Conditional Sparse Coding and Multiple Regression for Grouped Data

Min Xu
Machine Learning Department
Carnegie Mellon University
minx@cs.cmu.edu

John Lafferty
Department of Statistics
University of Chicago
lafferty@galton.uchicago.edu

May 26, 2012

Abstract

We study the problem of multivariate regression where the data are naturally grouped, and a regression matrix is to be estimated for each group. We propose an approach in which a dictionary of low rank parameter matrices is estimated across groups, and a sparse linear combination of the dictionary elements is estimated to form a model within each group. We refer to the method as *conditional sparse coding* since it is a coding procedure for the response vectors Y conditioned on the covariate vectors X. This approach captures the shared information across the groups while adapting to the structure within each group. It exploits the same intuition behind sparse coding that has been successfully developed in computer vision and computational neuroscience. We propose an algorithm for conditional sparse coding, analyze its theoretical properties in terms of predictive accuracy, and present the results of simulation as well as equities and brain imaging experiments that compare the new technique to reduced rank regression.

Contents

1 Introduction

Sparse coding is an approach to approximating a collection of signals by sparse linear combinations of a codewords chosen from a shared, learned dictionary. The method was proposed by [23] for encoding natural images, with the motivation of developing a simple computational model of neural coding in the visual cortex. Through the use of sparsity and a large learned dictionary of codewords, sparse coding is able to efficiently capture a rich collection of features that are common to a population of signals. Variants of sparse coding have enjoyed considerable success in computer vision [8, 15, 18, 25, 27, 5].

In this paper we apply the intuition behind sparse coding to design a new procedure for multivariate regression with data that fall into possibly overlapping groups or tasks. In traditional multivariate regression, the data consist of a set of response vectors $Y \in \mathbb{R}^q$, and for each Y, a corresponding covariate vector $X \in \mathbb{R}^p$. In a vector autoregressive time series model, for instance, $Y = Z_t$ is a vector at time t, and $X = Z_{t-1}$ is the vector at the previous step. In predicting brain activation patterns in neuroscience, Y might be the neural activations in the regions of the brain with X as outside stimuli. In a linear model, $Y = BX + \epsilon$, where $B \in \mathbb{R}^{q\times p}$ is the matrix of parameters and $\epsilon \in \mathbb{R}^q$ is a random, mean zero error.

In many applications, the data naturally occur in groups or tasks, and assuming the same model $Y = BX + \epsilon$ for each group may be unjustified. For instance, in a non-stationary time series, the distribution of $Y = Z_t$ varies over time. In the neuroscience example, different people may have different neuronal activation patterns. In both cases it may be natural to place the data into possibly overlapping groups. More generally, the groups could be determined by any factor in the data or experimental design.

In high-dimensional settings where p, q are large, the number of parameters in B maybe be too large to estimate accurately from limited data. One approach toward estimating reduced complexity models is to perform a least squares regression with a nuclear norm constraint on the coefficient matrix B; the nuclear norm serves as a convex surrogate for low rank constraints [26, 22]. For grouped data, a different model could be estimated for each group using this approach; however, carrying out separate regressions ignores commonality between the groups, and worsens the problem of limited data.

Our approach is to estimate the parameter matrices as

$$\widehat{B}^{(g)} = \sum_{k=1}^{K} \alpha_k^{(g)} D_k$$

where each dictionary entry D_k is a low rank matrix, and $\alpha^{(g)} = (\alpha_1^{(g)}, \dots, \alpha_K^{(g)})$ is a sparse vector; both $\{D_k\}$ and $\{\alpha^{(g)}\}$ are learned from data. The coefficients $\alpha_k^{(g)}$ are estimated for each group g, but the "codewords" or "dictionary elements" D_k are shared across groups. This exploits the same intuition behind sparse coding for image analysis. Sparsity allows the dictionary entries D_k to specialize and capture predictive aspects of the data shared by many groups, while the coefficients $\alpha^{(g)}$ tailor the model to the specific group g. Allowing the size K of the dictionary to be large enables a rich class of parameter matrices to be modeled, while a low rank condition on the individual codeword matrices D_k allows them to be estimated from limited data.

We perform both a "pessimistic" and "optimistic" analysis of our method. In the pessimistic analysis, the model may not be correct; that is, we do not assume any underlying common structure among the the groups. In this case the model cannot achieve lower risk than the alternative of separate low rank regressions within each group. However, our analysis shows that the method suffers little excess risk relative to separate regressions. In the optimistic analysis, when the learned dictionary has captured common structure between the groups, the method produces an accurate estimator with much lower sample complexity than required by low rank regression. In both analyses, we measure statistical accuracy through non-asymptotic bounds on the excess risk $R(D, \alpha^{(g)}) - R(B^*)$. We show that the new procedure is effective and practical with experiments on simulated data and stock data, reported in Section 6.

2 Related Work

[17] have studied a different way of using dictionary learning for supervised tasks; in this approach one first encodes data X and then uses the encoding to perform classification or regression. Our work is more related to multi-task learning [6, 9] and is in particular a generalization of a model by [1]. They require that all $\alpha^{(g)}$ have the same sparsity pattern, so that all groups use the same small subset of dictionary elements. By allowing different groups to use different subsets of the dictionary, our model is much more flexible, though at the cost of requiring a non-convex optimization. [14] used mixed integer programming to generalize the model of [1] although our formulation is still more flexible and our optimization simpler. Approach of [16] could be adapted to our setting, although their notion of task-relatedness is very different from ours.

Existing approaches toward a theoretical analysis of multi-task learning differ significantly from our analysis by focusing on PAC-learnability with respect to a more abstract notion of task-relatedness [19, 4].

Theoretical analysis of sparse coding is rather limited. Some work studies the generalization error of dictionary learning [24, 20] and the local correctness of the non-convex objective for dictionary learning [11]. [13] consider sparse approximability and prove an information theoretic lower bound on sparse approximability of general p-dimensional vectors. They further show, non-constructively, that the lower bound can be achieved via an optimally constructed dictionary. We instead consider sparse approximability of a variety of structured spaces with respect to a dictionary that could plausibly be learned by a practical procedure.

3 Problem Formulation

In this work we focus attention on cases where the data are naturally grouped. Suppose we have G groups, indexed by $g = 1, \ldots, G$. Let $X_i^{(g)} \in \mathbb{R}^p, Y_i^{(g)} \in \mathbb{R}^q$ denote the explanatory and response variables for the ith sample in group g. For each group, we let $B^{*(g)} = \arg\min_{B^{(g)}} R(B^{(g)})$ be the oracle regression matrix where we define

$$R(B^{(g)}) = \mathbb{E}_{X^{(g)},Y^{(g)}} \|Y^{(g)} - B^{(g)} X^{(g)}\|_F^2$$

For convenience, we will assume the sample size n is the same for all groups, noting that more generally it will vary with g. Let $X^{(g)} = (X_1^{(g)}, \ldots, X_n^{(g)}) \in \mathbb{R}^{p\times n}$ and $Y^{(g)} = (Y_1^{(g)}, \ldots, Y_n^{(g)}) \in \mathbb{R}^{q\times n}$, with the n samples of group g arranged as matrix columns.

Our goal is to estimate $B^{*(g)}$ with empirical estimates $\widehat{B}^{(g)}$. We will consider estimates of the form $\widehat{B}^{(g)} = \sum_{k=1}^K \widehat{\alpha}_k^{(g)} D_k$ where each D_k is a low rank matrix, and $\widehat{\alpha}^{(g)} = (\widehat{\alpha}_1^{(g)}, \ldots, \widehat{\alpha}_K^{(g)})$ is an estimated sparse vector. The codewords, or dictionary entries D_k are themselves estimated from data using nuclear norm regularization from data pooled across groups, as described in Section 4.

4 Conditional Sparse Coding

The basic idea underlying conditional sparse coding is to learn a collection of low rank matrices $\{D_1, ..., D_K\}$ (a dictionary) and estimate $\widehat{B}^{(g)}$ as a sparse linear combination of the dictionary entries. We optimize the overall objective function $f(\alpha, D)$ defined by

$$f(\alpha, D) = \quad \frac{1}{G}\sum_{g=1}^{G}\left\{\frac{1}{n}\left\|Y^{(g)} - \Big(\sum_{k=1}^{K}\alpha_k^{(g)}D_k\Big)X^{(g)}\right\|_F^2 + \lambda\|\alpha^{(g)}\|_1\right\}$$

where the optimization $\min_\alpha \min_{D\in\mathcal{C}_D(\tau)} f(\alpha, D)$ is carried out over the set

$$\mathcal{C}_D(\tau) = \left\{D \in \mathbb{R}^{q\times p} : \|D\|_* \leq \tau \text{ and } \|D\|_2 \leq 1\right\}.$$

The ℓ_1 norm penalty induces sparsity on the α vectors and the nuclear-norm restriction forces the matrices D_k to be low rank. The spectral norm constraint ensures no particular dictionary entry can be too large, and serves as an identifiability constraint; a similar constraint in sparse coding requires that all dictionary vectors must have norm no larger than one.

The objective function is biconvex but not jointly convex in α and D. Thus, we follow the standard sparse coding approach and perform block-coordinate descent by alternately optimizing over $\{\alpha^{(g)}\}$ with fixed $\{D_k\}$, and optimizing over $\{D_k\}$ with fixed $\{\alpha^{(g)}\}$. We refer to the algorithm as *conditional sparse coding* (CSC) since it is a coding procedure for the response vectors Y conditioned on the covariate vectors X.

Algorithm 1 Conditional Sparse Coding (CSC)

```
Input: Data $\{(Y^{(g)}, X^{(g)}\}_{g=1,\ldots,G}$, regularization parameters $\lambda$ and $\tau$.
 1. Initialize dictionary $\{D_1, ..., D_K\}$ as random rank one matrices.
 2. Alternate between the following steps until convergence of $f(\alpha, D)$:
     a. Encoding step: $\{\alpha^{(g)}\} \leftarrow \arg\min_{\alpha^{(g)}} f(\alpha, D)$
     b. Learning step:
        $\{D_k\} \leftarrow \arg\min_{D_k\in\mathcal{C}_D(\tau)} f(\alpha, D)$
```

The encoding step is equivalent to an independent ℓ_1-constrained least squares fit, or lasso optimization, for each group g:

$$\min_{\alpha^{(g)}\in\mathbb{R}^K} \frac{1}{n}\sum_{i=1}^{n}\left\|Y_i^{(g)} - \sum_{g=1}^{G}\alpha_k^{(g)}(D_k X_i^{(g)})\right\|_2^2 + \lambda\|\alpha^{(g)}\|_1. \tag{4.1}$$

A variety of algorithms are available to solve the lasso efficiently, notably iterative soft thresholding, a form of coordinate descent [10].

For optimizing the dictionary entries, we propose a projected gradient descent algorithm. A complication is that since the constraint set $\mathcal{C}_D(\lambda)$ is an intersection of nuclear norm and spectral norm balls, the projection needs to be done with care.

In this algorithm, Q_L is defined by

$$Q_L(D', D) = \quad f(\alpha, D) + \sum_{k=1}^{K}\langle D'_k - D_k, \nabla_k\rangle + \frac{L}{2}\sum_{k=1}^{K}\|D'_k - D_k\|_F^2.$$

The $\texttt{SimulProject}(\Sigma, \tau)$ function performs a simultaneous projection of the diagonal of the matrix Σ onto the intersection of the ℓ_1-ball of radius τ and the ℓ_∞-ball of radius one. The details of this projection are described in algorithm 4.

Algorithm 2 Projected Gradient Descent for Dictionary Learning

```
Input: $Y^{(g)} \in \mathbb{R}^{q \times n}, X^{(g)} \in \mathbb{R}^{p \times n}$, and $\alpha^{(g)} \in \mathbb{R}^K$ for $g = 1, \ldots, G$; $\tau \in \mathbb{R}$, $\gamma > 1$.
Output: $D_1, ..., D_K \in \mathbb{R}^{q \times p}$
1. For $k = 1, ..., K$, generate random unit vectors $\overline{u}, \underline{u}$, and set $D_k = \overline{u}\underline{u}^{\mathsf{T}}$. Set $L = 1$.
2. Iterate until convergence:
   Precompute
       $$\nabla^{(g)}_{all} = \frac{1}{n} Y^{(g)} - \frac{1}{n} \sum_{k=1}^{K} D_k \left( \alpha^{(g)}_k X^{(g)} \right)$$
   (a) For each $k$, compute the gradient $\nabla_k = -\frac{1}{G} \sum_{g=1}^{G} \nabla_{all} \left( \alpha^{(g)}_k X^{(g)} \right)^{\mathsf{T}}$
   (b) For each $k$:
       compute the SVD $D_k - \frac{1}{L} \nabla_k = U_k \Sigma_k V_k^{\mathsf{T}}$;
       project $\Sigma'_k = \texttt{SimulProject}(\Sigma_k, \tau)$;
       update $D'_k = U_k \Sigma'_k V_k^{\mathsf{T}}$.
   (c) If $f(\alpha, D') > Q_L(D', D)$, set $L \leftarrow \gamma L$ and repeat from (b).
   (d) Set $D \leftarrow D'$.
```

4.1 FISTA

The Fast Iterative Shrinkage and Thresholding Algorithm, based on Nesterov's optimal first-order optimization algorithm, proposed by [3], is generally faster than the projected gradient descent algorithm.

FISTA is not guaranteed to improve the objective at every iteration and this can lead to instability and divergence. We adapt the monotonic version of FISTA described in [2] for dictionary learning.

4.2 `SimulProject`

Proof of correctness of Algorithm 4. Let $B_1(\tau)$ denote the l_1-ball of radius τ and let $B_\infty(1)$ denote the l_∞-ball of radius 1.

We first note that the instructions given in computing λ in step 2 is correct as shown in [7].

We proceed by induction on the dimensionality of the input vector. If $v \in \mathbb{R}$, then clearly the projection is just $v^{new} = \min(\tau, 1)$ and would be outputted by either step 3 or 4 of the algorithm.

Suppose then we have a vector of dimension p. We now perform case analysis:

In case 1, step 3 terminates. In this case, we projected to l_1-ball and have also landed in the l_∞-ball. We claim that v^{new} is the correct projection onto the intersection. Let $u \in B_1(\tau) \cap B_\infty(1)$, by definition, $\|v - v^{new}\|_2 \leq \|u - v\|_2$. Since $v^{new} \in B_\infty(1)$ as well, we get that v^{new} is the correct projection.

In case 2, step 4 terminates. In this case, we projected to the l_∞-ball and have also landed in the l_1-ball. By same argument as before, v^{new} is the correct projection.

Algorithm 3 Monotonic FISTA for Dictionary Learning

```
Input: $Y^{(g)} \in \mathbb{R}^{q \times n}, X^{(g)} \in \mathbb{R}^{p \times n}$, and $\alpha^{(g)} \in \mathbb{R}^K$ for $g = 1, \dots, G$; $\tau \in \mathbb{R}$, $\gamma > 1$.
Output: $D_1, ..., D_K \in \mathbb{R}^{q \times p}$
1. For $k = 1, ..., K$, generate random unit vectors $\overline{u}, \underline{u}$, and set $D_{k,1} = \overline{u}\underline{u}^\mathsf{T}$. Set $L = 1$.
   Set $D'_{k,1} = D_{k,1}$, set $c_1 = 1$
2. Iterate $t = 1, ..., T$
   Precompute
       $\nabla^{(g)}_{all} = \frac{1}{n} Y^{(g)} - \frac{1}{n} \sum_{k=1}^{K} D'_{k,t} \left( \alpha_k^{(g)} X^{(g)} \right)$
   (a) For each $k$, compute the gradient $\nabla_k = -\frac{1}{G} \sum_{g=1}^{G} \nabla^{(g)}_{all} \left( \alpha_k^{(g)} X^{(g)} \right)^\mathsf{T}$
   (b) For each $k$:
       compute the SVD $D'_{k,t} - \frac{1}{L} \nabla_k = U_k \Sigma_k V_k^\mathsf{T}$;
       project $\Sigma'_k = \texttt{SimulProject}(\Sigma_k, \tau)$;
       update $D''_{k,t} = U_k \Sigma'_k V_k^\mathsf{T}$.
   (c) If $f(\alpha, D''_{k,t}) > Q_L(D''_{k,t}, D'_{k,t})$, set $L \leftarrow \gamma L$ and repeat from (b).
   (d) Set $c_{t+1} = \frac{1+\sqrt{1+4c_t^2}}{2}$
   (e) For each $k$:
       Set $D_{k,t} = \arg\min\{f(\alpha, D) : D = D'_{k,t}, D_{k,t-1}\}$
       Set $D'_{k,t+1} = D_{k,t} + \frac{c_t}{c_{t+1}}(D''_{k,t} - D_{k,t}) + \frac{c_t - 1}{c_{t+1}}(D_{k,t} - D_{k,t-1})$
```

Algorithm 4 `SimulProject`, simultaneous projection onto l_1-ball of radius τ and l_∞-ball of radius 1

```
• IN: $\Sigma$ diagonal matrix, $\tau > 0$
• OUT: $\Sigma'$
1. Let $v \in \mathbb{R}^p$ be the diagonal of $\Sigma$, suppose without loss of generality that $v_1 \geq v_2 \geq ...v_p$.
2. Compute $\lambda \geq 0$, with the following steps, such that soft-thresholding by $\lambda$ projects $v$ onto $l_1$-ball
   of radius $\tau$
   (a) let $k = \max \left\{ j = 1, ..., p : v_j - \frac{1}{j} \left( \sum_{r=1}^{j} u_r - \tau \right) > 0 \right\}$
   (b) Let $\lambda = \frac{1}{k} \left( \sum_{i=1}^{j} v_j - \tau \right)$
3. Soft-threshold $v$ by $\lambda$ to get $v^{new}$. If $v_1^{new} \leq 1$, set diagonal of $\Sigma'$ as $v^{new}$ and return.
4. Set all $v_i \geq 1$ to be 1 to get $v^{new}$. If $\sum_i v_i^{new} \leq \tau$, set diagonal of $\Sigma'$ as $v^{new}$ and return.
5. Set $v_1$ as 1. Let $v' = (v_2, ..., v_n)$. Recursively call $\texttt{SimulProject}(v', \tau - 1)$
```

In case 3: we go into step 5. We must now project to boundary of both the l_∞-ball and l_1-ball. Thus, we need to find a v^{new} to minimize $||v - v^{new}||_2^2$ and such that BOTH $\max_i v_i^{new} = 1$ and $\sum_i v_i^{new} = \tau$.

v_1^{new} must equal 1 since $v_1 = \max_i v_i$. The remaining v' must then both be in the l_1-ball of radius $\tau - 1$ and be in the l_∞-ball of 1. The correctness of the algorithm then follows by inductive hypothesis. □

4.3 Remarks on Implementation details

Although learning the dictionary is computationally intensive, fitting the coefficients to the dictionary is very fast due to efficient lasso optimization algorithms. Thus, an easy way to speed up CSC is to learn the dictionary with a smaller number of groups. The CSC optimization, being non-convex, is sensitive to initialization. We suggest random initialization both because our theoretical guarantees assume random initialization and because it works well in practice.

In sparse coding, one never picks a dictionary size K equal to or greater than number of vectors to encode to avoid the trivial solution of letting each vector be a dictionary element itself. In CSC however, one can choose $K > G$ because of the nuclear-norm constraint on the dictionary entries. Based both on theory and experimental results, We recommend, when picking tuning parameters τ, λ, that τ be held to a constant between 1 and 0.5 and λ be then chosen with cross-validation.

5 Theoretical Analysis

To get a more complete understanding of CSC, we perform both a pessimistic analysis and an optimistic analysis. In the pessimistic analysis, we do not assume that our model is correct, that is, we do not assume any underlying common structure among the the groups. It is obvious that, under the general pessimistic setting, we cannot achieve higher statistical accuracy with CSC than with the alternative of estimating separate low-rank matrices for each group. Our pessimistic analysis provides a simple rule for determining, in the worst case, how much worse CSC is than the alternative.

In the optimistic analysis, we focus on a very specific setting where we only have to fit the coefficients to a pre-existing set of learned dictionary entries. We assume that the learned dictionary has thus captured common structure that exists among the groups. We then show that in this setting CSC can produce an accurate estimator with fewer samples than the alternative of estimating separate matrices.

In all of our analyses, we measure statistical accuracy through non-asymptotic bounds on the excess risk $R(D, \alpha^{(g)}) - R(B^*)$. For clarify of presentation, we will use same symbols c and C to represent possibly different, generic constants in the theorem statements.

Before we begin our analysis, we first enumerate and justify the underlying assumptions.

A1. For all groups g, $X^{(g)}$ and $Y^{(g)}$ are zero mean Gaussian random vectors. Let Σ be the $(p+q) \times (p+q)$ covariance matrix $\Sigma = \mathbb{E}[(X^{(g)}, Y^{(g)})(X^{(g)}, Y^{(g)})^\mathsf{T}]$. Then the spectral norm $\|\Sigma\|_2$ is a constant independent of n.

A2. For all groups g, $\|B^{*(g)}\|_* \leq L$ and $B^{*(g)}$ is of rank at most r.

A3. The sample size satisfies $n \geq (p+q)$.

We make assumption A1 only to leverage results on concentration of measure; we do not use any other properties of the Gaussian distribution. Our analysis will thus easily extend to subgaussian random vectors. Assumption A2 is merely notation, allowing us to state our bounds in term of L and r. Assumption A3 is made so that many of results in our pessimistic analysis can be stated more compactly; we do not make this assumption in our optimistic analysis.

It should be emphasized that since we are carrying out an excess risk analysis, we do not require incoherence conditions on our samples $X_1^{(g)}, \ldots, X_n^{(g)}$, as are often assumed in high-dimensional statistical analysis.

Because we will repeatedly compare the excess risk rate of CSC against estimating separate matrices, we first prove an excess risk bound on for using nuclear-norm regularization in each group.

Theorem 5.1. *Suppose that assumptions A1, A2, A3 hold. Let*

$$\widehat{B}^{(g)} = \underset{\{B \,:\, \|B\|_* \leq L\}}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \|Y_i^{(g)} - BX_i^{(g)}\|_2^2.$$

Then with probability at least $1 - \exp(-cp)$*, we have that*

$$\max_{g=1,\ldots,G} R(\widehat{B}^{(g)}) - R(B^{*(g)}) \leq CL^2 \sqrt{\frac{(p+q)\log(nG)}{n}}$$

where c, C *are constants depending only on* $\|\Sigma\|_2$ *as defined in A1.*

We provide proof sketches of all theorems in Section 5.3.

5.1 Pessimistic Analysis

Let $D^{\text{learn}}, \alpha_\lambda^{learn(g)}$ be the dictionary and coefficients outputted by Conditional Sparse Coding. The results of this section establish bounds on the excess risk $R(D^{\text{learn}}, \alpha_\lambda^{\text{learn}(g)}) - R(B^{*(g)})$. We stress that we do not assume $D^{\text{learn}}, \alpha_\lambda^{\text{learn}(g)}$ is the global minimizer of the non-convex CSC objective $f(\alpha, D)$. We use only the fact that the learned dictionary and coefficients must achieve a lower objective than the random initial dictionary, that is $f(D^{\text{learn}}, \alpha_\lambda^{\text{learn}(g)}) \leq f(D^{\text{init}}, \alpha)$ for all α, because the dictionary learning procedure performs block coordinate descent and is guaranteed to improve the objective at every iteration.

Before we state our main theorem, it is instructive to first consider the excess risk bound we would obtain if using only the random initial dictionary entries with oracle coefficients, with no additional dictionary learning.

Proposition 5.1. *Suppose that assumptions A1, A2, A3 hold. For a given sparsity level s, define*

$$\alpha_{oracle}^{init(g)} = \underset{\{\alpha^{(g)}:\|\alpha^{(g)}\|_0 \leq s, \|\alpha^{(g)}\|_1 \leq L\sqrt{s}\}}{\arg\min} R(D^{init}, \alpha^{(g)}).$$

Let $K \geq \max(n, r(p+q))$*, and* $\lambda \leq \sqrt{\frac{\log K}{n}}$*. Suppose* $s \leq r(p+q)$*. Then with probability at least* $1 - \frac{1}{K}$*,*

$$\max_{g=1,\ldots,G} R(D^{init}, \alpha_{oracle}^{init(g)}) - R(B^{*(g)}) \leq CL^2 \left(\frac{(p+q)\log(GK)}{n}\right)^{s/r(p+q)}$$

where C *is a constant depending only on* $\|\Sigma\|_2$ *as defined in A1.*

Setting $s = \frac{r(p+q)}{2}$, we observe that a large enough dictionary of random rank one matrices with the (non-sparse) oracle coefficients yields an excess risk bound that, up to multiplicative constants, matches the bound in Theorem 5.1–the best we can hope for. But because the oracle coefficients $\alpha_{\text{oracle}}^{\text{init}(g)}$ are not sparse, the learned coefficients $\alpha_\lambda^{\text{init}(g)}$ will be a poor estimate of the oracle coefficients, and the resulting excess risk may be significantly larger.

Prop 5.1 and the preceding discussion motivate the need for learning the dictionary—we may improve statistical accuracy if we can customize the dictionary toward the $B^{*(g)}$, allowing reconstruction of $B^{*(g)}$ from the dictionary using sparse coefficients. Our main theorem in this subsection formalizes this intuition.

Theorem 5.2. *Suppose assumptions A1, A2, A3 hold. Suppose $K \geq \max(n, r(p+q))$, $\lambda \leq \sqrt{\frac{\log K}{n}}$, and $\tau \leq 1$. Then with probability at least $1 - \frac{1}{K}$,*

$$\max_{g=1,\ldots,G} R(D^{learn}, \alpha_\lambda^{learn}) - R(B^{*(g)}) \leq C \max(L^2, \|\alpha_\lambda^{learn}\|_1^2)\sqrt{\frac{(p+q)\log(GK)}{n}}.$$

This result implies that if the learned coefficients are sparse, that is, if $\|\alpha_\lambda^{\text{learn}(g)}\|_1 \leq L$, then the excess risk of conditional sparse coding is, up to a multiplicative constant factor, no greater that the excess risk for estimating separate low-rank matrices within each group. Of course, the excess risk can be worse if $\|\alpha_\lambda^{\text{learn}(g)}\|_1$ increases with $(p+q)$ or n; we cannot rule out this possibility because the dictionary learning optimization is nonconvex and does not admit a direct analysis. We note in our experimental section, however, that $\alpha_\lambda^{\text{learn}(g)}$ is very sparse in our simulations. We note also that our proof uses critically the fact that our algorithm places a nuclear-norm constraint on the dictionary entries, thus showing that the constraint is necessary to reduce overfitting when learning the dictionary.

Theorem 5.2 and Proposition 5.1 suggest a rule of thumb in applying conditional sparse coding. If the sparsity levels of the coefficients do not decrease with the iterations of dictionary learning, then the resulting statistical accuracy may be poor.

5.2 Optimistic Analysis

For our optimistic analysis, we consider the specific setting where the dictionary is already learned and we analyze the excess risk incurred when we fit the coefficients from data that were not used in the dictionary learning process. This setting is limited, but is relevant to situations such those in our experiments with financial data.

A4. The learned dictionary $\{D_1^{\text{learn}}, ..., D_K^{\text{learn}}\}$ is independent of the data $X_i^{(g)}$ for all groups g and items $i = 1, ..., n$.

With the dictionary fixed, we let

$$\alpha_{\text{oracle}}^{\text{learn}(g)} \equiv \operatorname*{arg\,min}_{\{\alpha^{(g)}:\|\alpha^{(g)}\|_1 \leq L\}} R(D^{\text{learn}}, \alpha^{(g)})$$

be the sparse coefficients that minimize the true risk. We can then interpret the oracle excess risk $R(D^{\text{learn}}, \alpha_{\text{oracle}}^{\text{learn}(g)}) - R(B^{*(g)})$ as a measure of the extent to which the oracle regression matrices $B^{*(g)}$ share structure, and the learned dictionary has captured this structure.

Theorem 5.3. *Suppose assumptions A1, A2, A4 hold. Let C be a constant. Suppose $\lambda \leq \sqrt{\frac{\log K}{n}}$. Then with probability at least $1 - \frac{1}{n}$,*

$$\max_{g=1,\ldots,G} R(D^{learn}, \alpha_\lambda^{learn(g)}) - R(B^{*(g)}) \ \leq$$
$$C \max(L^2, \|\alpha_\lambda^{learn(g)}\|_1^2)\sqrt{\frac{\log(npKG)}{n}} \qquad + R(D^{learn}, \alpha_{oracle}^{learn(g)}) - R(B^{*(g)})$$

where C is some constant dependent only on $\|\Sigma\|_2$ as defined in A1.

Under the optimistic assumption that the excess risk $R(D^{\text{learn}}, \alpha_{\text{oracle}}^{\text{learn}(g)}) - R(B^{*(g)})$ is small, that is, that the dictionary has effectively learned the common information among the groups, then we require on the order of $\sqrt{p+q}$ times fewer samples here to achieve the same excess risk as in Theorem 5.2. If we further assume that $\|\alpha_\lambda^{\text{learn}(g)}\|_1$ does not increase with p, q, meaning that the oracle coefficients are sparse, then the excess risk in the optimistic setting is also lower than the bound in Theorem 5.1.

5.3 Proof Sketches

Proof sketch of theorem 5.1: The crux of our argument is the following uniform generalization error bound:

Lemma 5.1. *We have with probability at least* $1-\exp(-cp)$*, for all matrices* $B^{(g)}$ *such that* $\|B^{(g)}\|_* \leq L$*,* $R(B^{(g)}) - \widehat{R}(B^{(g)}) \leq CL^2\sqrt{\frac{(p+q)\log(Gn)}{n}} + R_u$*. where* c, C *are some constants dependent only on* $\|\Sigma\|_2$ *as defined in A1 and* R_u *is a term that does not depend on* $B^{(g)}$*.*

We prove lemma 5.1 by combining the technique of [12] with a concentration result from random matrix theory which states that for independent subgaussian random vector $Z_1, ..., Z_n$, $\|\frac{1}{n}\sum_{i=1}^{n} Z_i Z_i^{\mathsf{T}} - \Sigma_Z\|_2 \leq C\sqrt{\frac{p}{n}}$ with probability at least $1 - \exp -cp$ for some constants c, C. Theorem 5.1 then follows from a standard argument.

Proof sketch of Proposition 5.1: The proof is constructive. It uses a theoretical procedure, similar to orthogonal matching pursuit but infeasible to implement, to produce a set of $\alpha^{(g)}$ with sparsity level s for the random rank 1 dictionary entries so that the reconstruction error $\|B^{*(g)} - \sum_{k=1}^{K} D^{\text{init}}\alpha_k^{(g)}\|_F$ and the associated excess risk would be sufficiently low. Since $\alpha_{\text{oracle}}^{\text{init}}$ is the optimal set of s-sparse coefficients, we can upper bound its risk with the risk of our constructed coefficients. We do not prove that our bound is tight, but analysis by [13] suggest that our bound cannot be significantly improved. We discuss this point further in the appendix.

Proof sketch of Theorem 5.2: We first re-write the excess risk as

$$
\begin{aligned}
& R(D^{\text{learn}}, \alpha_\lambda^{\text{learn}(g)}) - R(B^{*(g)}) \\
& \quad = R(D^{\text{learn}}, \alpha_\lambda^{\text{learn}(g)}) - \widehat{R}(D^{\text{learn}}, \alpha_\lambda^{\text{learn}(g)}) && (5.1) \\
& \quad\quad + \widehat{R}(D^{\text{learn}}, \alpha_\lambda^{\text{learn}(g)}) - \widehat{R}(D^{\text{init}}, \alpha_{\text{oracle}}^{\text{init}(g)}) && (5.2) \\
& \quad\quad + \widehat{R}(D^{\text{init}}, \alpha_{\text{oracle}}^{\text{init}(g)}) - R(D^{\text{init}}, \alpha_{\text{oracle}}^{\text{init}(g)}) && (5.3) \\
& \quad\quad + R(D^{\text{init}}, \alpha_{\text{oracle}}^{\text{init}(g)}) - R(B^{*(g)}) && (5.4)
\end{aligned}
$$

where $\alpha_{\text{oracle}}^{\text{init}(g)}$ is as defined in Proposition 5.1 with s set to $\frac{r(p+q)}{2}$.

We then bound (5.1) using Lemma 5.1. To control (5.2), we observe that although the dictionary learning procedure is nonconvex, it is guaranteed to improve the objective. Thus, we have immediately that (5.2) is at most $\lambda\|\alpha_{\text{oracle}}^{\text{init}(g)}\|_1$. A bound on (5.4) follows from Proposition 5.1. Term (5.3) requires the following lemma concerning uniform generalization error of learning coefficients for a fixed dictionary:

Lemma 5.2. *Let* $D_1, ..., D_K$ *be a fixed set of dictionary entries with* $\|D_k\|_* \leq 1$*. We have that with probability at least* $1 - \frac{1}{n}$*, for all coefficients* $\alpha^{(g)}$*,* $\max_g R(D, \alpha^{(g)}) - \widehat{R}(D, \alpha^{(g)}) \leq C\|\alpha^{(g)}\|_1^2\sqrt{\frac{\log(GKpn)}{n}} + R_u$ *where* C *is a constant dependent only on* $\|\Sigma\|_2$ *as defined in A1 and* R_u *is a term that does not depend on* $\alpha^{(g)}$

Proof sketch of Theorem 5.3: The proof is straightforward by combining assumption A4, Theorem 5.3, and Lemma 5.2.

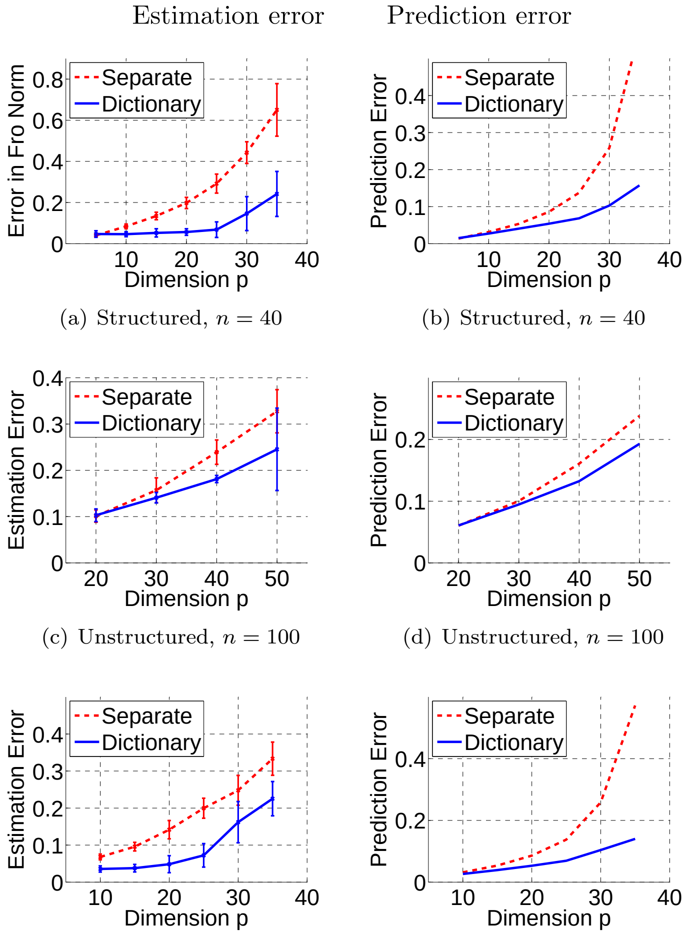
6 Experiments

The main purpose of our experiments is to compare conditional sparse coding against reduced-rank regression. We will also illustrate that the coefficients estimated by CSC are indeed sparse and that the dictionaries learned are indeed low rank.

6.1 Simulation Data

We generate data via the linear model $Y^{(g)} = B^{*(g)}X^{(g)} + \epsilon^{(g)}$ where $\epsilon^{(g)} \sim N(0, \sigma^2 I_q)$ and each $B^{*(g)}$ is a $p \times p$ square matrix. We build a random design matrix $X^{(g)}$ by drawing each sample $X_i^{(g)} \sim N(0, I_p)$. We consider three different settings:

1. In the **structured** case, we construct each $B^{*(g)}$ as a random 3-sparse linear combination of a set of 30 rank one true dictionary matrices. Groups constructed by this method will share considerable common information. Our estimator have no knowledge of the true dictionary of course.
2. In the **unstructured** case, we construct each $B^{*(g)}$ as simply a random rank 3 matrix.
3. The **structured same design** case is the same as the structured case except that every group shares the same design $X^{(g)}$. We study this case because real-world data can have overlapping groups.



(a) Structured, $n = 40$ (b) Structured, $n = 40$

(c) Unstructured, $n = 100$ (d) Unstructured, $n = 100$

(e) Same design, $n = 60$ (f) Same design, $n = 60$

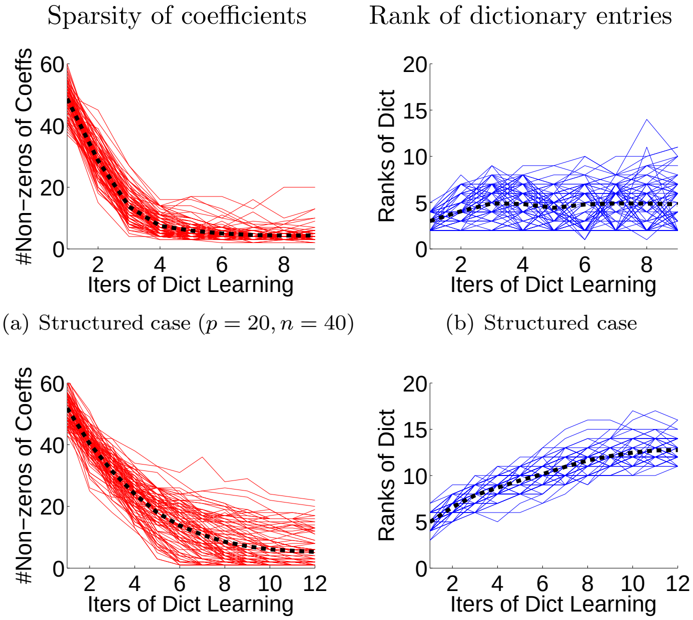
Figure 1: Comparison of CSC to reduced rank regression.

We measure performance of the algorithms in term of both *estimation error* $\frac{1}{G}\sum_{g=1}^{G}\|B^{*(g)} - \widehat{B}^{(g)}\|_F$ and *prediction error* $\widehat{R}_{test}(\widehat{B}^{(g)})$, which is computed from a large test set of $(X^{(g)}, Y^{(g)})$ pairs.

We compare CSC against performing separate reduced rank regressions for each group using nuclear norm-regularization.

It can be seen from Figure 6.1 that when the parameter matrices $\{B^{*(g)}\}$ have significant common structure, CSC easily outperforms separate regressions with either different or the same design for each group. CSC performs worse in the unstructured case as expected, but is still competitive with separate regression.

In Figure 6.1, we show the sparsity of the coefficients together with the ranks of the learned dictionary entries as a function of iterations of alternation in the algorithm. It is seen that (1) CSC does not require many iterations to converge, (2) the coefficients become increasingly sparse, and (3) although the ranks of the dictionary entries may increase, the learned dictionary entries are still relatively low rank.



(a) Structured case ($p = 20, n = 40$)

(b) Structured case

(c) Unstructured case ($p = 20, n = 100$)

(d) Unstructured case

Figure 2: Variation in sparsity and dictionary rank with iterations of alternation in CSC dictionary learning, on simulated data. Each line represents one group or one dictionary entry; dashed black line represents the average.

We note that in section 8 of the appendix, we also perform simulations to see how CSC can adapt to overlapping groups.

6.2 Equities Data

Here we apply CSC to stock returns of 50 of the S&P 500 companies between 2001 to 2007. We predict the returns of 30 companies from the same-day returns of 20 different companies. Predicting the same day stock returns could have applications in high-frequency trading since the stock prices of different companies may not be updated at the same time. Each covariate vector $X_t \in \mathbb{R}^{20}$ and each response vector $Y_t \in \mathbb{R}^{30}$ is the one-day log-return $\log \frac{P_t}{P_{t-1}}$ where P_t is a vector of prices at the close on day t.

We compare the performance of CSC to reduced rank regression fit on the previous 50, 100, 400, and 800 days as training data. For CSC, we form groups according to blocks of 50 consecutive days of data, resulting in 60 overlapping groups with which to learn the dictionary. Once the dictionary is learned, we fit a model on each of two new groups and evaluate the predictive accuracy of the fitted models on 10 test data points in each of the two new groups, containing data points in future days. We repeat this process for 14 sessions at different starting days. We use 7 sessions for parameter tuning and use 7 sessions to report the predictive accuracy.

We evaluate the predictive peformance through explained variance, that is, the predictive $R^2 = 1 - \frac{\|Y_{\text{true}} - Y_{\text{predict}}\|_2^2}{\|Y_{\text{true}}\|_2^2}$ measure. When predictive R^2 is negative, the performance is worse than the baseline model, which is to simply estimate a log return of zero.

As shown in Figure 3, conditional sparse coding usually outperforms nuclear-norm-regularized regression. It is interesting to note that, with the learned dictionary in place, CSC fits the predictive model using only 40 previous days of data. The fact that CSC outperforms models that use several hundred past days of data suggest that CSC is able to adapt to fact that the true model varies with time.

	50 days back	100 days back	400 days back	800 days back	Dictionary
Session 1	0.1239	0.0752	0.2510	0.2660	**0.2718**
Session 2	0	-0.1963	0.1344	0.1304	**0.1732**
Session 3	0.1437	0.0735	0.1394	0.1269	**0.1878**
Session 4	0.0237	-0.0546	0.1210	**0.1587**	0.1551
Session 5	-0.0689	-0.2214	0.0405	-0.0135	**0.0846**
Session 6	-0.0784	-0.1074	0.0807	**0.1103**	0.0931
Session 7	0.0018	0.0876	0.1814	0.1494	**0.1917**
Overall	0.0303	-0.0380	0.1358	0.1302	**0.1669**

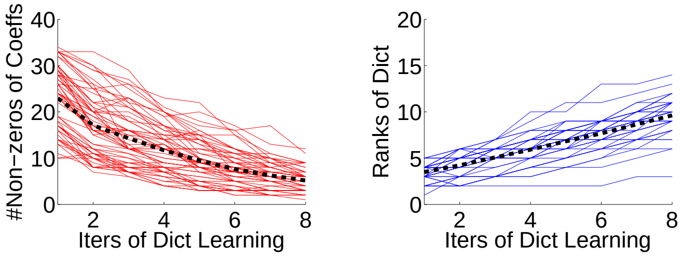


Figure 3: Same-day stock prediction accuracy of Conditional Sparse Coding versus nuclear-norm regularized regression. Each session contains 20 test data points. Right plots show variation in coefficients sparsity levels and dictionary ranks.

6.3 fMRI Data

The dataset, gathered by [21], comprises the brain activity pattern of 9 human subjects when presented with a single concrete noun of the English language. More precisely, we have X as a design matrix of neural signals with dimensions $(p = 17000) \times (n = 60)$ and Y as the response matrix (dimension $(q = 218) \times (n = 60)$) of semantic features of the 60 nouns being shown to the subjects. We let each subject be a group and hence we have the number of groups $G = 9$.

The goal is to predict the semantic features of the noun being shown to the subject, based only on the neural signal of the subject's brain. The predicted semantic features can then be used to guess which word the subject was viewing and thus "read the subject's mind". We refer the readers to [21] for a comprehensive discussion on the significance of this problem. The full dimensionality of $p = 17000$ is too large for our implementation; we down-sample the regions of brain by taking only one value in every 3 voxel by 4 voxel by 4 voxel 3D region so that we reduce the dimensionality to $p = 434$. We also discard some semantic features so that we have $q = 192$.

We use hold-two-out cross-validation for evaluation. In each trial, we hold out two words, use the remaining 58 words for training, and then compute three evaluation metrics: 2 vs. 2 classification, 1 vs. 2 classifcation, and square-error. The square-error is a standard measure of prediction effectiveness. To explain 2 vs. 2 and 1 vs. 2 classification, we first define some notation.

Let y_1, y_2 be the semantic feature vectors of the heldout words. Let $\widehat{y}_1, \widehat{y}_2$ be the predicted semantic feature vectors. We say that we correctly made a 2 vs. 2 classification if

$$d(y_1, \widehat{y}_1) + d(y_2, \widehat{y}_2) < d(y_1, \widehat{y}_2) + d(y_2, \widehat{y}_1)$$

and we say that we correctly made a 1 vs. 2 classification if BOTH of the following hold:

$$\begin{aligned} d(y_1, \widehat{y}_1) &< d(y_1, \widehat{y}_2) \\ d(y_2, \widehat{y}_2) &< d(y_2, \widehat{y}_1) \end{aligned}$$

We observe that 1 vs. 2 classification is harder than 2 vs. 2 classification. If we make random predictions, then the expected 1 vs. 2 classification accuracy is 0.25 and the expected 2 vs. 2 classification accuracy is 0.5.

Our parameters are tuned by separate cross-validation trials. For CSC, we cross-validated among the following list of possible parameter settings: $(\tau = 0.7, \lambda = 0.01, K = 20), (\tau = 1.1, \lambda = 0.02, K = 16), (\tau = 2, \lambda = 0.05, K = 20), (\tau = 0.9, \lambda = 0.01, K = 20), (\tau = 0.5, \lambda = 0.005, K = 10)$. For separate trace-norm regularized regression, we cross-validate among $\lambda = (0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001)$.

In figure 4, 5, 6, we compare the performance of Conditional Sparse Coding vs. separate trace-norm-regularized regression for each subject. We note that CSC often show confident improvement in both 2 vs. 2 and 1 vs. 2 classification tasks with very few cases of significant degradation. In square-error, CSC show improvement in most subjects although on average, the improvement is insignificant.

	Subj A	B	C	D	E	F	G	H	I
Dictionary	0.8833	**0.8667**	0.9000	**0.9333**	**0.8333**	0.7500	**0.9000**	**0.7833**	**0.6667**
Separate	**0.9500**	0.7000	**0.9167**	0.8167	0.8167	**0.7667**	0.8000	0.6667	0.6333
Confidence	0.6-	0.92+	0.05-	0.86+	0.03+	0.02-	0.70+	0.65+	0.07-

Figure 4: fMRI data analysis. 2 vs. 2 classification accuracy from 60 cross-validation trials. Last row list confidence in either improvement (+) or degradation (-)

	Subj A	B	C	D	E	F	G	H	I
Dictionary	0.7000	**0.5333**	0.6667	**0.7667**	0.4833	0.3667	**0.6000**	**0.4333**	**0.3000**
Separate	0.7000	0.3833	0.6667	0.6333	**0.5000**	**0.5000**	0.5333	0.3167	0.2333
Confidence	0	0.75+	0	0.72+	0.01-	0.67-	0.24+	0.58+	0.29+

Figure 5: fMRI data analysis. 1 vs. 2 classification accuracy from 60 cross-validation trials. Last row list confidence in either improvement (+) or degradation (-)

	Subj A	B	C	D	E	F	G	H	I
Dictionary	**243.25**	**276.95**	**256.10**	**247.48**	**291.48**	**310.12**	282.77	329.08	327.27
Separate	255.00	290.94	270.30	253.73	299.84	322.59	**272.45**	**314.69**	**303.41**

Figure 6: fMRI data analysis. Square-error of CSC vs. separate regression. Average across all subjects, we have CSC: 284.95 and separate: 287.00

Although there is indeed sharing of dictionary entries across the various groups (subjects), it is important to mention that the pattern of sharing is highly unstable from trial to trial. Figure 6.3 shows two patterns of group-dictionary utilization derived from the $\alpha^{(g)}$'s. We see that in the first trial, subject 3 shares significantly with subject 7 while subject 1 shares with no other subjects; in the second trial, subject 3 shares with subject 5 and subject 1 shares with subject 6 and subject 9. The instability is possibly due to the low sample size. As a result of the instability, we cannot deduce subject-subject similarities from the group-dictionary utilization patterns.

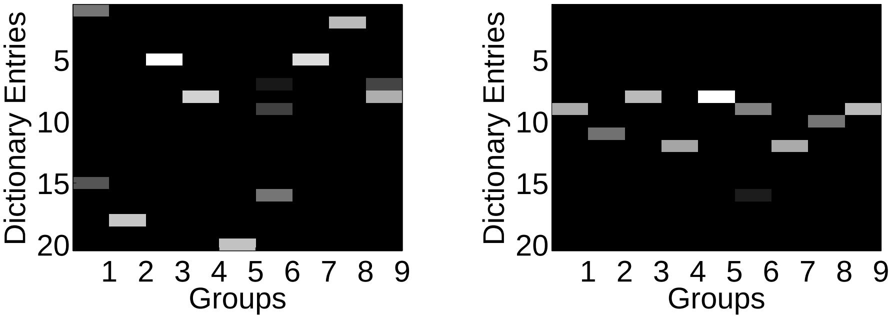


Figure 7: Coefficients $\alpha^{(g)}$'s interpreted as dictionary utilization per group. Lighter color indicates greater utilization

7 Conclusion

We have presented Conditional Sparse Coding, an algorithm for multiple regression with grouped data where each group is associated with a distinct task of multiple regression. CSC exploits information shared between the groups by letting the parameter matrix of each group be a sparse linear combination of a common learned set of dictionary entries. Theoretically, we showed that our procedure cannot achieve significantly greater error than performing separate rank-reduced regression for each group so long as we indeed only use *sparse* linear combinations of the dictionary entries. Experimentally, we showed that our procedure can be more accurate if the groups indeed share common information.

An important open question is whether we can guarantee that the α's, the coefficients for the dictionary entries, are always sparse; such an analysis would be important because our existing theory states that the more sparse the α's are, the more accurate Conditional Sparse Coding is. Simulations and empirical studies have shown that this is always the case.

References

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *NIPS 19*, pages 41–48, Cambridge, MA, 2006. MIT Press.

[2] Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image den oising and deblurring problems. *IEEE Trans. Image Process*, 2009.

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

[4] Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Conference on Learning Theory*, 2003.

[5] Samy Bengio, Fernando Pereira, Yoram Singer, and Dennis Strelow. Group sparse coding. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *NIPS 22*. MIT Press, 2009.

[6] Rich Caruana. Multitask learning. *Mach. Learn.*, 28:41–75, July 1997.

[7] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. *ICML*, 2008.

[8] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 54(12):3736–3745, 2006.

[9] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In Won Kim, Ron Kohavi, Johannes Gehrke, and William DuMouchel, editors, *Proceedings of the Tenth ACM SIGKDD*, pages 109–117, 2004.

[10] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007.

[11] Quan Geng, Huan Wang, and John Wright. On the local correctness of l_1-minimization for dictionary learning, 2011.

[12] E. Greenshtein and Y. Ritov. Persistency in high dimensional linear predictor-selection and the virtue of over-parametrization. *Bernoulli*, 10:971–988, 2004.

[13] H. Jeong and Y-H. Kim. Sparse linear representation. *ISIT*, 2009.

[14] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multitask feature learning. *ICML*, 2011.

[15] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *NIPS 19*, pages 801–808. MIT Press, Cambridge, MA, 2007.

[16] Yan Liu, Alexandru Niculescu-Mizil, Aurelie Lozano, and Yong Lu. Learning temporal causal graphs for relational time-series analysis. *ICML*, 2010.

[17] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *arXiv: 1009.5358*, 2010.

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *International Conference on Computer Vision*, 2009.

[19] Andreas Maurer. Bounds for linear multi-task learning. *Journal of Machine Learning Research*, 7, 2006.

[20] Andreas Maurer and Massimiliano Pontil. K-dimensional coding scheme in hilbert spaces. *IEEE Trans. Inform. Theory*, 54:5839–5846, 2010.

[21] Tom M. Mitchell, Svetlana V. Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L Malave, Robert A. Mason, and Marcel Adam Just. Predicting human brain activity associated with the meanings of nouns. *Science*, pages 1191 – 1195, 2008.

[22] S. Negahban and M.J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Annals of Statistics*, 39:1069–1097, 2011.

[23] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, June 1996.

[24] D. Vainsencher, S. Mannor, and A.M. Bruckstein. The sample complexity of dictionary learning. *CoRR*, 2010.

[25] J. Yang, K. Yu, Y. Gong, , and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[26] Ming Yuan, A. Ekici, Zhaosong Lu, and R. Monteiro. Dimension reduction and coefficient estimation in multivariate linear regression. *J. R. Statist. Soc. B*, 69:329–346, 2007.

[27] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S. Huang. Image classification using super-vector coding of local image descriptors. In *ECCV'10*, pages 141–154. Springer-Verlag, 2010.