Learning Compressible Models

Yi Zhang YIZHANG1@CS.CMU.EDU
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Jeff Schneider SCHNEIDE@CS.CMU.EDU
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Artur Dubrawski AWD@CS.CMU.EDU
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Abstract

Regularization is a principled way to control model complexity, prevent overfitting, and incorporate ancillary information into the learning process. As a convex relaxation of $\ell 0$-norm, $\ell 1$-norm regularization is popular for learning in high-dimensional spaces, where a fundamental assumption is the sparsity of model parameters. However, model sparsity can be restrictive and not necessarily the most appropriate assumption in many problem domains. In this paper, we relax the sparsity assumption to compressibility and propose learning compressible models: a compression operation can be included into $\ell 1$-regularization and thus model parameters are compressed before being penalized. We concentrate on the design of different model compression transforms, which can encode various assumptions on model parameters, e.g., local smoothness, frequency-domain energy compaction, and correlation. Use of a compression transform inside the $\ell 1$ penalty term provides an opportunity to include information from domain knowledge, coding theories, unlabeled data, etc. We conduct extensive experiments on brain-computer interface, handwritten character recognition, and text classification. Empirical results show significant improvements in prediction performance by learning compressible models instead of sparse models. We also analyze the model fitting and learned model coefficients under different compressibility assumptions, which demonstrate the advantages of learning compressible models instead of sparse models.

1. Learning Compressible Models

Since the introduction of lasso (Tibshirani, 1996), $\ell 1$-regularization has become very popular for learning in high-dimensional spaces. A fundamental assumption of $\ell 1$-regularization is the sparsity of model parameters, i.e., a large fraction of coefficients are zeros. This assumption might be too restrictive and not necessarily appropriate in some application domains. However, many signals in the real world (e.g., images, audio, videos) are found to

be compressible: sparse after being compressed. Naturally, the assumption of sparsity can be extended to compressibility. Inspired by the recent development of compressive sampling (or compressed sensing) (Candes, 2006; Donoho, 2006), we propose learning compressible models: model compression can be included in the $\ell 1$ penalty, and model parameters are compressed before being penalized.

In this section, we will review $\ell 1$-norm regularization, introduce the recent development of compressive sampling, and propose learning compressible models with $\ell 1$ regularization.

1.1 Regularization, $\ell 1$-Norm, and Learning Sparse Models

Regularization was initially proposed to solve ill-posed problems (Tikhonov and Arsenin, 1977). In statistical learning, regularization is widely used to control model complexity and prevent overfitting (Hastie et al., 2001). Regularization seeks a trade-off between fitting the observations and reducing the model complexity, which is justified by the minimum description length (MDL) principle in information theory (Rissanen, 1978) and the bias-variance dilemma in statistics (Sullivan, 1986). A general formula is shown as follows:

$$\min_{\beta} L_{\mathbf{D}}(\beta) + \lambda J(\beta) \tag{1}$$

where regularization is achieved by adding a penalty term $J(\beta)$ on model parameters β to the empirical loss $L_{\mathbf{D}}(\beta)$ defined on observations $\mathbf{D}$. The regularization parameter λ is usually empirically determined to control the strength of regularization and seek a balance between fitting the observations in terms of minimizing the empirical loss $L_{\mathbf{D}}(\beta)$ and searching a reasonable (e.g., simple) model in terms of minimizing the penalty term $J(\beta)$.

Since the introduction of lasso (Tibshirani, 1996), $\ell 1$-regularization has become popular for learning in high-dimensional spaces. As a specific example of the regularization framework in eq. (1), lasso is formulated as follows:

$$\min_{\alpha,\beta} ||\mathbf{y} - \mathbf{1}\alpha - \mathbf{X}\beta||_2^2 + \lambda||\beta||_1 \tag{2}$$

where the $n \times p$ training data $\mathbf{X}$ indicates n samples and p explanatory variables (features), and the $n \times 1$ vector $\mathbf{y}$ is the response variable (or labels) of n training examples, which usually contains $\{+1, -1\}$ for classification problems and real values for regression problems. The sum of squares error $||\ ||_2^2$ is an instantiation of the empirical loss $L_{\mathbf{D}}$ defined on $\mathbf{D} = \{\mathbf{X}, \mathbf{y}\}$. Also, $\mathbf{1}$ is the column of 1s, and the intercept α and $p \times 1$ vector β are model parameters. The intercept α is separated from β and not penalized in regularization, as shown in the formula above. The regularization parameter λ is empirically determined.

A notable part of lasso is the use of $\ell 1$ norm $||\beta||_1$ in regularization. As the closest convex relaxation of $\ell 0$-norm, $\ell 1$-norm regularization not only controls model complexity but also selects relevant features and produces parsimonious estimation (Tibshirani, 1996). Analytical results also show that $\ell 1$ regularization will lead to sparse estimation and is capable of consistently recovering sparse signals from noisy observations (Tropp, 2006; Zhao and Yu, 2006). As a result, a fundamental assumption of $\ell 1$-norm regularization is the sparsity of model parameters.

1.2 Compressive Sampling

The theory of compressive sampling (Candes, 2006) or compressed sensing (Donoho, 2006) was recently developed for signal acquisition and reconstruction, and has received considerable attention (Baraniuk et al., 2008). According to this theory, one can successfully recover a signal (e.g., an image) from many fewer measurements (i.e., samples) than required by Nyquist-Shannon sampling theory. The key is the requirement that the true signal is compressible, i.e., sparse in a certain transform domain. Under this assumption, accurate signal acquisition given a few (linear) measurements of the true signal β^* can be achieved by solving the problem:

$$\begin{aligned} \min_{\beta} \quad & ||\mathbf{W}\beta||_1 \\ \text{subject to} \quad & \mathbf{\Phi}\beta = \mathbf{y} \end{aligned} \tag{3}$$

Here β is a $p \times 1$ vector representing a candidate signal. $\mathbf{W}$ is a known $p \times p$ compression (or sparsifying) transform. The $n \times p$ matrix $\mathbf{\Phi} = [\phi_1, \ldots, \phi_n]^T$ is usually referred to as the sensing (or projection) matrix (Candes and Wakin, 2008; Donoho, 2006), whose n rows are the bases corresponding to the n linear measurements we want to sample from the signal. A common choice of $\mathbf{\Phi}$ is n random projections. Finally, the $n \times 1$ vector $\mathbf{y} = [\langle \beta^*, \phi_1 \rangle, \ldots, \langle \beta^*, \phi_n \rangle]^T$ are the n linear measurements (i.e., projections) we sampled from the true signal β^*. Since we have only a few measurements, i.e., $n < p$, the constraints $\mathbf{\Phi}\beta = \mathbf{y}$ give an underdetermined linear system. In practice, the measurements may contain noise. As a result, the problem is usually reformulated as:

$$\begin{aligned} \min_{\beta} \quad & ||\mathbf{W}\beta||_1 \\ \text{subject to} \quad & ||\mathbf{y} - \mathbf{\Phi}\beta||_2^2 \leq \epsilon \end{aligned} \tag{4}$$

where ϵ is our tolerance to noise. We minimize the $\ell 1$ norm of β under the compression transform $\mathbf{W}$, with the constraint that the recovered signal β is consistent with the n measurements $\mathbf{y}$ from the true signal. With n (e.g., $n < p$) available measurements on the signal, the sampling rate is lower than the Nyquist rate. But it is still possible to reconstruct the true signal β^* given the compressibility assumption. The Nyquist rate is a sufficient condition for exact signal recovery, not a necessary condition (Baraniuk et al., 2008): it refers to the worst case where we have no knowledge about the structure, e.g., compressibility, of the true signal.

The constrained optimization problem in eq. (4) is equivalent to the following form:

$$\min_{\beta} ||\mathbf{y} - \mathbf{\Phi}\beta||_2^2 + \lambda ||\mathbf{W}\beta||_1 \tag{5}$$

where both ϵ in eq. (4) and λ in eq. (5) are chosen empirically. As mentioned in (Kim et al., 2008), this regularized least squares formula is closely related to lasso in eq. (2), where the $n \times p$ sensing matrix $\mathbf{\Phi}$ is replaced by the training data $\mathbf{X}$ with n samples and p explanatory variables (features), and linear measurements $\mathbf{y}$ are replaced by the response variable of n training examples. A main difference between eq. (5) and eq. (2) is the compression transform $\mathbf{W}$ included in the $\ell 1$ penalty term. By adding a compression transform, the assumption of sparse models can be generalized to compressible models: model parameters are sparse after being compressed.

1.3 Learning Compressible Models

Extended from eq. (2), learning compressible models can be formulated as follows:

$$\min_{\alpha,\beta} L(\mathbf{y}, \mathbf{1}\alpha + \mathbf{X}\beta) + \lambda||\mathbf{W}\beta||_1 \tag{6}$$

The $p \times p$ matrix $\mathbf{W}$ represents the model compression transform, which encodes our assumptions on model parameters (as detailed in Section 2). The loss function L depends on the model, e.g., squared loss for linear regression, log-likelihood loss for logistic regression, hinge loss for SVMs, and so forth.

Mathematically, a linear transform $\mathbf{W}$ in the penalty term does not lead to any new formula. For kernel-based models that work on the inner products of examples (Muller et al., 2001), a linear transform in the penalty term corresponds to defining a specific linear kernel. More generally, eq. (6) can be solved by applying a linear transform to the feature space and solving the standard $\ell 1$ regularization in the transformed space, given that $\mathbf{W}$ is invertible (Kim et al., 2008). First, transform the training examples by

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{W}^{-1} \tag{7}$$

Second, solve the following standard $\ell 1$-regularized model (e.g., lasso, sparse logistic regression, etc.):

$$\min_{\alpha,\tilde{\beta}} L(\mathbf{y}, \mathbf{1}\alpha + \tilde{\mathbf{X}}\tilde{\beta}) + \lambda||\tilde{\beta}||_1 \tag{8}$$

Finally, the solution for eq. (6) is obtained by:

$$\beta = \mathbf{W}^{-1}\tilde{\beta} \tag{9}$$

$$\alpha = \alpha \tag{10}$$

This equivalence is derived from $\mathbf{X}\beta = \mathbf{X}\mathbf{W}^{-1}\tilde{\beta} = \tilde{\mathbf{X}}\tilde{\beta}$ and $||\mathbf{W}\beta||_1 = ||\mathbf{W}\mathbf{W}^{-1}\tilde{\beta}||_1 = ||\tilde{\beta}||_1$.

2. Model Compression

The focus of the present paper is to study the combination of $\ell 1$ regularization and compression operations. Model compression operations encode our assumptions about model parameters $\boldsymbol{\beta}$, which might come from domain knowledge, coding theories, unlabeled data, related tasks, etc. In this section, we explore a few reasonable assumptions and the corresponding compression transforms, which make learning compressible models an useful generalization to learning sparse models.

2.1 Local Smoothness

In this section, we discuss compression operations related to local smoothness assumptions on models. Smoothness characterizes the properties of derivatives of a function. For example, a constant (or piecewise constant) function has zero first-order derivatives at all (or most) locations, and a quadratic function has zero third-order derivatives at all locations. Here we will show that use of a compression transform is very flexible and can represent various smoothness assumptions.

2.1.1 Order-1 Smoothness

Suppose we have a natural order over model coefficients $\{\beta_j\}_{j=1}^p$, e.g., in temporal domains where each dimension corresponds to a time point, or spectral domains where each dimension corresponds to a frequency. The *order*-1 *smoothness* assumes the coefficients "do not change very often" along the natural order. Such an assumption characterizes the first-order derivatives. It has been studied in fused lasso (Tibshirani et al., 2005) where absolute values of the difference of successive coefficients, i.e., $\sum_{j=2}^p |\beta_j - \beta_{j-1}|$, are penalized[1]. It was also explored, more previously, in total variation minimization for noise removal and image enhancement (Rudin et al., 1992). As a motivating example, we show that the fused lasso penalty can be approximated by a linear and invertible compression in the $\ell 1$ penalty.

The $p \times p$ matrix $\mathbf{W}$ for model compression based on order-1 smoothness can be defined as:

$$\mathbf{W} = \mathbf{S}_p^1 = \begin{bmatrix} \frac{1}{p} & \frac{1}{p} & \dots & \dots & \dots & \frac{1}{p} \\ 1 & -1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \dots & \vdots \\ 0 & 0 & \dots & \dots & 1 & -1 \end{bmatrix} \tag{11}$$

The compressed model coefficients $\mathbf{W}\boldsymbol{\beta} = [\overline{\beta}, \beta_1 - \beta_2, \dots, \beta_{p-1} - \beta_p]$ tend to be sparse under the order-1 smoothness assumption. The averaging operation at the first row of $\mathbf{W}$ makes the transform invertible, and it can be rescaled by a small factor to approxiate the fused lasso penalty as $||\mathbf{W}\boldsymbol{\beta}||_1$. Also, invertible compression operations make the optimization simple, as mentioned in Section **??**.

2.1.2 Order-2 and Higher-Order Smoothness

Smoothness of high orders is also common. For example, a piecewise linear function has piecewise constant first-order derivatives, indicating zero second-order derivatives at most locations. This is defined as the *order*-2 *smoothness*. In this case, the $p \times p$ compression transform $\mathbf{W}$ can be:

$$\mathbf{W} = \mathbf{S}_p^2 = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{S}_{p-1}^1 \end{bmatrix} \cdot \mathbf{S}_p^1 \tag{12}$$

where $\mathbf{0}$ is a $(p-1) \times 1$ column vector. By this definition, the compressed model coefficients are $\mathbf{W}\beta = [\overline{\beta}, \overline{\Delta\beta}, \Delta\beta_{1,2} - \Delta\beta_{2,3}, \Delta\beta_{2,3} - \Delta\beta_{3,4}, \dots, \Delta\beta_{p-2,p-1} - \Delta\beta_{p-1,p}]$, where $\Delta\beta_{i,i+1} = \beta_i - \beta_{i+1}$. The compressed model coefficients are likely to form a nearly sparse vector given the order-2 smoothness assumption. Also, $\mathbf{S}_p^2$ is invertible since both $\mathbf{S}_{p-1}^1$ and $\mathbf{S}_p^1$ are invertible. Finally, model compression for *higher-order smoothness* assumptions can be defined recursively.

2.1.3 Hybrid Smoothness

Sometimes features under consideration do not follow an universal order, but can be divided into groups, where each group of features has an order or at least some groups of features are

1. In fused lasso, both standard $\ell 1$-norm and $\sum_{j=2}^p |\beta_j - \beta_{j-1}|$ are penalized to pursue both sparsity and smoothness. In this section, we focus on smoothness ($\sum_{j=2}^p |\beta_j - \beta_{j-1}|$) as a specific case of compressibility.

ordered. The model compression matrix for this case can be defined as a block matrix. For example, suppose features can be divided into three groups, and we assume the first group of p_1 features satisfy order-1 smoothness, the second group of p_2 features satisfy order-2 smoothness, and we have no knowledge about the third group of p_3 features. In this case, model compression can be defined as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{S}^1_{p_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}^2_{p_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{p_3} \end{bmatrix} \tag{13}$$

2.1.4 Multi-dimensional Orders

In some problem domains, features do not follow a unidimensional order. Consider a p-dimensional feature space defined on images, where the p pixels of an image are organized as m rows and n columns, i.e., $p \times 1 = m \times n$. The smoothness assumption can hold both row-wise and column-wise. In this case, the $p \times p$ compression transform $\mathbf{W}$ for model β is obtained as follows. First, imagine that we reshape the $p \times 1$ vector β into a $m \times n$ matrix $\mathbf{G}(\beta)$, according to the two-dimensional order in the feature space. Second, define the compression operation on an $m \times n$ object as coding an image (Wallace, 1992), e.g., if we assume order-1 smoothness on both rows and columns:

$$\mathbf{S}^1_m \cdot \mathbf{G}(\beta) \cdot {\mathbf{S}^1_n}^T \tag{14}$$

Thirdly, rewrite eq. (14), a linear operation on $m \times n$ matrices, into an equivalent linear operation on $p \times 1$ vectors. This leads to the $p \times p$ transform $\mathbf{W}$.

2.2 Energy Compaction

In this section, we discuss energy compaction in the frequency domain to compress models. Energy of many real-world signals can be compacted by transforming the signals to a frequency domain where most of their energy is concentrated in a few frequencies (Rao and Yip, 1990), e.g., images are effectively compressed this way (Wallace, 1992; Christopoulos et al., 2000). Consider object recognition problems. If the target model $\boldsymbol{\beta}$ is applied to images with such energy compaction properties, it is reasonable to assume that $\boldsymbol{\beta}$ also has compacted energy in frequency domains. Otherwise, energy of $\boldsymbol{\beta}$ is wasted. Naturally, including an appropriate spectral transform in $\ell 1$ penalty can emphasize compacted energy in frequency domains.

The discrete cosine transform (DCT) is used in the JPEG standard (Wallace, 1992), which compresses an object (e.g., an image) by representing it as a sum of cosine functions at various frequencies, and as a result, small coefficients can be discarded. The 2D discrete cosine transform for an $m \times n$ object is:

$$\begin{aligned} \mathbf{G}'(u,v) &= \frac{2}{\sqrt{mn}}\Lambda(u)\Lambda(v)\sum_{y=0}^{m-1}\sum_{x=0}^{n-1}\mathbf{G}(x,y)\cos\frac{(2x+1)u\pi}{2n}\cos\frac{(2y+1)v\pi}{2m} \\ \textit{where} \quad & u = 0, 1, \ldots, n-1 \\ & v = 0, 1, \ldots, m-1 \\ & \Lambda(t) = \begin{cases} 2^{-\frac{1}{2}} & if\ t = 0 \\ 1 & otherwise \end{cases} \end{aligned} \tag{15}$$

The above formula is a linear operation on $m \times n$ matrices, and can be rewritten as a linear operation on $p \times 1$ vectors, where $p = mn$ is the dimension of linear models on images. This gives a $p \times p$ matrix $\mathbf{W}$. Combining such a compression operation with $\ell 1$-norm regularization will lead to sparse models in an appropriate frequency domain, representing the compacted energy assumption on model coefficients. Note that transforms in real-world image compression protocols are more sophisticated (Wallace, 1992; Christopoulos et al., 2000), but studying sophisticated image codings is not the focus of this paper.

2.3 Correlation

Another common situation is that model coefficients are likely to be correlated. For example, in text classification problems, a document is represented by a bag of words, where each feature is a binary or count variable indicating the occurrence of a word. In this case, a true model $\boldsymbol{\beta}$ (intercept α omitted) for a problem is a linear function defined on the vocabulary, and each dimension β_j indicates the effect of the jth word in the decision. In any language, there exist certain semantic structure among words, which leads to the correlation of words in constituting the meaning in an expression, and more specifically, the correlation of their roles in a natural function $\boldsymbol{\beta}$. This structure has been studied as the semantic correlation of words (Raina et al., 2006; Zhang et al., 2008; Nallapati et al., 2007; Yang et al., 2008) in machine learning context.

In this case, sparse estimation with ℓ-norm penalty might be questionable. From the frequentist perspective, the true model is unlikely to be very sparse if coefficients are highly correlated: nonzero coefficients on a few words suggest nonzeros on many other semantically correlated words. It is also easy to understand from the Bayesian perspective. Imposing a $\ell 1$ penalty corresponds to assuming a Laplacian prior over model coefficients. However, a Laplacian prior indicates independent (and thus uncorrelated) model coefficients, which contradicts the existence of semantic word correlation.

A simple but effective solution is to decorrelate (i.e., compress) the model parameters before penalization. Given a correlation structure $\boldsymbol{\Sigma}$ on model coefficients, the compression transform is:

$$\mathbf{W} = \boldsymbol{\Sigma}^{-\frac{1}{2}} \tag{16}$$

As a result, eq. (6) is rewritten as:

$$\min_{\alpha,\boldsymbol{\beta}} L(\mathbf{y}, \mathbf{1}\alpha + \mathbf{X}\boldsymbol{\beta}) + \lambda ||\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\beta}||_1 \tag{17}$$

The true model is more likely to be sparse under the transform $\mathbf{W} = \boldsymbol{\Sigma}^{-\frac{1}{2}}$ since the transformed model coefficients has a correlation structure $\boldsymbol{\Sigma}' = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-\frac{1}{2}} = \mathbf{I}$. From the Bayesian perspective, eq. (17) assumes a Laplacian prior on $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\beta}$. Thus, $\boldsymbol{\beta}$ itself has a transformed (by $\boldsymbol{\Sigma}^{\frac{1}{2}}$) Laplacian prior, which is consistent with the correlation structure $\boldsymbol{\Sigma}$ on $\boldsymbol{\beta}$ in our belief. To optimize eq. (17), we will solve a standard $\ell 1$ regularization as eq. (8), where data space is transformed by $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{W}^{-1} = \mathbf{X}\boldsymbol{\Sigma}^{\frac{1}{2}}$. Interestingly, we are not trying to decorrelate the data space. Quite the reverse, we further correlate the data $\mathbf{X}$ to decorrelate the model space.

The last question is how can we estimate the correlation structure $\mathbf{\Sigma}$. Indeed, the correlation $\mathbf{\Sigma}$ is even harder to estimate than the linear model β itself, since the correlation matrix has more degrees of freedom. However, unlike the linear function β for a specific decision problem, the correlation structure of the function space is sometimes more general and can be learned from other sources of information, e.g., unlabeled data, related problems, or domain knowledge. For text classification, we propose a method to learn the semantic correlation of words from unlabeled documents (Zhang et al., 2008). Given a large mixture of unlabeled documents on various topics (which are readily available from the web), the semantic correlation of words can be learned by 1) repeatedly sampling the unlabeled documents (i.e., bootstrapping); 2) extracting latent topics from the sampled documents (via topic models); 3) collecting all the extracted latent topics; and 4) observing the correlation of word occurrence in these latent topics. Such a correlation structure, unlike the correlation of word occurrence in documents, is shown analytically and empirically to be transferable from seemingly irrelevant unlabeled documents (i.e., on various random topics) to a problem defined in the same vocabulary space (Zhang et al., 2008).

3. Empirical Study: Brain-Computer Interface

In this section, we report our empirical study on brain-computer interface data (Blankertz et al., 2004): classifying single-trial Electroencephalography (EEG) signals. The EEG signals contain information from human brains, and understanding EEG signals is important for human-computer interaction. An EEG signal contains multiple channels (i.e., multiple scalp positions), and each channel is sampled over time to produce sequential measurements. As a result, an EEG signal is a multivariate time series. If we assume local smoothness over time on an univariate time series, the block smoothness assumption introduced in Section 2.1.3 is suitable for multivariate EEG classifiers.

3.1 Experimental Settings

Data Set. We use data set IV, self-paced tapping, of BCI Competition 2003 (Blankertz et al., 2004), which is a binary classification problem. The data set contains a training set of 316 examples and a testing set of 100 examples. Each example has 1400 features, corresponding to 28 channels and 50 measurements from each channel. Each example is measured when a healthy subject, sitting in a chair with fingers in the standard typing position, tries to press the keys using either the left hand or right hand. The objective is to classify an EEG signal to either a left-hand movement or right-hand movement.

Tasks. The data set contains a fixed training and testing set for competition. We train lasso, sparse logistic regression, compressible lasso, compressible logistic regression on the training set and measure their errors on the testing set. Note that there is still randomness: cross-validation is used to determine the optimal regularization parameter. Therefore, we have 50 random runs.

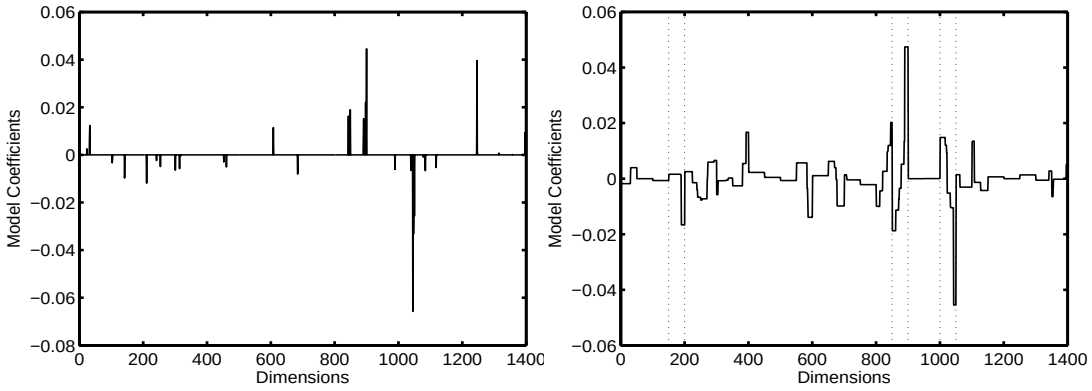
Models and Implementations. We consider lasso with labels as $\{+1, -1\}$ (Lasso), sparse logistic regression (SLgr), compressible lasso (lassoCP), and compressible logistic regression (SLgrCP). Following Section 2.1.3, compression transform $\mathbf{W}$ is a 1400×1400 block matrix, with 28 blocks and each defined as an order-1 smoothness matrix $\mathbf{S}_{50}^{1}$. Lasso

Table 1: Classification errors over 50 random runs: mean (standard deviation)

Lasso	LassoCP	SLgr	SLgrCP
30.22%(2.4%)	25.98%(2.07%)	30.00%(0%)	20.92%(1.14%)

Table 2: Model fitting (MSE or -2Loglike on test set) over 50 random runs: mean (std)

Lasso: MSE	LassoCP: MSE	SLgr: -2Loglike	SLgrCP: -2Loglike
0.764(0.00001)	0.688(0.00167)	119.92(0)	99.97(1.07)



(a) Sparse LGR coefficients (b) Compressible LGR coefficients

Figure 1: Learned Model Coefficients on BCI

is implemented using the spgl1 Matlab solver[2], and $\ell 1$-regularized logistic regression is implemented as (Lee et al., 2006) using lasso. The regularization parameter is chosen from 10^{-7} to 10^{7} (step $10^{0.5}$) by 5-fold cross-validation.

3.2 Results and Analysis

Results are shown in Table 1, with both mean and standard deviation of classification errors over 50 random runs. By learning compressible models, the test error is reduced from 30.22% to 25.98% for lasso, and from 30% to 20.92% for logistic regression. During the competition (Blankertz et al., 2004), 15 submissions were received. Compressible logistic regression is comparable to the 2nd best submission (19%), which uses 188 selected time-based, frequency-based, and correlational features "compiled by hand" (Blankertz et al., 2004). The best submission of the competition achieves 16% error, using features "based on Bereitschaftspotential and event-related desynchronization" (Wang et al., 2004). For other 13 submissions, six attained errors between 23% and 29%, and other seven were worse. In our study, linear models using 1400 raw features can be comparable to two best submissions with domain-specific features.

2. http://www.cs.ubc.ca/labs/scl/spgl1/

Table 3: Classification errors averaged over 45 tasks

	10 per class	20 per class	50 per class
Lasso	9.96%	6.94%	4.91%
LassoCP	7.80%	5.30%	3.45%
SLgr	9.79%	6.24%	3.91%
SLgrCP	7.46%	4.99%	3.26%
(Lasso - LassoCP)	2.16%(1.52%)	1.64%(1.21%)	1.46%(0.81%)
(SLgr - SLgrCP)	2.33%(1.40%)	1.25%(1.04%)	0.65%(0.62%)

Table 4: Performance comparison on individual tasks: win/loss

	10 per class	20 per class	50 per class
LassoCP vs. Lasso	41/4	42/3	44/1
SLgrCP vs. SLgr	43/2	42/3	40/5

Table 5: Model fitting (MSE or -2Loglike on test set) averaged over 45 tasks

	10 per class	20 per class	50 per class
Lasso: MSE	0.2791	0.2522	0.2394
LassoCP: MSE	0.2885	0.2483	0.1939
SLgr: -2Loglike	1147.10	859.24	621.47
SLgrCP: -2Loglike	931.94	659.16	460.10

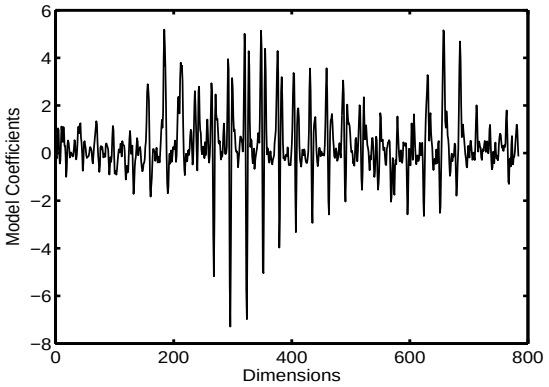
In addition to classification errors, we also report model fitting (MSE for lasso, -2Log-likelihood for logistic regression) on the test set as a performance measure, shown in Table 2. If the compression penalty achieves a better bias-variance tradeoff, we might expect to see improvements of model fitting on the test set, especially for logistic regression, because the model fitting of logistic regression is a good indicator for classification performance.

We also plot the model coefficients learned by logistic regression in Figure 1. From the plot we can see that 1) sparse LGR estimates sparse coefficients, and compressible LGR leads to smooth coefficients; 2) Although in compressible models we only penalize the difference of successive coefficients, most coefficients themselves are close to zeros; 3) In compressible models, there still exist large coefficient jumps within channels, corresponding to large coefficients after compression. We plot the boundaries (vertical dashed lines) of three channels that contain large coefficient jumps[3].

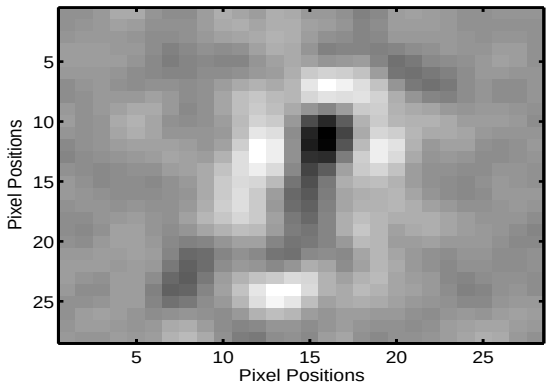
4. Empirical Study: Handwritten Character Recognition

In this section, we study the handwritten character recognition problem on images. As discussed in Section 2.2, the compacted energy assumption can be used for model compression.

3. Ideally, $\ell 0$ norm will allow some large coefficient jumps, while $\ell 1$ norm is a convex relaxation. Also, $\ell 1$-norm can be superior to $\ell 2$-norm in this case: large coefficient jumps will be severely penalized by $\ell 2$ norm.

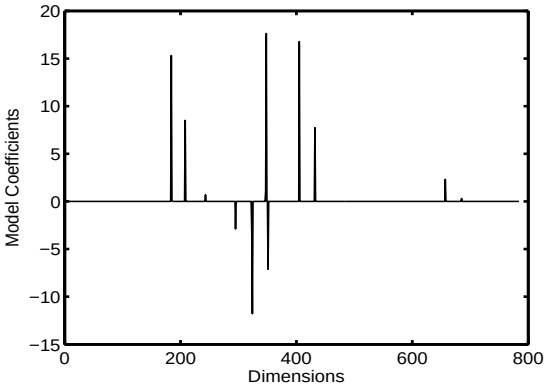


(a) Model coefficients (change at a few frequencies)

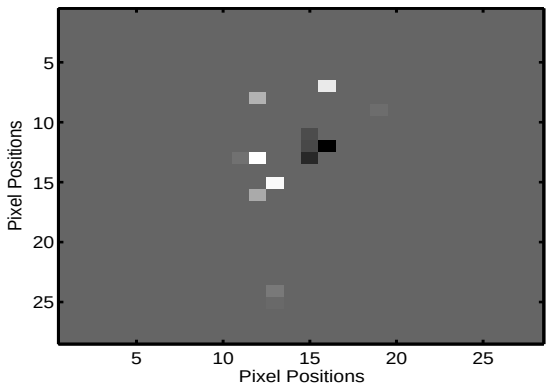


(b) Model coefficients as an image

Figure 2: Compressible logistic regression on MNIST. Task: "1" vs. "8"



(a) Model coefficients (sparse)



(b) Model coefficients as an image

Figure 3: Sparse logistic regression on MNIST. Task: "1" vs. "8"

4.1 Experimental Settings

Data Set. We use the MNIST database of handwritten digits[4], which contains images for 10 different digits (from 0 to 9). Images are represented by pixels (in grayscale). The number of features is $p = 784$, corresponding to 28×28 pixels of an image.

Tasks. We construct 45 binary classification tasks, each to classify two digits. For each task, a few labeled examples of the two digits are selected from the training set (e.g., 10, 20, or 50 images *per class*). Performance for each task is averaged from 20 random runs, with training data randomly selected. For each task, the testing data are fixed as all the images of the two digits in the testing set.

Models and Implementations. DCT-based compression is as eq. (15). Others are as Section 3.

4. http://yann.lecun.com/exdb/mnist/

4.2 Results and Analysis

Results are shown in Table 3, Table 4 and Figure 3. The first part of Table 3 is average classification errors over 45 tasks of Lasso, compressible Lasso, sparse LGR and compressible LGR. We omit standard deviations since they correspond to the intrinsic variation of difficulty of different tasks, which is not of interest. The last two rows of Table 3 are means (and standard deviations) of the *difference* of classification errors between a sparse and a compressible model. Although difficulty of tasks is "normalized" to a certain degree in the sense that difficult tasks are difficult for both sparse and compressible models, standard deviations here still represent some intrinsic variation among different tasks. To be more comprehensive, we also compare model performance on individual tasks in Table 4, which shows that the majority of tasks are benefited from learning compressible models.

It's also interesting to see the actual model coefficients with compacted energy in a frequency domain. We plot in Figure 3 a compressible logistic regression model for the task "1" (-) vs. "8" (+). Coefficients in the original domain in Fig. 2a indicate a compacted energy in the frequency domain, and when plotted as a 28×28 image in Fig. 2b, highlight the difference between "1" (-) and "8" (+).

Finally, we also report model fitting (MSE for lasso, -2Log-likelihood for logistic regression) on the test set, averaged over 45 tasks. This is shown in Table 5. Note that the model fitting of logistic regression is significantly improved by learning compressible models instead of sparse models. On the other hand, the MSE of lasso is not consistently improved by learning compressible models (however, classification errors on the test set are clearly improved, as in Table 3 and Table 4). This is perhaps because the model fitting error (MSE) for regression models is not a reasonable loss function for classification purpose, and as a result, the compressibility assumption on model coefficients, even as a reasonable assumption for true (classification) models, does not necessarily improve regression errors (MSE).

5. Empirical Study: Text Classification

This section studies text classification. As mentioned in Section 2.3, we include a decorrelation transform $\mathbf{W} = \mathbf{\Sigma}^{-\frac{1}{2}}$ as compression. The semantic word correlation $\mathbf{\Sigma}$ is estimated from unlabeled text (Zhang et al., 2008), and thus learning compressible models offers an approach for semi-supervised learning.

5.1 Experimental Settings

Data set. We use the 20-Newsgroups data set[5]. It contains 11314 training and 7532 testing documents from 20 newsgroups, denoted as D_{tr} and D_{ts}, respectively. Documents are represented as bags of words. The vocabulary includes the most frequent 200 words in each newsgroups except the 20 most frequent common words across all newsgroups, leading to $p = 1443$ features (words).

Tasks. Documents are from 20 newsgroups, so we construct 190 binary classification tasks, each to classify a pair of newsgroups. For each task, we randomly sample 2% and 5% of the relevant documents in D_{tr} as the labeled training examples. Two newsgroups

5. http://people.csail.mit.edu/jrennie/20newsgroups

Table 6: Classification errors averaged over 190 tasks

	2% sampling	5% sampling
Lasso	22.17%	17.02%
ElasNet	19.97%	12.87%
LassoCP	11.13%	7.76%
SLgr	21.69%	15.28%
SLgrCP	9.31%	6.19%
(Lasso - LassoCP)	11.05%(3.09%)	9.26%(2.91%)
(ElasNet - LassoCP)	8.84%(3.82%)	5.11%(2.53%)
(SLgr - SLgrCP)	12.38%(4.09%)	9.09%(2.98%)

Table 7: Performance comparison on individual tasks: win/loss

	2% sampling	5% sampling
LassoCP vs. Lasso	190/0	190/0
LassoCP vs. ElasNet	188/2	188/2
SLgrCP vs. SLgr	190/0	190/0

of a task are sampled together to simulate imbalanced training examples for text learning. Results of each task are averaged over 10 random runs. Testing data for each task are fixed to be all relevant documents in D_{ts}.

Models and Implementations. In addition to lasso, compressible lasso, sparse logistic regression and compressible logistic regression, we also test elastic net (Zou and Hastie, 2005). Elastic net is designed to handle correlated model coefficients, and as a convex combination of $\ell 1$ norm and $\ell 2$ norm, provides superior performance to both norms[6]. For all models, the intercept α is added into the penalty term, which slightly improves the performance. This is possibly because α tends to overfit the imbalanced class distribution in training examples. For lasso and logistic regression, the $\ell 1$ norm bound is chosen from 10^{-7} to 10^{7} with a larger step 10^{1} (for computation efficiency). Other details are the same as Section 3. For elastic net, the $\ell 1$ norm bound is choose in the same way, and the second parameter λ_2 (Zou and Hastie, 2005) is chosen from 10^{-4} to 10^{4} with step 10^{1}. For compressible models, decorrelation is used for compression, as eq. (17). Correlation $\mathbf{\Sigma}$ is estimated using all the documents in D_{tr} as unlabeled data, and is used for all 190 tasks. Note that D_{tr} is a mixture of documents from 20 newsgroups, so for each binary task, most unlabeled data are irrelevant. We use the method in (Zhang et al., 2008) to estimate word correlation.

5.2 Results and Analysis

Results are shown in Table 6 and Table 7. The first part of Table 6 contains average classification errors over 190 different tasks using lasso, elastic net, compressible lasso, sparse logistic regression, and compressible logistic regression. The second part is the difference

6. Elastic net has two regularization parameters controlling both $\ell 1$ and $\ell 2$ norm. With cross-validation to determine those parameters, it includes $\ell 1$ and $\ell 2$ regularization as two specific cases.

of classification errors between related models, including both means and standard deviations. Again, standard deviations represent the intrinsic variations of different tasks. For regression-based models (i.e., squared error as loss function), elastic net improves standard lasso by using a convex combination of $\ell 1$ and $\ell 2$ penalty. Elastic net is designed to address correlated model coefficients (Zou and Hastie, 2005), which justifies that there exist correlation among model coefficients, corresponding to semantic correlation of words. Further, compressible lasso show significant improvements over both lasso and elastic net. It is not surprising to see compressible lasso is superior to elastic net, because it includes additional information from unlabeled text (i.e., correlation structure) for model compression. For logistic regression based models, compressible models also show notable improvements over sparse models. Finally, Table 7 compares model performance on individual tasks. Compressible models dominate other models in almost all tasks, which shows the significance of results and indicates that learning compressible models is very effective and reliable in text domains.

Due to very large in-class variations of documents, model fitting errors (MSE or -loglike) on the test set of both lasso and logistic regression are high, especially for logistic regression, where a few documents in the test set are assigned zero probability. We omit results here.

6. Related Work and Discussion

Recently, compressive sampling has drawn considerable attention from the machine learning community. In (Ji and Carin, 2007), researchers suggest using sparse Bayesian regression and active learning to solve the CS problem in eq. (3) and adaptively "learn" the optimal projection matrix $\mathbf{\Phi}$. Authors in (Calderbank et al.; Zhou et al., 2007) consider classification and regression problems, respectively, where data are compressed and not directly observable, e.g., only random projections of data can be accessed. Hegde et al. (Hegde et al., 2007) relate CS to manifold learning. They assume that signals to be reconstructed are generated from a low-dimensional manifold, and propose to discover this manifold from a few measurements on signals. In the present work we study another extension of compressive sampling to machine learning, where model coefficients are compressed before being penalized, and the focus of this paper is the combination of different compression operations with $\ell 1$-norm penalty useful in real-world applications.

Researchers has proposed various penalties in regularization to incorporate model assumptions and explore different bias-variance tradeoff. Fused lasso (Tibshirani et al., 2005) includes a penalty on the absolute difference of successive coefficients, which can be approximated by a linear invertible transform in $\ell 1$ penalty as in Section 2.1.1. Elastic net (Zou and Hastie, 2005) combines $\ell 1$ and $\ell 2$ norms to address correlated coefficients and achieve better model fitting than both $\ell 1$ and $\ell 2$ regularization. Group lasso (Yuan et al., 2006) is useful when a natural grouping of variables is available, and variables in the same group tend to be eliminated together. Adaptive lasso (Zou, 2006) includes adaptive weights for coefficients in $\ell 1$ penalty and reduces the bias in lasso. It can be a complement to learning compressible models: including adaptive weights in the compression transform.

One might also consider the combination of model compression with other penalty norms, e.g., $\ell 2$ norm, which leads to specific cases of generalized Tikhonov regularization. However, whether a specific model compression assumption is consistent with $\ell 2$ penalty

deserves further studies. For example, for smoothness compression used for EEG signal classifiers (Section 3), we observed large coefficient jumps in the empirically chosen model. These jupms will be severely penalized in $\ell 2$ norm. Also, if most energy of the model is concentrated at a few frequencies, as in Section 4, we will expect to have large coefficients in the frequency domain, which contradicts the use of $\ell 2$ penalty. This topic will be explored in our future work.

7. Summary

We study the inclusion of different compression operations into the $\ell 1$ penalty, which encodes various compressibility assumptions useful in real-world applications, e.g., local smoothness, compacted energy in frequency domains, and correlation. We show that use of a compression transform provides an opportunity to incorporate information from domain knowledge, coding theories, unlabeled data, etc. We conduct extensive experiments on brain-computer interface, handwritten character recognition, and text classification. Empirical results show significant improvements in prediction performance by including compression in the $\ell 1$-norm penalty. We also analyze the learned model coefficients under different compressibility assumptions, which demonstrate the advantages of learning compressible models instead of sparse models.

References

R. G. Baraniuk, E. J. Candes, R. Nowak, and M. Vetterli. Compressive Sampling (Special Issue). *IEEE Signal Processing Magazine*, 25:12–101, 2008.

B. Blankertz et al. The BCI Competition 2003: Progress and Perspectives in Detection and Discrimination of EEG Single Trails. *IEEE Trans. Biomedical Engineering*, 51(6): 1044–1051, 2004.

Robert Calderbank, Sina Jafarpour, and Robert Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Preprint, 2009.

E. J. Candes. Compressive Sampling. In *Proceedings of International Congress of Mathematicians*, 2006.

E. J. Candes and M. B. Wakin. An Introduction to Compressive Sampling. *IEEE Signal Processing Magazine*, 25:21–30, 2008.

C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG2000 Still Image Coding System: An Overview. *IEEE Trans. Consumer Electronics*, 46(4):1103–1127, 2000.

D. L. Donoho. Compressed Sensing. *IEEE Trans. Information Theory*, 52(4):1289–1306, 2006.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, New York, 2001.

C. Hegde, M. B. Wakin, and R. G. Baraniuk. Random Projections for Manifold Learning. In *NIPS*, 2007.

S. Ji and L. Carin. Bayesian Compressive Sensing and Projection Optimization. In *ICML*, 2007.

S. J. Kim, K. Ko, M. Lustig, S. Boyd, and D. Gorinevsky. An Interior-Point Method for Large-Scale L1-Regularized Least Squares. *IEEE Journal of Selected Topics in Signal Processing*, 1:606–617, 2008.

S. I. Lee, H. Lee, P. Abbeel, and A. Y. Ng. Efficient L-1 Regularized Logistic Regression. In *AAAI*, 2006.

K-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An Introduction to Kernel-Based Learning Algorithms. *IEEE Trans. Neural Networks*, 12(2):181–201, 2001.

R. Nallapati, A. Ahmed, W. Cohen, and E. Xing. Sparse Word Graphs: A Scalable Algorithm for Capturing Word Correlations in Topic Models. In *ICDM workshop on High Performance Data Mining*, 2007.

Rajat Raina, Andrew Y. Ng, and Daphne Koller. Constructing Informative Priors using Transfer Learning. In *ICML*, pages 713–720, 2006.

K. R. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications.* Academic Press, 1990.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

F. O. Sullivan. A statistical perspective on ill-posed inverse problems. *Statist. Sci.*, 1: 502–518, 1986.

R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Soceity, Series B*, 58(1):267–288, 1996.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal Of The Royal Statistical Society Series B*, 67(1):91–108, 2005.

A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed Problems.* Winston and Sons, 1977.

Joel A. Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.

G. K. Wallace. The JPEG Still Picture Compression Standard. *IEEE Trans. Consumer Electronics*, 38(1):xviii–xxxiv, 1992.

Y. Wang et al. The BCI Competition 2003 - Data Set IV: An Algorithm Based on CSSD and FDA for Classifying Single-Trail EEG. . *IEEE Trans. Biomedical Engineering*, 51(6): 1081–1086, 2004.

Liu Yang, Rong Jin, and Rahul Sukthankar. Semi-supervised learning with weakly-related unlabeled data: Towards better text categorization. In *NIPS*, pages 1857–1864, 2008.

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

Y. Zhang, J. Schneider, and A. Dubrawski. Learning the Semantic Correlation: An Alternative Way to Gain from Unlabeled Text. In *NIPS*, 2008.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7: 2541–2563, 2006.

S. Zhou, J. Lafferty, and L. Wasserman. Compressed Regression. In *NIPS*, 2007.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320, 2005.