_i\|_2^2 + \lambda_1|\mathbf{a}_i|_1$ where $\mathbf{a}_i$ is the coefficient vector of the i-th sample. Lasso can be solved by many algorithms, such as proximal gradient descent (PGD). Fixing $\mathbf{A}$, the sub-problem defined over $\mathbf{W}$ is (**P2**): $\min_{\mathbf{W}} \ \frac{1}{2}\|\mathbf{X} - \mathbf{W}\mathbf{A}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 + \lambda_4|\mathbf{W}|_1 + \frac{\lambda_3}{2}(\text{tr}(\mathbf{W}^\top\mathbf{W}) - \log\det(\mathbf{W}^\top\mathbf{W}))$. We solve this problem using an ADMM-based algorithm. First, we write the problem into an equivalent form: $\min_{\mathbf{W}} \frac{1}{2}\|\mathbf{X} - \mathbf{W}\mathbf{A}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 + \lambda_4|\widetilde{\mathbf{W}}|_1 + \frac{\lambda_3}{2}(\text{tr}(\mathbf{W}^\top\mathbf{W}) - \log\det(\mathbf{W}^\top\mathbf{W}))$, subject to $\mathbf{W} = \widetilde{\mathbf{W}}$. Then we write down the augmented Lagrangian function (**P3**): $\frac{1}{2}\|\mathbf{X} - \mathbf{W}\mathbf{A}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 + \lambda_4|\widetilde{\mathbf{W}}|_1 + \langle\mathbf{U}, \mathbf{W} - \widetilde{\mathbf{W}}\rangle + \frac{\rho}{2}\|\mathbf{W} - \widetilde{\mathbf{W}}\|_F^2 + \frac{\lambda_3}{2}(\text{tr}(\mathbf{W}^\top\mathbf{W}) - \log\det(\mathbf{W}^\top\mathbf{W}))$. We minimize this Lagrangian function by alternating among $\mathbf{W}$, $\widetilde{\mathbf{W}}$, and $\mathbf{U}$.

Update $\mathbf{W}$ The subproblem defined on $\mathbf{W}$ is (**P4**): $\min_{\mathbf{W}} \ \frac{1}{2}\|\mathbf{X} - \mathbf{W}\mathbf{A}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{W}\|_F^2 + \langle\mathbf{U}, \mathbf{W}\rangle + \frac{\lambda_3}{2}(\text{tr}(\mathbf{W}^\top\mathbf{W}) - \log\det(\mathbf{W}^\top\mathbf{W})) + \frac{\rho}{2}\|\mathbf{W} - \widetilde{\mathbf{W}}\|_F^2$, which can be solved using a coordinate descent (CD) algorithm. In each iteration of CD, one basis vector is chosen for update while the others are fixed. Without loss of generality, we assume it is $\mathbf{w}_1$. The loss function defined over $\mathbf{w}_1$ is $\frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x}_i - \sum_{l=2}^{m} a_{il}\mathbf{w}_l - a_{i1}\mathbf{w}_1\|_2^2 + \frac{\lambda_2+\lambda_3}{2}\|\mathbf{w}_1\|_2^2 - \frac{\lambda_3}{2}\text{logdet}(\mathbf{W}^\top\mathbf{W}) + \mathbf{u}^\top\mathbf{w}_1 + \frac{\rho}{2}\|\mathbf{w}_1 - \widetilde{\mathbf{w}}_1\|_2^2$. The optimal solution can be obtained via the following procedures: (1) calculate $\mathbf{M} = \mathbf{I} - \mathbf{W}_{\neg 1}(\mathbf{W}_{\neg 1}^\top\mathbf{W}_{\neg 1})^{-1}\mathbf{W}_{\neg 1}^\top$, where $\mathbf{W}_{\neg 1} = [\mathbf{w}_2, \cdots, \mathbf{w}_m]$; (2) perform eigen-decomposition: $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$; (3) solve the scalar quadratic equation $\gamma\sum_{s=m}^{d}(\mathbf{U}^\top\mathbf{b})_s^2 = (\gamma c - \lambda_3)^2$ w.r.t γ, where $c = \sum_{i=1}^{n} a_{i1}^2 + \lambda_2 + \lambda_3 + \rho$ and $\mathbf{b} = \sum_{i=1}^{n} a_{i1}(\mathbf{x}_i - \sum_{l=2}^{m} a_{il}\mathbf{w}_l) - \mathbf{u} + \rho\widetilde{\mathbf{w}}_j$; (4) calculate $\mathbf{w}_1$ as:

$$\mathbf{w}_1 = \gamma\mathbf{U}(\gamma c\mathbf{I} - \lambda_3\mathbf{\Sigma})^{-1}\mathbf{U}^\top\mathbf{b}. \tag{6}$$

The detailed derivation is deferred to the supplements.

Update $\widetilde{\mathbf{W}}$ The subproblem defined on $\widetilde{\mathbf{W}}$ is (**P5**) : $\min_{\widetilde{\mathbf{W}}} \ \lambda_4|\widetilde{\mathbf{W}}|_1 - \langle\mathbf{U}, \widetilde{\mathbf{W}}\rangle + \frac{\rho}{2}\|\mathbf{W} - \widetilde{\mathbf{W}}\|_F^2$, which is a Lasso problem and can be solved using PGD.

Update $\mathbf{U}$ The update equation of $\mathbf{U}$ is simple: $\mathbf{U} = \mathbf{U} + (\mathbf{W} - \widetilde{\mathbf{W}})$.

Algorithm 1 Algorithm for solving the LDD-L1-SC problem

Initialize $\mathbf{W}$ and $\mathbf{A}$
repeat
  Update $\mathbf{A}$ with $\mathbf{W}$ being fixed, by solving n Lasso problems (**P1**).
  repeat
    repeat
      for $i \leftarrow 1$ to m **do**
        Update the ith column vector $\mathbf{w}_i$ of $\mathbf{W}$ using Eq.(6)
      end for
    until convergence of the problem (**P4**)
    Update $\widetilde{\mathbf{W}}$ by solving the Lasso problem (**P5**)
    $\mathbf{U} \leftarrow \mathbf{U} + (\mathbf{W} - \widetilde{\mathbf{W}})$
  until convergence of the problem (**P3**)
until convergence of the problem defined in Eq.(5)

4. Generalization Error Analysis

In this section, we analyze how the LDD-L1 regularizer affects the generalization error of ML models. We use the distance metric learning (DML) model to perform the study. In DML, the hypothesis function is $u(\mathbf{x}, \mathbf{y}) = \|\mathbf{W}^\top(\mathbf{x} - \mathbf{y})\|_2^2$ and the loss function ℓ is the logistic loss $\ell(u(\mathbf{x}, \mathbf{y}), t) = \log(1 + \exp((2t - 1)u(\mathbf{x}, \mathbf{y})))$. Let $\mathcal{U} = \{u : (\mathbf{x}, \mathbf{y}) \mapsto \|\mathbf{W}^\top(\mathbf{x} - \mathbf{y})\|_2^2, \Omega(\mathbf{W}) \leq \tau\}$ denote the hypothesis set and $\mathcal{A} = \{\ell : (\mathbf{x}, \mathbf{y}, t) \mapsto \ell(u(\mathbf{x}, \mathbf{y}), t), u \in \mathcal{U}\}$ denote the loss class, which is the composition of the loss function with each of the hypothe-

ses. In $\mathcal{U}$, we add the constraint $\Omega(\mathbf{W}) \leq \tau$ to incorporate the impact of the LDD-L1 regularizers $\Omega(\mathbf{W})$. τ controls the strength of regularization. A smaller τ entails stronger promotion of nonoverlap. τ is controlled by the regularization parameter λ in Eq.(4). Increasing λ reduces τ. Given the joint distribution p^* of input data pair $(\mathbf{x}, \mathbf{y})$ and the binary label t indicating whether this data pair is similar or dissimilar, the risk of the hypothesis u is $L(u) = \mathbb{E}_{(\mathbf{x},\mathbf{y},t)\sim p^*}[\ell(u(\mathbf{x}, \mathbf{y}), t)]$. Its empirical counterpart (training error) can be defined as $\widehat{L}(u) = \frac{1}{N}\sum_{n=1}^{N} \ell(u(\mathbf{x}_n, \mathbf{y}_n), t_n)$. The generalization error of a hypothesis u is defined as $L(u) - \widehat{L}(u)$, which represents how well the algorithm can learn and usually depends on the complexity of the hypothesis class and the number of training examples.

To facilitate the analysis, we define a *capacity variable* on $\mathbf{W}$. We first prove that the generalization error can be upper bounded by an increasing function of the capacity variable. Then we show that the capacity variable can be upper bounded by an increasing function of the LDD regularizer. Combining these two steps we reveal the relationship between the generalization error and the LDD-L1 regularizer.

Definition 1 (Capacity Variable) *Let $\pi_1, \cdots, \pi_m$ be the eigenvalues of $\mathbf{W}^\top\mathbf{W}$. Then the capacity variable is defined as:*

$$\mathcal{C}(\mathbf{W}) = \sum_{j=1}^{m} |\pi_j - 1|.$$

The following inequality helps us to understand the intuitive meaning of $\mathcal{C}(\mathbf{W})$:

$$\frac{1}{m}\|\mathbf{W}^\top\mathbf{W} - \mathbf{I}_m\|_1 \leq \mathcal{C}(\mathbf{W}). \tag{7}$$

$\frac{1}{m}\|\mathbf{W}^\top\mathbf{W} - \mathbf{I}_m\|_1$ measures the closeness between the Gram matrix $\mathbf{W}^\top\mathbf{W}$ and an identity matrix using L1 norm. The smaller this quantity is, the closer to being orthogonal the vectors in $\mathbf{W}$ are. Being an upper bound of this quantity, $\mathcal{C}(\mathbf{W})$ essentially determines the level of near-orthogonality among vectors.

$\Omega_{ldd}(\mathbf{W}) = \Omega(\mathbf{W}) - \gamma\|\mathbf{W}\|_1 \leq \Omega(\mathbf{W}) \leq \tau$. Define $\mathcal{W} = \{\mathbf{W} \in \mathbb{R}^{d\times m} | \Omega(\mathbf{W}) \leq \tau]\}$ and $\widetilde{\mathcal{C}}(\mathcal{W}) = \sup_{\mathbf{W}\in\mathcal{W}} \mathcal{C}(\mathbf{W})$. The following lemma shows that the generalization error can be upper bounded using $\mathcal{C}(\mathcal{W})$.

Lemma 1 *Suppose $\sup_{(\mathbf{x},\mathbf{y})}\|\mathbf{x} - \mathbf{y}\|_2 \leq B_0$, then with probability at least $1 - \delta$, we have*

$$\begin{aligned} L(u) - \hat{L}(u) &\leq 4B_0^2 \sup_{\mathbf{W}'\in\mathcal{W}} \|\mathbf{W}'\|_1 \sqrt{\tfrac{(\widetilde{\mathcal{C}}(\mathcal{W})+1)m}{N}} \\ &+[B_0^2(\widetilde{\mathcal{C}}(\mathcal{W}) + m) + 1]\sqrt{\tfrac{2\log(1/\delta)}{N}}. \end{aligned} \tag{8}$$

The upper bound is an increasing function of $\widetilde{\mathcal{C}}(\mathcal{W})$. The next lemma shows that $\mathcal{C}(\mathbf{W})$ can be upper bounded by an increasing function of the LDD regularizer $\Omega_{ldd}(\mathbf{W}) = \mathrm{tr}(\mathbf{W}^\top\mathbf{W}) - \log\det(\mathbf{W}^\top\mathbf{W}) - m$.

Lemma 2 *Let $g(x) = x - \log(x + 1)$. Then we have*

$$\mathcal{C}(\mathbf{W}) \leq g^{-1}(\Omega_{ldd}(\mathbf{W})/m)m.$$

where $g^{-1}(\cdot)$ is the inverse function of g on $[0, \infty)$ and is an increasing function.

Since $\Omega_{ldd}(\mathbf{W}) \leq \tau$, we have $\mathcal{C}(\mathbf{W}) \leq g^{-1}(\tau/m)m$ for any $\mathbf{W} \in \mathcal{W}$, i.e., $\widetilde{\mathcal{C}}(\mathcal{W}) \leq g^{-1}(\tau/m)m$. Similarly, $\|\mathbf{W}\|_1 = (\Omega(\mathbf{W}) - \Omega_{ldd}(\mathbf{W}))/\gamma \leq \tau/\gamma$. Substituting these two inequalities into Lemma 1, we obtain the following theorem where the generalization error is upper bounded based on τ.

Theorem 1 *Suppose $\sup_{(\mathbf{x},\mathbf{y})}\|\mathbf{x} - \mathbf{y}\|_2 \leq B_0$. With probability at least $1 - \delta$, we have*

$$\begin{aligned} L(u) - \hat{L}(u) &\leq \tfrac{4B_0^2\tau}{\gamma}\sqrt{\tfrac{m(g^{-1}(\tau/m)m+1)}{N}} \\ &+[B_0^2 m(g^{-1}(\tau/m) + 1) + 1]\sqrt{\tfrac{2\log(1/\delta)}{N}}. \end{aligned} \tag{9}$$

From this generalization error bound (GEB), we can see two major implications. First, LDD-L1 can effectively control the GEB. Increasing the strength of LDD-L1 regularization (by enlarging λ) reduces τ, which decreases the GEB since it is an increasing function of τ. Second, the GEB converges with rate $O(1/\sqrt{N})$, where N is the number of training data pairs. This rate matches with that in (Bellet & Habrard, 2015; Verma & Branson, 2015).

5. Experiments

5.1. Simulation Study

The simulation study is performed on the multiclass logistic regression model. We set the number of classes to 10. Each class is relevant to 5 variables. The variables of different classes have no overlap. We generate 1000 data samples from a multivariate Gaussian distribution with zero mean and the covariance matrix is set to an identity matrix. In the coefficient vector of each class, the entries corresponding to the relevant variables are uniformly sampled from $[-1, 1]$ and the rest entries are set to zero. Given a generated sample $\mathbf{x}$ and the generated coefficient matrix $\mathbf{W} \in \mathbb{R}^{10\times 50}$, the class label of sample $\mathbf{x}$ is determined as $y = \mathrm{argmax}_k\ [\mathbf{W}\mathbf{x} + \mathbf{b}]_k$, where $\mathbf{b} \in \mathbb{R}^{10}$ is a randomly generated bias vector whose entries are sampled independently from a univariate normal distribution. We split the dataset into train/validation/test set with 600/200/200 examples respectively. The regularization parameter is tuned

	Sensitivity	Specificity	Error rate
L1	0.76	0.71	0.31
Elastic Net	0.74	0.72	0.30
LDD-L1	0.82	0.75	0.24

Table 1. Sensitivity and specificity for support recovery and error rate for prediction.

on the validation set to achieve the best prediction performance. We generate 50 simulated datasets. The performance is averaged over these 50 datasets. We compare our method with L1-regularization (Tibshirani, 1996) and elastic net (Zou & Hastie, 2005). We did not compare with LDD since it is not able to select variables.

Following (Kim & Xing, 2012), we use sensitivity (true positive rate) and specificity (true negative rate) to measure the performance of recovering the true supports of the coefficient vectors, shown in the second and third column of Table 1. Our method outperforms the baselines with a large margin. LDD-L1 encourages the supports of different weight vectors to have less overlap, which makes it more suitable to select nonoverlapping variables. We also compare the performance of different methods in terms of prediction errors, shown in the fourth column of Table 1. LDD-L1 achieves the lowest error rate. Since the variables selected by our method are closer to the ground-truth, the predictions made by our method based upon these selected variables are more accurate.

5.2. Experiments on Real Data

We apply the LDD-L1 to 3 ML models and 4 datasets and verify whether it is able to improve generalization performance. In each experiment, the hyperparameters were tuned on the validation set.

Sparse Coding for Text Representation Learning The SC experiments were conducted on two text datasets: 20-Newsgroups[1] (20-News) and Reuters Corpus[2] Volume 1 (RCV1). The 20-News dataset contains newsgroup documents belonging to 20 categories, where 11314, 3766 and 3766 documents were used for training, validation and testing respectively. The original RCV1 dataset contains documents belonging to 103 categories. Following (Cai & He, 2012), we chose the largest 4 categories which contain 9625 documents, to carry out the study. The number of training, validation and testing documents are 5775, 1925, 1925 respectively. For both datasets, stopwords were removed and all words were changed into lower-case. Top 1000 words with the highest document frequency were selected to form the vocabulary. We used tf-idf to represent documents and the feature vector of each document is normalized to have unit L2 norm. For 20-News, the number of basis vectors in LDD-L1-SC is set to 50. λ_1, λ_2, λ_3 and λ_4 are set to 1, 1, 0.1 and 0.001 respectively. For RCV1, the number of basis vectors is set to 200. λ_1, λ_2, λ_3 and λ_4 are set to 0.01, 1, 1 and 1 respectively. We compared LDD-L1 with LDD-only and L1-only.

[1] http://qwone.com/~jason/20Newsgroups/

[2] http://www.daviddlewis.com/resources/testcollections/rcv1/

Method	20-News		RCV1	
	Test	Gap	Test	Gap
SC	0.592	0.119	0.872	0.009
LDD-SC	0.605	0.108	0.886	0.005
L1-SC	0.606	0.105	0.897	0.005
LDD-L1-SC	**0.612**	0.099	**0.909**	-0.015

Table 2. Classification accuracy on the test sets of 20-News and RCV1, and the gap between training and test accuracy.

To evaluate the model performance quantitatively, we applied the dictionary learned on the training data to infer the linear coefficients ($\mathbf{A}$ in Eq.5) of test documents, then performed k-nearest neighbors (KNN) classification on $\mathbf{A}$. Table 2 shows the classification accuracy on test sets of 20-News and RCV1 and the gap[3] between the accuracy on training and test sets. Without regularization, SC achieves a test accuracy of 0.592 on 20-News, which is lower than the training accuracy by 0.119. This suggests that an overfitting to training data occurs. With LDD-L1 regularization, the test accuracy is improved to 0.612 and the gap between training and test accuracy is reduced to 0.099, demonstrating the ability of LDD-L1 in alleviating overfitting. Though LDD alone and L1 alone improve test accuracy and reduce train/test gap, they perform less well than LDD-L1, which indicates that for overfitting reduction, encouraging nonoverlap is more effective than solely promoting orthogonality or solely promoting sparsity. Similar observations are made on the RCV1 dataset. Interestingly, the test accuracy achieved by LDD-L1-SC on RCV1 is better than the training accuracy.

Table 3 shows the selected variables (words that have nonzero weights) for 9 exemplar basis vectors learned by LDD-L1-SC on the 20-News dataset. From the selected words, we can see basis vector 1-9 represent the following semantics respectively: crime, faith, job, war, university, research, service, religion and Jews. The selected words of different basis vectors have no overlap. As a result, it is easy to associate each vector with a unique concept, in other words, easy to interpret. Figure 2 visualizes the learned vectors where the black dots denote vectors' supports. As can be seen, the supports of different basis vectors are landed over different words and their overlap is small.

LSTM for Language Modeling We apply LSTM networks (Hochreiter & Schmidhuber, 1997) to learn language models on the Penn Treebank (PTB) dataset (Marcus et al.,

[3] Training accuracy minus test accuracy.

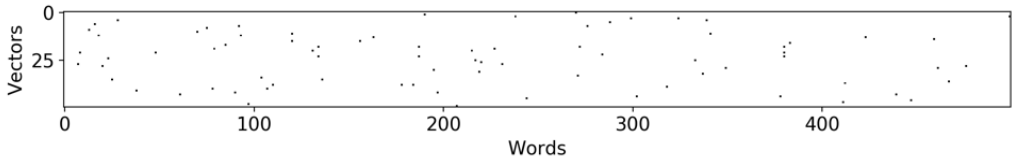

Figure 2. Visualization of basis vectors

Basis Vector	Selected Words
1	crime, guns
2	faith, trust
3	worked, manager
4	weapons, citizens
5	board, uiuc
6	application, performance, ideas
7	service, quality
8	bible, moral
9	christ, jews, land, faq

Table 3. Selected words of 9 exemplar basis vectors

1993), which consists of 923K training, 73K validation, and 82K test words. Following (Mikolov et al.), top 10K words with highest frequency were selected to form the vocabulary. All other words are replaced with a special token UNK. The LSTM network architecture follows the word language model (PytorchTM) provided in Pytorch[4]. The number of hidden layers is set to 2. The embedding size is 1500. The size of hidden state is 1500. Following (Press & Wolf, 2016), the word embedding and softmax weights are tied. The number of training epochs is 40. Dropout with 0.65 is used. The initial learning rate is 20. Gradient clipping threshold is 0.25. The size of mini-batch is 20. In LSTM training, the network is unrolled for 35 iterations. Perplexity is used for evaluating language modeling performance (lower is better). The weight parameters are initialized uniformly between [-0.1, 0.1]. The bias parameters are initialized as 0. We compare with the following regularizers: (1) L1 regularizer; (2) orthogonality-promoting regularizers based on cosine similarity (CS) (Yu et al., 2011), incoherence (IC) (Bao et al., 2013), mutual angle (MA) (Xie et al., 2015), decorrelation (DC) (Cogswell et al., 2015), angular constraint (AC) (Xie et al., 2017a) and LDD (Xie et al., 2017b).

Table 4 shows the perplexity on the PTB test set. Without regularization, PytorchLM achieves a perplexity of 72.3. With LDD-L1 regularization, the perplexity is significantly reduced to 71.1. This shows that LDD-L1 can effectively improve generalization performance. Compared with the sparsity-promoting L1 regularizer and orthogonality-promoting regularizers, LDD-L1 – which promotes nonoverlap by simultaneously promoting sparsity and orthogonality – achieves lower perplexity. For the convenience of readers, we also list the perplexity achieved by other state of the art deep learning models. The LDD-L1 regularizer can be applied to these models as well to potentially boost their performance.

Network	Test
RNN (Mikolov & Zweig, 2012)	124.7
RNN+LDA (Mikolov & Zweig, 2012)	113.7
Deep RNN (Pascanu et al., 2013)	107.5
Sum-Product Network (Cheng et al., 2014)	100.0
RNN+LDA+KN-5+Cache (Mikolov & Zweig, 2012)	92.0
LSTM (medium) (Zaremba et al., 2014)	82.7
CharCNN (Kim et al., 2016)	78.9
LSTM (large) (Zaremba et al., 2014)	78.4
Variational LSTM (Gal & Ghahramani, 2016)	73.4
PytorchLM	72.3
CS-PytorchLM (Yu et al., 2011)	71.8
IC-PytorchLM (Bao et al., 2013)	71.9
MA-PytorchLM (Xie et al., 2015)	72.0
DC-PytorchLM (Cogswell et al., 2015)	72.2
AC-PytorchLM (Xie et al., 2017a)	71.5
LDD-PytorchLM (Xie et al., 2017b)	71.6
L1-PytorchLM	71.8
LDD-L1-PytorchLM	71.1
Pointer Sentinel LSTM (Merity et al., 2016)	70.9
Ensemble of 38 Large LSTMs (Zaremba et al., 2014)	68.7
Variat. LSTM Ensem. (Gal & Ghahramani, 2016)	68.7
Variational RHN (Zilly et al., 2016)	68.5
Variational LSTM + REAL (Inan et al., 2016)	68.5
Neural Architecture Search (Zoph & Le, 2016)	67.9
Variational RHN + RE (Inan et al., 2016)	66.0
Variational RHN + WT (Zilly et al., 2016)	65.4
Variational RHN+WT+Dropout (Zilly et al., 2016)	64.4
Architecture Search + WT V1 (Zoph & Le, 2016)	64.0
Architecture Search + WT V2 (Zoph & Le, 2016)	62.4

Table 4. Word-level perplexities on PTB test set

CNN for Image Classification The CNN experiments were performed on the CIFAR-10 dataset[5]. It consists of 32x32 color images belonging to 10 categories, where 50,000 images were used for training and 10,000 for testing. 5000 training images were used as the validation set for hyperparameter tuning. We augmented the dataset by first zero-padding the images with 4 pixels on each side, then randomly cropping the padded images to reproduce 32x32 images. The CNN architecture follows that of the wide residual network (WideResNet) (Zagoruyko, 2016).

[4] https://github.com/pytorch/examples/tree/master/word_language_model

[5] https://www.cs.toronto.edu/~kriz/cifar.html

Network	Error
Maxout (Goodfellow et al., 2013)	9.38
NiN (Lin et al., 2013)	8.81
DSN (Lee et al., 2015)	7.97
Highway Network (Srivastava et al., 2015)	7.60
All-CNN (Springenberg et al., 2014)	7.25
ResNet (He et al., 2016)	6.61
ELU-Network (Clevert et al., 2015)	6.55
LSUV (Mishkin & Matas, 2015)	5.84
Fract. Max-Pooling (Graham, 2014)	4.50
WideResNet (Huang et al., 2016)	3.89
CS-WideResNet (Yu et al., 2011)	3.81
IC-WideResNet (Bao et al., 2013)	3.85
MA-WideResNet (Xie et al., 2015)	3.68
DC-WideResNet (Cogswell et al., 2015)	3.77
LCD-WideResNet (Rodríguez et al., 2016)	3.69
AC-WideResNet (Xie et al., 2017a)	3.63
LDD-WideResNet (Xie et al., 2017b)	3.65
L1-WideResNet	3.81
LDD-L1-WideResNet	3.60
ResNeXt (Xie et al., 2016)	3.58
PyramidNet (Huang et al., 2016)	3.48
DenseNet (Huang et al., 2016)	3.46
PyramidSepDrop (Yamada et al., 2016)	3.31

Table 5. Classification error (%) on CIFAR-10 test set

The depth and width are set to 28 and 10 respectively. The networks are trained using SGD, where the epoch number is 200, the learning rate is set to 0.1 initially and is dropped by 0.2 at 60, 120 and 160 epochs, the minibatch size is 128 and the Nesterov momentum is 0.9. The dropout probability is 0.3 and the L2 weight decay is 0.0005. Model performance is measured using error rate, which is the median of 5 runs. We compared with (1) L1 regularizer; (2) orthogonality-promoting regularizers including CS, IC, MA, DC, AC, LDD and one based on locally constrained decorrelation (LCD) (Rodríguez et al., 2016).

Table 5 shows classification errors on CIFAR-10 test set. Compared with the unregularized WideResNet which achieves an error rate of 3.89%, the proposed LDD-L1 regularizer greatly reduces the error to 3.60%. LDD-L1 outperforms the L1 regularizer and orthogonality-promoting regularizers, demonstrating that encouraging nonoverlap is more effective than encouraging sparsity alone or orthogonality alone in improving generalization performance. The error rates achieved by other state of the art methods are also listed.

5.3. LDD-L1 and Nonoverlap

We verify whether the LDD-L1 regularizer is able to promote nonoverlap. The study is performed on the SC model and the 20-News dataset. The number of basis vectors was set to 50. For 5 choices of the regularization parameter of LDD-L1: $\{10^{-4}, 10^{-3}, \cdots, 1\}$, we ran the LDD-L1-SC model until convergence and measured the overlap score (defined in Eq.2) of the basis vectors. The tradeoff parameter γ inside LDD-L1 is set to 1. Figure 3 shows that the overlap score consistently decreases as the regularization parameter of LDD-L1 increases, which implies that LDD-L1 can effectively encourage nonoverlap. As a comparison, we replaced LDD-L1 with LDD-only and L1-only, and measured the overlap scores. As can be seen, for LDD-only, the overlap score remains to be 1 when the regularization parameter increases, which indicates that LDD alone is not able to reduce overlap. This is because under LDD-only, the vectors remain dense, which renders their supports to be completely overlapped. Under the same regularization parameter, LDD-L1 achieves lower overlap score than L1, which suggests that LDD-L1 is more effective in promoting nonoverlap. Given that γ – the tradeoff parameter associated with the L1 norm in LDD-L1 – is set to 1, the same regularization parameter λ imposes the same level of sparsity for both LDD-L1 and L1-only. Since LDD-L1 encourages the vectors to be mutually orthogonal, the intersection between vectors' supports is small, which consequently results in small overlap. This is not the case for L1-only, which hence is less effective in reducing overlap.

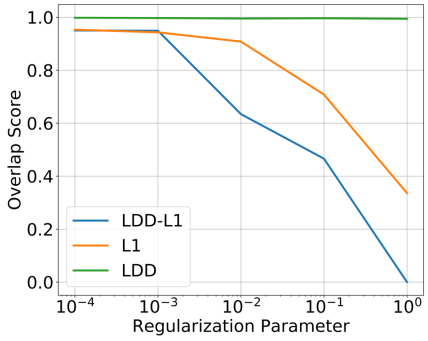

Figure 3. Overlap score versus the regularization parameter

6. Conclusions

In this paper, we propose a new type of regularization approach promoting a nonoverlap effect in variable selection. This regularizer encourages the weight vectors of different responses to be simultaneously sparse and orthogonal, which reduces the overlap among vectors' supports. We apply this regularizer to four exemplar ML models: multiclass logistic regression, distance metric learning, sparse coding, and deep neural networks. Efficient algorithms are developed for solving these regularized problems. Experiments on both simulated and real datasets demonstrate the effectiveness of this regularizer in selecting less-overlapped variables and improving generalization performance.

Acknowledgements

We would like to thank the anonymous reviewers for their very constructive and helpful comments and suggestions. Pengtao Xie and Eric P. Xing are supported by National Institutes of Health P30DA035778, Pennsylvania Department of Health BD4BH4100070287, and National Science Foundation IIS1617583.

References

Bach, F. R. Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in neural information processing systems*, pp. 105–112, 2009.

Bao, Y., Jiang, H., Dai, L., and Liu, C. Incoherent training of deep neural networks to de-correlate bottleneck features for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6980–6984. IEEE, 2013.

Bellet, A. and Habrard, A. Robustness and generalization for metric learning. *Neurocomputing*, 2015.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *the Journal of machine Learning research*, 2003.

Cai, D. and He, X. Manifold adaptive experimental design for text categorization. *Knowledge and Data Engineering, IEEE Transactions on*, 2012.

Cheng, W.-C., Kok, S., Pham, H. V., Chieu, H. L., and Chai, K. M. A. Language modeling with sum-product networks. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Cogswell, M., Ahmed, F., Girshick, R., Zitnick, L., and Batra, D. Reducing overfitting in deep networks by decorrelating representations. *ICLR*, 2015.

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*. ACM, 2007.

Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Gal, Y. and Ghahramani, Z. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 1019–1027, 2016.

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Graham, B. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014.

Guillaumin, M., Verbeek, J., and Schmid, C. Is that you? metric learning approaches for face identification. In *IEEE International Conference on Computer Vision*. IEEE, 2009.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

Inan, H., Khosravi, K., and Socher, R. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*, 2016.

Jacob, L., Obozinski, G., and Vert, J.-P. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pp. 433–440. ACM, 2009.

Kim, S. and Xing, E. P. Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eqtl mapping. *Annals of Applied Statistics*, 6 (3):1095–1117, 2012.

Kim, Y., Jernite, Y., Sontag, D., and Rush, A. M. Character-aware neural language models. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Kulis, B., Sustik, M. A., and Dhillon, I. S. Low-rank kernel learning with bregman matrix divergences. *Journal of Machine Learning Research*, 10:341–376, 2009.

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., and Tu, Z. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pp. 562–570, 2015.

Lin, M., Chen, Q., and Yan, S. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Mikolov, T. and Zweig, G. Context dependent recurrent neural network language model. 2012.

Mikolov, T., Karafiat, M., and Burget, L. Recurrent neural network based language model.

Mishkin, D. and Matas, J. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

Olshausen, B. A. and Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 1997.

Parikh, N. and Boyd, S. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

Press, O. and Wolf, L. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*, 2016.

Rodríguez, P., Gonzàlez, J., Cucurull, G., Gonfaus, J. M., and Roca, X. Regularizing cnns with locally constrained decorrelations. *arXiv preprint arXiv:1611.01967*, 2016.

Springenberg, J. T., Dosovitskiy, A., Brox, T., and Riedmiller, M. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Srivastava, R. K., Greff, K., and Schmidhuber, J. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

Verma, N. and Branson, K. Sample complexity of learning mahalanobis distance metrics. *arXiv preprint arXiv:1505.02729*, 2015.

Weinberger, K. Q., Blitzer, J., and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, 2005.

Xie, P., Deng, Y., and Xing, E. Diversifying restricted boltzmann machine for document modeling. In *SIGKDD*, 2015.

Xie, P., Deng, Y., Zhou, Y., Kumar, A., Yu, Y., Zou, J., and Xing, E. P. Learning latent space models with angular constraints. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3799–3810, 2017a.

Xie, P., Poczos, B., and Xing, E. P. Near-orthogonality regularization in kernel methods. 2017b.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

Xing, E. P., Jordan, M. I., Russell, S., and Ng, A. Y. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, 2002.

Yamada, Y., Iwamura, M., and Kise, K. Deep pyramidal residual networks with separated stochastic depth. *arXiv preprint arXiv:1612.01230*, 2016.

Yu, Y., Li, Y.-F., and Zhou, Z.-H. Diversity regularized machine. In *IJCAI*, 2011.

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

Zagoruyko, S. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Zaremba, W., Sutskever, I., and Vinyals, O. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Zhao, P. and Yu, B. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov): 2541–2563, 2006.

Zhao, P., Rocha, G., and Yu, B. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, pp. 3468–3497, 2009.

Zilly, J. G., Srivastava, R. K., Koutník, J., and Schmidhuber, J. Recurrent highway networks. *arXiv preprint arXiv:1607.03474*, 2016.

Zoph, B. and Le, Q. V. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.