# Learning Compressible Models

Yi Zhang* Jeff Schneider† Artur Dubrawski‡

**Abstract**

In this paper, we study the combination of compression and $\ell 1$-norm regularization in a machine learning context: learning compressible models. By including a compression operation into the $\ell 1$ regularization, the assumption on model sparsity is relaxed to compressibility: model coefficients are compressed before being penalized, and sparsity is achieved in a compressed domain rather than the original space. We focus on the design of different compression operations, by which we can encode various compressibility assumptions and inductive biases, e.g., piecewise local smoothness, compacted energy in the frequency domain, and semantic correlation. We show that use of a compression operation provides an opportunity to leverage auxiliary information from various sources, e.g., domain knowledge, coding theories, unlabeled data. We conduct extensive experiments on brain-computer interfacing, handwritten character recognition and text classification. Empirical results show clear improvements in prediction performance by including compression in $\ell 1$ regularization. We also analyze the learned model coefficients under appropriate compressibility assumptions, which further demonstrate the advantages of learning compressible models instead of sparse models.

**Keywords:** linear models, $\ell 1$ regularization, sparse models, compressible models, compression, inductive bias.

## 1 Learning Compressible Models

Since the introduction of lasso [21], $\ell 1$-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. Sparse models have the advantage of being easy to interpret and good generalization ability in very high-dimensional problems. However, the sparsity assumption on model coefficients might be too restrictive and not necessarily appropriate in many application domains. Indeed, many signals in the real world (e.g., images, audio, videos, time series) are found to be compressible (i.e., sparse in certain compressed domain) but not directly sparse in the observed space. Naturally, the assumption of sparsity can be relaxed to compressibility. Inspired by the recent development of compressive sampling (or compressed sensing) [5, 9], we study *learning compressible models*: a compression on model coefficients can be included in the $\ell 1$ penalty, and model is assumed to be sparse *after* compression.

The rest of this paper is organized as follows. In section 1.1, we will briefly introduce learning sparse models with $\ell 1$-norm regularization. In Section 1.2 we discuss the definition, computation issues and potential benefits of learning compressible models. In Sections 2–4, we propose three classes of model compressibility assumptions and corresponding compression operations: piecewise local smoothness, energy compaction in the frequency domain, and semantic correlation. In Sections 5–7, we empirically study several real-world problems: brain-human interfacing, handwritten digit recognition and text classification, using compressibility as a more appropriate inductive bias than sparsity. Experimental results demonstrate the advantages of learning compressible models. Section 8 discusses related work and Section 9 concludes and mentions future work.

### 1.1 Learning Sparse Models with $\ell$-Norm Regularization.

Regularization was initially proposed to solve ill-posed problems [23]. In statistical learning, regularization is widely used to control model complexity and prevent overfitting [11]. 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 [18] and the bias-variance dilemma in statistics [20].

Lasso [21] is a specific example of $\ell 1$-norm regularization, which is formulated as:

$$\min_{\alpha,\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{1}\alpha - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1 \tag{1.1}$$

where the $n \times p$ matrix $\mathbf{X}$ contains $n$ examples and $p$ explanatory variables (i.e., features), and the $n \times 1$ vector $\mathbf{y}$

*Machine Learning Department, Carnegie Mellon University. Email: yizhang1@cs.cmu.edu

†Robotics Institute, Carnegie Mellon University. Email: schneide@cs.cmu.edu

‡Robotics Institute, Carnegie Mellon University. Email: awd@cs.cmu.edu

is the response variable (or 0/1 labels) of training examples. The sum of squares error $|| \, ||_2^2$ is an instantiation of the empirical loss function on observations. Also, $\mathbf{1}$ is a column of 1s, and the intercept $\alpha$ and $p \times 1$ vector $\boldsymbol{\beta}$ are model parameters. The intercept $\alpha$ is usually separated from $\boldsymbol{\beta}$ and not penalized in regularization. The regularization parameter $\lambda$ is a balance between minimizing the empirical loss and controlling the model complexity $||\boldsymbol{\beta}||_1$, and is usually determined by cross-validation.

A notable part of lasso is the use of $\ell 1$ norm $||\boldsymbol{\beta}||_1$ as the regularization term. As the closest convex relaxation of $\ell 0$-norm, $\ell 1$-norm in regularization not only controls model complexity but also leads to sparse estimation [21]. This provides both interpretable model coefficients and generalization ability. Analytical results also show that $\ell 1$ regularization is capable of consistently recovering the true signal from noisy measurements [25, 30], given that the true signal is sparse.

**1.2 Learning Compressible Models:** Assuming the model to be sparse and shrinking model coefficients to exactly zero might not be the most appropriate inductive bias in many problems. For example, real-world signals (such as audio, images, videos and time series) are usually compressible but not directly sparse in the observation domain. Interestingly, compressive sampling [5] or compressed sensing [9] was recently developed in signal acquisition, which is also based on $\ell 1$ penalized optimization framework but assumes that the target signal is compressible, i.e., sparse after being compressed.

The compressibility can also be used as an inductive bias in a machine learning context, which relaxes the sparse assumption used by lasso and provides a more appropriate assumption. We consider the problem of learning compressible models as follows:

$$\min_{\alpha,\boldsymbol{\beta}} L(\mathbf{y}, \mathbf{1}\alpha + \mathbf{X}\boldsymbol{\beta}) + \lambda ||\mathcal{W}(\boldsymbol{\beta})||_1 \tag{1.2}$$

The loss function $L$ depends on the prediction model, e.g., sum of squares loss for linear regression, log-likelihood loss for logistic regression, hinge loss for SVMs, and so forth. The compression operation $\mathcal{W}()$ encodes our assumption on compressibility: model coefficients are compressed by $\mathcal{W}()$ before being penalized, and thus tend to follow the compression pattern (i.e., sparse in the compressed domain) rather than simply shrink to zero.

For simplicity, we restrict our attention to linear compression. Given that the compression operation is a linear and invertible[1] transform, learning compressible models is represented by:

$$\min_{\alpha,\boldsymbol{\beta}} L(\mathbf{y}, \mathbf{1}\alpha + \mathbf{X}\boldsymbol{\beta}) + \lambda ||\mathbf{W}\boldsymbol{\beta}||_1 \tag{1.3}$$

The $p \times p$ matrix $\mathbf{W}$ denotes the linear and invertible compression transform, where $p$ is the dimensionality of model coefficients as in (1.1). The optimization of eq. (1.3) can be achieved by applying the inverse compression transform (i.e., the decompression operation) to the feature space and solving the standard $\ell 1$ regularization in the transformed space [14]. First, transform the training examples by

$$\tilde{\mathbf{X}} = \mathbf{X}\mathbf{W}^{-1} \tag{1.4}$$

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

$$\min_{\alpha,\tilde{\boldsymbol{\beta}}} L(\mathbf{y}, \mathbf{1}\alpha + \tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}}) + \lambda ||\tilde{\boldsymbol{\beta}}||_1 \tag{1.5}$$

Finally, the solution for eq. (1.3) is obtained by:

$$\boldsymbol{\beta} = \mathbf{W}^{-1}\tilde{\boldsymbol{\beta}} \tag{1.6}$$

$$\alpha = \alpha \tag{1.7}$$

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

Why do we want to learn compressible models, which are not necessarily sparse in the original space? Compressible models are useful in several aspects. The first is model fitting and prediction accuracy. The inductive bias of model compressibility might be more appropriate than model sparsity, especially if an informative compression operation is specified based on additional information from domain knowledge, unlabeled data or related problems. The second reason, as claimed for standard sparse models, is interpretability. Model coefficients that are sparse in a compressed domain can still be insightful in the original space in many problems, as later shown by our empirical studies on brain-computer interface (in Section 5) and handwritten digit recognition (in Section 6). The third reason is that, when the compression operation is known in advance, compressible models are very efficient for storage and transmission in the compressed domain. This advantage has been widely recognized in compressive sensing [5, 9] for general signals and thus also valid when signals are model coefficients.

## 2 Model Compression: 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

[1] A compression transform needs to be invertible, so that the compressed signal can be decompressed.

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 on model coefficients.

**2.1 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. *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 [22] where absolute values of the difference of successive coefficients, i.e., $\sum_{j=2}^p |\beta_j - \beta_{j-1}|$, are penalized[2]. This idea was also explored in total variation minimization for noise removal and image enhancement [19]. 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:

$$\mathbf{W} = \mathbf{S}_p^1 = \begin{bmatrix} \frac{1}{p} & \frac{1}{p} & \dots & \dots & \dots & \frac{1}{p} \\ 1 & -1 & 0 & \dots & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \dots & \vdots \\ 0 & 0 & \dots & \dots & 1 & -1 \end{bmatrix} \tag{2.8}$$

Model coefficients in the compressed domain $\mathbf{W}\boldsymbol{\beta} = [\overline{\beta}, \beta_1 - \beta_2, \dots, \beta_{p-1} - \beta_p]$ tend to be sparse due to $\ell 1$ regularization, which achieves the order-1 smoothness. The averaging operation in the first row of $\mathbf{W}$ makes the transform invertible. Note that if the first row of $\mathbf{W}$ is multiplied by a small constant (e.g., 0.001), $||\mathbf{W}\boldsymbol{\beta}||_1$ approximates the fused lasso penalty. In our study, we will use the compression in eq. (2.8) without scaling the averaging operation. Also, we keep the compression operation invertible to make the optimization efficient, as discussed in eq. (1.4) - eq. (1.6).

**2.2 Order-2 Smoothness and Higher-Order Smoothness:** Smoothness of higher 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 *order-2 smoothness*. In this case, the $p \times p$ compression transform $\mathbf{W}$ can be:

$$\mathbf{W} = \mathbf{S}_p^2 = \begin{bmatrix} 1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{S}_{p-1}^1 \end{bmatrix} \cdot \mathbf{S}_p^1 \tag{2.9}$$

where $\mathbf{0}$ is a $(p-1)\times 1$ column vector. By this definition, model coefficients in the compressed domain 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 $\Delta\beta_{i,i+1} = \beta_i - \beta_{i+1}$. In this sense, sparsity of the compressed model coefficients corresponds to order-2 smoothness assumption in the original space. 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* can be defined recursively.

**2.3 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 ordered. The compression operation can be defined as a block matrix to handle the use of different groups of features. For example, suppose features can be divided into three groups. We assume $p_1$ model coefficients on the first group of features satisfy order-1 smoothness, $p_2$ coefficients on the second group of features satisfy order-2 smoothness, and we have no knowledge about the third group of $p_3$ features. In this case, model compression is defined as:

$$\mathbf{W} = \begin{bmatrix} \mathbf{S}_{p_1}^1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{p_2}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{p_3} \end{bmatrix} \tag{2.10}$$

## 3 Model Compression: Energy Compaction

In this section, we discuss another important compressibility assumption, energy compaction in the frequency domain, to compress model coefficients in regularized learning. The energy of many real-world signals can be compacted by transforming signals to a frequency domain where most of their energy is concentrated in a few frequencies, e.g., images are compressed this way [26, 8]. If the target model is applied to classify objects (e.g., images) with compacted energy in a frequency domain, it is reasonable to assume that the model only needs to operate on a few relevant frequencies and thus also has compacted energy in the same frequency domain. Otherwise, most energy of the model is wasted. Naturally, we can include an appropriate compression transform in the $\ell 1$ penalty when learning model coefficients $\boldsymbol{\beta}$ in order to emphasize energy compaction in a frequency domain.

The discrete cosine transform (DCT) is used in the JPEG standard [26], which compresses an object (e.g., an image) by representing it as a sum of cosine functions

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

at various frequencies, and as a result, small coefficients can be discarded. The 2D discrete cosine transform for an $m \times n$ object is:

(3.11)

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

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 [26, 8], but studying sophisticated image codings is not the focus of this paper.

## 4 Model Compression: 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 $\alpha$ omitted) for a problem is a linear function defined on the vocabulary, and each dimension $\beta_j$ indicates the effect of the $j$th word in the decision. In any language, there exists a 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 [17, 29, 16] in a machine learning context.

Given a correlation structure on model coefficients, assuming model sparsity and imposing an $\ell 1$-norm penalty are no longer appropriate. It is easy to understand the problem from the Bayesian perspective. Imposing an $\ell 1$ penalty is equivalent to assuming independent Laplacian priors on model coefficients [11, 24]. But the independence assumption clearly contradicts the existence of a semantic correlation structure. Also, 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.

A simple solution is to decorrelate (i.e., compress) model coefficients before penalization. Given a correlation structure $\boldsymbol{\Sigma}$ (e.g., semantic correlation of words) on coefficients, we set $\mathbf{W} = \boldsymbol{\Sigma}^{-\frac{1}{2}}$ and eq. (1.3) becomes:

$$\min_{\alpha,\boldsymbol{\beta}} L(\mathbf{y}, \mathbf{1}\alpha + \mathbf{X}\boldsymbol{\beta}) + \lambda||\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\beta}||_1 \tag{4.12}$$

Model coefficients in the compressed domain $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\beta}$ are more likely to be sparse since they have a correlation structure $\boldsymbol{\Sigma}' = \boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-\frac{1}{2}} = \mathbf{I}$. From the Bayesian perspective, eq. (4.12) assumes that it is $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\beta}$ that actually follows Laplacian priors on coefficients, and $\boldsymbol{\beta}$ is generated by first sampling $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\beta}$ from the Laplacian priors and then applying a transform $\boldsymbol{\Sigma}^{\frac{1}{2}}$ on the sample. This explains the correlation structure $\boldsymbol{\Sigma}$ on $\boldsymbol{\beta}$.

Optimizing eq. (4.12) leads to sparse coefficients in the compressed domain $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\beta}$. In the original space, the penalty on $\boldsymbol{\beta}$ corresponds to applying a large cost for choosing significantly different coefficients on semantically correlated words. To actually solve eq. (4.12), we follow eq. (1.4) to "decompress" the data space ($\tilde{\mathbf{X}} = \mathbf{X}\mathbf{W}^{-1} = \mathbf{X}\boldsymbol{\Sigma}^{\frac{1}{2}}$) and solve the standard $\ell 1$ regularization in the new space $\tilde{\mathbf{X}}$ as eq. (1.5). It is interesting to note that we actually further correlate the data space ($\tilde{\mathbf{X}} = \mathbf{X}\boldsymbol{\Sigma}^{\frac{1}{2}}$) to decorrelate the model space.

For eq. (4.12) to be useful, the last question is how can we obtain the correlation structure $\boldsymbol{\Sigma}$, e.g., the semantic correlation of words. Indeed, the correlation $\boldsymbol{\Sigma}$ is even harder to estimate than the linear model $\boldsymbol{\beta}$ itself, since the correlation matrix generally has more degrees of freedom. However, in text learning problems, the semantic correlation of words is considered an intrinsic structure of a language and can be learned from other problems on text [17] or even from seemingly irrelevant unlabeled text [29] in the same language.

## 5 Empirical Study: Brain-Computer Interface

In this section, we report our empirical study on brain-computer interface data [2]: classifying single-trial Electroencephalography (EEG) signals. The EEG signals contain important information from human brains. Being able to read and understand those signals is a critical step 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 a univariate time series, the hybrid smoothness assumption introduced in Section 2.3 is suitable for multivariate EEG classifiers.

**Descriptions of the Data Set.** We use data

Table 1: Classification errors on classifying EEG brain signals: means (standard errors) over 50 random runs

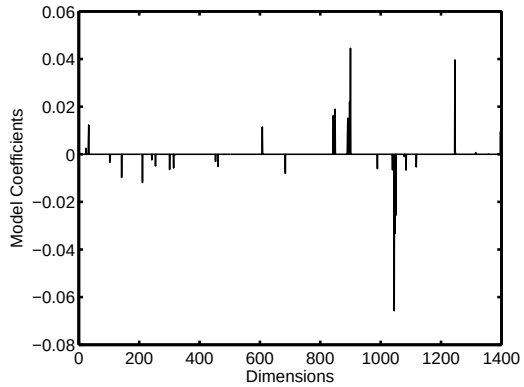
| | means (standard errors) over 50 runs |
|---|---|
| Lasso | 30.22%(0.34%) |
| LassoCP | 25.98%(0.29%) |
| SLgr | 30.00%(0%) |
| SLgrCP | 20.92%(0.16%) |

set IV, self-paced tapping, of BCI Competition 2003 [2], which is a binary classification task. The task 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. The number of features is much larger than the number of training examples, indicating the importance of regularization. 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.

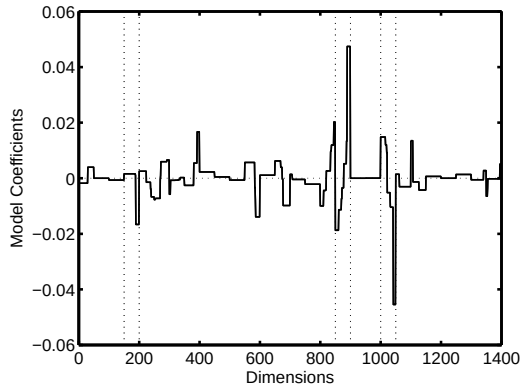
**Experimental Procedures.** The data set contains a fixed training and testing set for competition. For standard $\ell 1$ regularized models, we train lasso and sparse logistic regression. For compressible models, we train compressible lasso, compressible logistic regression, where the compression operation is discuss later in "model and implementation details" part. We learn the four models from the training set and measure their classification errors on the testing set. Note that there is still randomness in the procedure: cross-validation is used to determine the optimal regularization parameter $\lambda$. Therefore, we have 50 random runs on each model.

**Model and Implementation Details.** We consider lasso with labels as $\{+1, -1\}$ (denoted as Lasso in our discussions), sparse logistic regression (denoted as SLgr), compressible lasso (denoted as lassoCP), and compressible logistic regression (denoted as SLgrCP). Following eq. (2.10), the compression operation $\mathbf{W}$ for learning two compressible models is a $1400 \times 1400$ block matrix, with 28 blocks and each block is an order-1 smoothness matrix $\mathbf{S}_{50}^1$ defined as in eq. (2.8). Lasso is implemented using the spgl1 Matlab solver[3], and $\ell 1$-regularized logistic regression is implemented as [15] using lasso. The regularization parameter is chosen from $10^{-7}$ to $10^7$ (step $10^{0.5}$) by 5-fold cross-validation.

**Experimental Results and Analysis.** Classification errors are shown in Table 1, with both means and standard errors (of means) over 50 random runs. By learning compressible models instead of learning sparse models, the testing error is reduced from 30.22% to 25.98% for lasso (Lasso vs. LassoCP), and from 30% to 20.92% for logistic regression (SLgr vs. SLgrCP). During the competition, 15 submissions were received [2]. The best submission achieves 16% error, using features "based on Bereitschaftspotential and event-related desynchronization" [27]. The 2nd best submission achieved 19% using 188 time-based, frequency-based, and correlational features "compiled by hand" [2]. For the other 13 submissions, six attained errors between 23% and 29%, and the other seven were worse. In our study, compressible logistic regression using 1400 raw features are comparable to the two best submissions with domain-specific features.




(a) Sparse LGR coefficients




(b) Compressible LGR coefficients

Figure 1: Model coefficients of sparse and compressible (i.e., piecewise smooth) logistic regression on brain-computer interfacing (EEG signal classification)

We also plot the model coefficients learned by a sparse logistic regression and a compressible logistic regression in Figure 1. From the plot we have several interesting observations. 1) Sparse logistic regression learns sparse coefficients, and compressible LGR leads

[3] http://www.cs.ubc.ca/labs/scl/spgl1/

to (piecewise) smooth coefficients. These two different patterns represent the inductive biases we incorporate into the learning process (via different regularization penalties). 2) Although in the compressible logistic regression we mainly penalize the difference of successive coefficients, most learned coefficients are actually close to zero. The proposed regularization (piecewise local smoothness) effectively controls the model complexity not only in terms of smoothness but also in terms of the norm of coefficients. 3) In the compressible logistic regression, there still exist a few large coefficient jumps over successive dimensions (within the same channel): we plot in Fig. 1b the boundaries (vertical dashed lines) of three selected channels that contain large coefficient jumps. These jumps correspond to large coefficients in the compressed domain (recall that the compressed domain defined by our compression operation is composed of the difference of successive coefficients within the same channel in the original space). The existence of a few large coefficients in the compressed domain is consistent with the notation of compressibility: most information of the original signal is concentrated on a few components after being compressed. Mathematically, this is achieved by performing $\ell 1$ regularization[4] in the compressed domain rather than the original domain.

## 6 Empirical Study: Handwritten Character Recognition

In this section, we study handwritten character recognition on images. As discussed in Section 3, compacted energy in the frequency domain can be used as the inductive bias for regularization, assuming that the model only needs to operate on a few frequencies to classify images.

**Descriptions of the Data Set**. We use the MNIST handwritten digits data set[5], which has 70000 images for 10 digits (from 0 to 9). Images are represented by pixels (in grayscale). The number of features is $p = 784$, corresponding to $28 \times 28$ pixels of an image.

**Experimental Procedures**. 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* in our experiments). As a result, we aim to learn classifiers in a high-dimensional space (784 dimensions) using only a few training examples. 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 target digits in the testing set.

[4]Ideally, $\ell 0$ norm is the best candidate for allowing a few large coefficients, while $\ell 1$ norm is the closest convex relaxation.

[5]http://yann.lecun.com/exdb/mnist/

Table 2: Classification errors over 45 tasks on MNIST, 10 training examples per class: means (and standard errors) over tasks

| | 10 training examples per class |
|---|---|
| Lasso | 9.96% |
| LassoCP | 7.80% |
| SLgr | 9.79% |
| SLgrCP | 7.46% |
| (Lasso - LassoCP) | 2.16%(0.23%) |
| (SLgr - SLgrCP) | 2.33%(0.21%) |

Table 3: Classification errors over 45 tasks on MNIST, 20 training examples per class: means (and standard errors) over tasks

| | 20 training examples per class |
|---|---|
| Lasso | 6.94% |
| LassoCP | 5.30% |
| SLgr | 6.24% |
| SLgrCP | 4.99% |
| (Lasso - LassoCP) | 1.64%(0.18%) |
| (SLgr - SLgrCP) | 1.25%(0.16%) |

Table 4: Classification errors over 45 tasks on MNIST, 50 training examples per class: means (and standard errors) over tasks

| | 50 training examples per class |
|---|---|
| Lasso | 4.91% |
| LassoCP | 3.45% |
| SLgr | 3.91% |
| SLgrCP | 3.26% |
| (Lasso - LassoCP) | 1.46%(0.12%) |
| (SLgr - SLgrCP) | 0.65%(0.09%) |

**Model and Implementation Details**. The standard $\ell 1$ regularized models: lasso and sparse logistic regression are the same as in Section 5. The model compression operation used for learning compressible lasso and compressible logistic regression is the DCT operation in eq. (3.11). By using a DCT operation in the $\ell 1$ penalty, we impose the assumption that model coefficients should be sparse in the DCT frequency domain, implying that the model only needs to operate in a few frequencies. Others implementation details are the same as in Section 5.

**Experimental Results and Analysis.** Empirical results are shown in Table 2–Table 7 and Figure 2.

Table 2 has two parts. The first part shows average classification errors over 45 tasks of lasso (Lasso),

Table 5: Performance comparison on individual tasks between compressible and sparse models on MNIST, 10 training examples per class: #win/#loss over 45 tasks

| | 10 training examples per class |
|---|---|
| LassoCP vs. Lasso | 41/4 |
| SLgrCP vs. SLgr | 43/2 |

Table 6: Performance comparison on individual tasks between compressible and sparse models on MNIST, 20 training examples per class: #win/#loss over 45 tasks

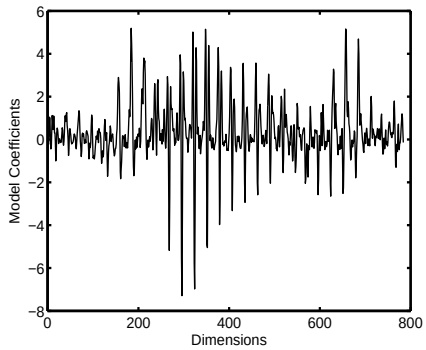
| | 20 training examples per class |
|---|---|
| LassoCP vs. Lasso | 42/3 |
| SLgrCP vs. SLgr | 42/3 |

Table 7: Performance comparison on individual tasks between compressible and sparse models on MNIST, 50 training examples per class: #win/#loss over 45 tasks

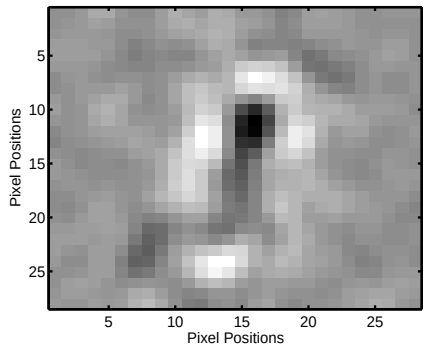
| | 50 training examples per class |
|---|---|
| LassoCP vs. Lasso | 44/1 |
| SLgrCP vs. SLgr | 40/5 |

compressible Lasso (LassoCP), sparse logistic regression (SLgr) and compressible logistic regression (SLgrCP), where models are learned using 10 training examples per class in each task. We omit standard errors (of classification errors) over 45 tasks since they correspond to the variation of the difficulty of different tasks, which is not of interest. The second part (the last two rows) of Table 2 are means (and also standard errors) over 45 tasks for the *paired difference* of classification errors between a sparse and a compressible model. Table 3 and Table 4 show similar results as in Table 2, with 20 and 50 training examples per class in each task, respectively. Table 2–Table 4 show that compressible lasso and compressible logistic regression generally outperform their sparse counterparts. The less the training examples available for learning, the more obvious the advantage of compressible models over sparse models. We also compare classification performance of compressible and sparse models on each individual task in Table 5–Table 7. The results shows that the majority of tasks benefit from learning compressible models.

In addition, we plot in Figure 2 the model coefficients of a compressible logistic regression from a random run, where the task is to classify "1" (negative class) and "8" (positive class). As shown in Fig. 2a, model coefficients in the original space (which correspond to image pixels) are mainly changing in a few frequencies and indicate that the model is sparse in the compressed (DCT) space, i.e., has compacted energy in the frequency domain. Interestingly, when we plot the model coefficients as a $28 \times 28$ image in Fig. 2b, the difference between the negative class "1" and the positive class "8" is well emphasized. The model coefficients, although sparse in the frequency domain induced by the DCT operation, are not sparse in the original pixel space and can represent meaningful patterns for a classification task. This should not be a surprise: DCT is the compression operation in the JPEG standard for compressing images, and information in an image tends to be sparse in the DCT domain.




(a) Compressible model coefficients as a 784 dimensional vector (intercept $\alpha$ omitted)




(b) Compressible model coefficients as a $28 \times 28$ image (the intercept $\alpha$ omitted)

Figure 2: Model coefficients of a compressible logistic regression on MNIST. Task: classifying "1" vs. "8".

## 7 Empirical Study: Text Classification

In this section, we study text classification. As mentioned in Section 4, we include a decorrelation transform $\mathbf{W} = \mathbf{\Sigma}^{-\frac{1}{2}}$ as the compression operation. The semantic word correlation $\mathbf{\Sigma}$ is estimated from unlabeled text [29], and thus learning compressible models offers an approach for semi-supervised learning.

Table 8: Classification errors over 190 tasks, 2% documents in $D_{tr}$ for training: means (and standard errors)

| | 2% sampling from $D_{tr}$ |
|---|---|
| Lasso | 22.17% |
| ElasNet | 19.97% |
| LassoCP | 11.13% |
| SLgr | 21.69% |
| SLgrCP | 9.31% |
| (Lasso - LassoCP) | 11.05%(0.22%) |
| (ElasNet - LassoCP) | 8.84%(0.28%) |
| (SLgr - SLgrCP) | 12.38%(0.30%) |

Table 9: Classification errors over 190 tasks, 5% documents in $D_{tr}$ for training: means (and standard errors)

| | 5% sampling from $D_{tr}$ |
|---|---|
| Lasso | 17.02% |
| ElasNet | 12.87% |
| LassoCP | 7.76% |
| SLgr | 15.28% |
| SLgrCP | 6.19% |
| (Lasso - LassoCP) | 9.26%(0.21%) |
| (ElasNet - LassoCP) | 5.11%(0.18%) |
| (SLgr - SLgrCP) | 9.09%(0.22%) |

**Descriptions of the Data set.** We use the 20-Newsgroups data set[6]. It contains 11314 training and 7532 testing documents from 20 newsgroups. We denote training and testing sets as $D_{tr}$ and $D_{ts}$, respectively. Documents are represented as bags of words. We select the vocabulary to include the most frequent 200 words in each newsgroups except the 20 most frequent common words across all newsgroups. This leads to $p = 1443$ features (words) in the vocabulary.

**Experimental Procedures.** Documents are from 20 newsgroups, so we construct 190 binary classification tasks, each to classify a pair of two selected newsgroups. For each task, we randomly sample 2% or 5% of the relevant documents in $D_{tr}$ as the labeled training examples. Two newsgroups 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 documents for each task are fixed to be all relevant documents in $D_{ts}$.

**Model and Implementation Details.** Sparse lasso and sparse logistic regression are the same as in previous experiments. For compressible models, decorrelation is used for compression, as eq. (4.12). The correlation $\mathbf{\Sigma}$ is estimated using all the documents in $D_{tr}$ as unlabeled data, as in [29], and then is used to specify the decorrelation operation in all 190 tasks. In addition to two sparse models and two compressible models, we also test elastic net [33]. 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 regularization by either norm[7]. For all models, the intercept $\alpha$ is added into the penalty term, which slightly improves the performance. This is possibly because $\alpha$ tends to overfit the imbalanced class distribution in training examples. For lasso and logistic regression, the regularization parameter for the $\ell 1$ norm is chosen from $10^{-7}$ to $10^{7}$ with a larger step $10^{1}$ (for computation efficiency). Other details are the same as Section 5. For elastic net, the $\ell 1$ norm bound is choose in the same way, and the second parameter $\lambda_2$ [33] is chosen from $10^{-4}$ to $10^{4}$ with step $10^{1}$.

Table 10: Performance comparison on individual tasks: #win/#loss over 190 tasks

| | 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 |

**Experimental Results and Analysis.** Results are shown in Table 8, Table 9 and Table 10.

The first part of Table 8 contains the means of classification errors over 190 different tasks, using lasso, elastic net, compressible lasso, sparse logistic regression, compressible logistic regression, respectively. The second part includes means and standard errors of the paired difference of classification errors on each task between competitive models. For regression-based models (i.e., lasso, compressible lasso and elastic net), elastic net improves standard lasso by using a convex combination of $\ell 1$ and $\ell 2$ penalty. Elastic net is designed to address correlated predictors [33], and its success confirms that there exists correlations among model coefficients, corresponding to the semantic correlation of words. Further, compressible lasso shows significant improvements over both lasso and elastic net. Compressible lasso is superior to elastic net, because it explicitly includes additional information (i.e., semantic correlation of words) from unlabeled text in the form of compression operation in regularization. For logistic regression, compressible logistic regression also show notable improvements over sparse logistic regression.

[6] http://people.csail.mit.edu/jrennie/20newsgroups

[7] Elastic net has two regularization parameters controlling $\ell 1$ and $\ell 2$ norm, respectively. With cross-validation to determine both parameters, elastic net includes $\ell 1$ regularization and $\ell 2$ regularization as two specific cases.

Table 9 reports similar results as in Table 8, where we sample 5% documents (instead of 2%) from $D_{tr}$ as the training data in each random run. Finally, Table 10 compares model performance on individual tasks. Compressible models dominate other models in almost all 190 tasks, which shows the significance of the results and indicates that learning compressible models is very effective and reliable for text classification problems.

## 8 Related Work and Discussion

Compressive sampling [5] or compressed sensing [9] was recently developed for signal acquisition, and has received considerable attention [1]. According to this theory, one can successfully acquire a signal (e.g., an image) from many fewer measurements than required by Nyquist-Shannon sampling theory. The key is to assume that the true signal is compressible, i.e., sparse in a compressed domain. Under this assumption, signal acquisition given a few linear measurements on the true signal $\boldsymbol{\beta}^*$ can be achieved by solving the problem:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & ||\mathbf{W}\boldsymbol{\beta}||_1 \\ \text{subject to} \quad & \boldsymbol{\Phi}\boldsymbol{\beta} = \mathbf{y} \end{aligned} \tag{8.13}$$

Here $\boldsymbol{\beta}$ is a candidate signal. $\mathbf{W}$ is a known compression operation. The matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n]^T$ is the sensing (or projection) matrix [6, 9], whose rows correspond to measuring or sensing operations conducted on the true signal $\boldsymbol{\beta}^*$. A common choice of $\boldsymbol{\Phi}$ is random projections. The vector $\mathbf{y}$ contains projections of the true signal $\boldsymbol{\beta}^*$ on predefined bases. Recently, compressive sampling has drawn considerable attention from machine learning and data mining communities. In [13], researchers suggest using sparse Bayesian regression and active learning to solve the CS problem in eq. (8.13) and adaptively "learn" the optimal projection matrix $\boldsymbol{\Phi}$. Authors in [4, 31] consider classification and regression problems, respectively, where data are compressed and not directly observable, e.g., only random projections of the data can be accessed. This paper is another application of compressive sampling theory in a machine learning context: model coefficients can be compressed before being penalized, and model sparsity is only required in a compressed domain rather than the original space. The focus of this paper to study the inclusion of different compression operations in $\ell 1$ regularization, which incorporates additional information and imposes more appropriate inductive bias in the learning process.

Researchers have proposed various improvements based on $\ell 1$ regularization and learning sparse models. Fused lasso [22] includes a penalty on the absolute difference of successive coefficients, which, as shown in Section 2.1, can be approximated by a compression operation in the $\ell 1$ penalty. Elastic net [33] combines $\ell 1$ and $\ell 2$ norms to address the issue of correlated coefficients, which is also the focus of OSCAR [3]. Group lasso [28] adds more restrictions on model sparsity: variables in the same group tend to be eliminated together. Structured sparsity [12] generalizes group lasso and studies the case that we have additional structured constraints on model sparsity, i.e., not all sparse patterns are equally likely and we prefer some of them to others. In this paper, we generalize model sparsity from another perspective: we *relax* the sparsity assumption by including compression into $\ell 1$ regularization. As a result, model coefficients are only assumed to be sparse after compression, which can be a more appropriate inductive bias when model sparsity is too restrictive.

## 9 Conclusion and Future Work

By including a compression operation into $\ell 1$ regularized learning, model coefficients are compressed before being penalized and sparsity is achieved in a compressed domain. This relaxes the assumption on model sparsity to compressibility, and provides an opportunity encode more appropriate inductive biases, e.g., piecewise local smoothness, compacted energy in the frequency domain, and semantic correlation. We conduct extensive experiments on brain-computer interfacing, 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 further demonstrate the advantages of learning compressible models instead of sparse models.

Future work will explore the possibility of combining compression with other penalty norms, e.g., $\ell 0$ or $\ell 2$ norm. The $\ell 2$ norm is easy to optimize and widely used in regularization. However, the notion of compressibility generally implies that most energy of the signal will concentrate on a few components after compression. This may contradict the use of $\ell 2$ norm, which will heavily penalize large coefficients (in the compressed domain). For example, most information of the compressible model in Fig. 2a is concentrated around a few frequencies, and thus we expect to have some large coefficients in the frequency domain[8]. On the other hand, $\ell 0$ norm is a good candidate for imposing compressibility assumptions, but $\ell 0$ regularization is combinatorial in nature and difficult to solve.

[8] Some specific compression operation may work well under $\ell 2$ norm penalty. For example, the decorrelation transform used in eq. (4.12), when combined with $\ell 2$ regularization, corresponds to a Gaussian prior based on a correlation structure [17, 29].

Another direction is to automate the design of the compression operation in learning compressible models. Although optimal compression is undecidable [10], it is possible to infer an effective (but not necessarily optimal) compression transform from auxiliary information (e.g., related tasks), to select a compression operation from a finite set, or to adaptively adjust a given compression transform. For example, we can use reweighted $\ell 1$ minimization [7] and adaptive lasso [32] in the compressed domain to improve the compression matrix.

**Acknowledgement** This work was funded in part by the National Science Foundation under grant number NSF-IIS0911032 and the Department of Energy under grant number DESC0002607.

## References

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

[2] 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.

[3] H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64:115–123, 2008.

[4] R. Calderbank, S. Jafarpour, and R. Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Preprint, 2009.

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

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

[7] E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905, 2008.

[8] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG2000 Still Image Coding System: An Overview. *IEEE Trans. Consumer Electronics*, 2000.

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

[10] C. Faloutsos and V. Megalooikonomou. On data mining, compression, and kolmogorov complexity. *Data Min. Knowl. Discov.*, 15(1):3–20, 2007.

[11] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2001.

[12] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *ICML '09*, 2009.

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

[14] 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.

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

[16] 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.

[17] R. Raina, A. Y. Ng, and D. Koller. Constructing Informative Priors using Transfer Learning. In *ICML*, pages 713–720, 2006.

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

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

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

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

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

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

[24] P. Trevor and C. George. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[25] J. A. Tropp. Just relax: convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3), 2006.

[26] G. K. Wallace. The JPEG Still Picture Compression Standard. *IEEE Trans. Consumer Electronics*, 1992.

[27] 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.

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

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

[30] P. Zhao and B. Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006.

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

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

[33] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320, 2005.