Smooth Sparse Coding via Marginal Regression for Learning Sparse Representations

Krishnakumar Balasubramanian KRISHNAKUMAR3@GATECH.EDU
College of Computing, Georgia Institute of Technology

Kai Yu YUKAI@BAIDU.COM
Baidu Inc.

Guy Lebanon LEBANON@CC.GATECH.EDU
College of Computing, Georgia Institute of Technology

Abstract

We propose and analyze a novel framework for learning sparse representations, based on two statistical techniques: kernel smoothing and marginal regression. The proposed approach provides a flexible framework for incorporating feature similarity or temporal information present in data sets, via non-parametric kernel smoothing. We provide generalization bounds for dictionary learning using smooth sparse coding and show how the sample complexity depends on the L_1 norm of kernel function used. Furthermore, we propose using marginal regression for obtaining sparse codes, which significantly improves the speed and allows one to scale to large dictionary sizes easily. We demonstrate the advantages of the proposed approach, both in terms of accuracy and speed by extensive experimentation on several real data sets. In addition, we demonstrate how the proposed approach can be used for improving semi-supervised sparse coding.

1. Introduction

Sparse coding is a popular unsupervised paradigm for learning sparse representations of data samples, that are subsequently used in classification tasks. In standard sparse coding, each data sample is coded independently with respect to the dictionary. We propose a smooth alternative to traditional sparse coding that incorporates feature similarity, temporal or other user-specified domain information between the samples, into the coding process.

The idea of smooth sparse coding is motivated by the relevance weighted likelihood principle. Our approach constructs a code that is efficient in a smooth sense and as a result leads to improved statistical accuracy over traditional sparse coding. The smoothing operation, which can be expressed as non-parametric kernel smoothing, provides a flexible framework for incorporating several types of domain information that might be available for the user. For example, in image classification task, one could use: (1) kernels in feature space for encoding similarity information for images and videos, (2) kernels in time space in case of videos for incorporating temporal relationship, and (3) kernels on unlabeled image in the semi-supervised learning and transfer learning settings.

Most sparse coding training algorithms fall under the general category of alternating procedures with a convex lasso regression sub-problem. While efficient algorithms for such cases exist (Lee et al., 2007), their scalability for large dictionaries remains a challenge. We propose a novel training method for sparse coding based on marginal regression, rather than solving the traditional alternating method with lasso subproblem. Marginal regression corresponds to several univariate linear regression followed by a thresholding step to promote sparsity. For large dictionary sizes, this leads to a dramatic speedup compared to traditional sparse coding methods (up to two orders of magnitude) without sacrificing statistical accuracy.

We also develop theory that extends the sample complexity result of (Vainsencher et al., 2011) for dictionary learning using standard sparse coding to the smooth sparse coding case. This result specifically shows how the sample complexity depends on the L_1

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

norm of the kernel function used.

Our main contributions are: (1) proposing a framework based on kernel-smoothing for incorporating feature, time or other similarity information between the samples into sparse coding, (2) providing sample complexity results for dictionary learning using smooth sparse coding, (3) proposing an efficient marginal regression training procedure for sparse coding, and (4) successful application of the proposed method in various classification tasks. Our contributions lead to improved classification accuracy in conjunction with computational speedup of two orders of magnitude.

2. Related work

Our approach is related to the local regression method (Loader, 1999; Hastie & Loader, 1993). More recent related work is (Meier & Bühlmann, 2007) that uses smoothing techniques in high-dimensional lasso regression in the context of temporal data. Another recent approach proposed by (Yu et al., 2009) achieves code locality by approximating data points using a linear combination of nearby basis points. The main difference is that traditional local regression techniques do not involve basis learning. In this work, we propose to learn the basis or dictionary along with the regression coefficients locally.

In contrast to previous sparse coding papers we propose to use marginal regression for learning the regression coefficients, which results in a significant computational speedup with no loss of accuracy. Marginal regression is a relatively old technique that has recently reemerged as a computationally faster alternative to lasso regression (Fan & Lv, 2008). See also (Genovese et al., 2012) for a statistical comparison of lasso regression and marginal regression.

3. Smooth Sparse Coding

Notations: The notations x and X correspond to vectors and matrices respectively, in appropriately defined dimensions; the notation $\|\cdot\|_p$ corresponds to the L_p norm of a vector (we use mostly use $p = 1, 2$ in this paper); the notation $\|\cdot\|_F$ corresponds to the Frobenius norm of a matrix; the notation $|f|_p$ corresponds to the L_p norm of the function f: $(\int |f|^p \, d\mu)^{1/p}$; the notation $x_i, i = 1, \ldots, n$ corresponds to the data samples, where we assume that each sample x_i is a d-dimensional vector. The explanation below uses L_1 norm to promote sparsity. But the method applies more generally to any structured regularizers, for e.g., (Bronstein et al., 2012; Jenatton et al., 2010).

The standard sparse coding problem consists of solving the following optimization problem,

$$\begin{aligned} \min_{\substack{D\in\mathbb{R}^{d\times K} \\ \beta_i\in\mathbb{R}^K, i=1,\ldots,n}} \quad & \sum_{i=1}^{n} \|x_i - D\beta_i\|_2^2 \\ \text{subject to} \quad & \|d_j\|_2 \le 1 \quad j = 1, \ldots K \\ & \|\beta_i\|_1 \le \lambda \quad i = 1, \ldots n. \end{aligned}$$

where $\beta_i \in \mathbb{R}^K$ corresponds to the encoding of sample x_i with respected to the dictionary $D \in \mathbb{R}^{d\times K}$ and $d_j \in \mathbb{R}^d$ denotes the j-column of the dictionary matrix D. The dictionary is typically over-complete, implying that $K > d$.

Object recognition is a common sparse coding application where x_i corresponds to a set of features obtained from a collection of image patches, for example SIFT features (Lowe, 1999). The dictionary D corresponds to an alternative coding scheme that is higher dimensional than the original feature representation. The L_1 constraint promotes sparsity of the new encoding with respect to D. Thus, every sample is now encoded as a sparse vector that is of higher dimensionality than the original representation.

In some cases the data exhibits a structure that is not captured by the above sparse coding setting. For example, SIFT features corresponding to samples from the same class are presumably closer to each other compared to SIFT features from other classes. Similarly in video, neighboring frames are presumably more related to each other than frames that are farther apart. In this paper we propose a mechanism to incorporate such feature similarity and temporal information into sparse coding, leading to a sparse representation with an improved statistical accuracy (for example as measured by classification accuracy).

We consider the following smooth version of the sparse coding problem above:

$$\begin{aligned} \min_{\substack{D\in\mathbb{R}^{d\times K} \\ \beta_i\in\mathbb{R}^K, i=1,\ldots,n}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} w(x_j, x_i)\|x_j - D\beta_i\|_2^2 && (1) \\ \text{subject to} \quad & \|d_j\|_2 \le 1 \quad j = 1, \ldots K && (2) \\ & \|\beta_i\|_1 \le \lambda \quad i = 1, \ldots n. && (3) \end{aligned}$$

where $\sum_{j=1}^{n} w(x_j, x_i) = 1$ for all i. It is convenient to define the weight function through a smoothing kernel

$$w(x_j, x_i) = \frac{1}{h_1}\mathcal{K}_1\left(\frac{\rho(x_j, x_i)}{h_1}\right)$$

where $\rho(\cdot,\cdot)$ is a distance function that captures the feature similarity, h_1 is the bandwidth, and $\mathcal{K}_1$ is a smoothing kernel. Traditional sparse coding minimizes the reconstruction error of the encoded samples.

Smooth sparse coding, on the other hand, minimizes the reconstruction of encoded samples with respect to their neighbors (weighted by the amount of similarity).

The smooth sparse coding setting leads to codes that represent a neighborhood rather than an individual sample and that have lower mean square reconstruction error (with respect to a given dictionary), due to lower estimation variance (see for example the standard theory of smoothed empirical process (Devroye & Lugosi, 2001)). There are several possible ways to determine the weight function w. One common choice for the kernel function is the Gaussian kernel whose bandwidth is selected using cross-validation. Other common choices for the kernel are the triangular, uniform, and tricube kernels. The bandwidth may be fixed throughout the input space, or may vary in order to take advantage of non-uniform samples. We use in our experiment the tricube kernel with a constant bandwidth.

The distance function $\rho(\cdot,\cdot)$ may be one of the standard distance functions (for example based on the L_p norm). Alternatively, $\rho(\cdot,\cdot)$ may be expressed by domain experts, learned from data before the sparse coding training, or learned jointly with the dictionary and codes during the sparse coding training.

3.1. Spatio-Temporal smoothing

In spatio-temporal applications we can extend the kernel to include also a term reflecting the distance between the corresponding time or space

$$w(x_j, x_i) = \frac{1}{h_1}\mathcal{K}_1\left(\frac{\rho(x_j, x_i)}{h_1}\right)\frac{1}{h_2}\mathcal{K}_2\left(\frac{j-i}{h_2}\right).$$

Above, $\mathcal{K}_2$ is a univariate symmetric kernel with bandwidth parameter h_2. One example is video sequences, where the kernel above combines similarity of the frame features and the time-stamp.

Alternatively, the weight function can feature only the temporal component and omit the first term containing the distance function between the feature representation. A related approach for that situation, is based on the Fused lasso which penalizes the absolute difference between codes for neighboring points. The main drawback of that approach is that one needs to fit all the data points simultaneously whereas in smooth sparse coding, the coefficient learning step decomposes as n separate problems which provides a computational advantage (see supplementary document for more details). Also, while fused Lasso penalty is suitable for time-series data to capture relatedness between neighboring frames, it may not be immediately suitable for other situations that the proposed smooth sparse coding method could handle.

4. Marginal Regression for Smooth Sparse Coding

A standard algorithm for sparse coding is the alternating bi-convex minimization procedure, where one alternates between (i) optimizing for codes (with a fixed dictionary) and (ii) optimizing for dictionary (with fixed codes). Note that step (i) corresponds to regression with L_1 constraints and step (ii) corresponds to least squares with L_2 constraints. In this section we show how marginal regression could be used to obtain better codes faster (step (i)). In order to do so, we first give a brief description of the marginal regression procedure.

Marginal Regression: Consider a regression model $y = X\beta + z$ where $y \in \mathbb{R}^n$, $\beta \in \mathbb{R}^p$, $X \in \mathbb{R}^{n\times p}$ with L_2 normalized columns (denoted by x_j), and z is the noise vector. Marginal regression proceeds as follows:

- Calculate the least squares solution
$$\hat{\alpha}^{(j)} = x_j^T y.$$
- Threshold the least-square coefficients
$$\hat{\beta}^{(j)} = \hat{\alpha}^{(j)} 1_{\{|\hat{\alpha}^{(j)}|>t\}}, \quad j = 1, \dots, p.$$

Marginal regression requires just $O(np)$ operations compared to $O(p^3 + np^2)$, the typical complexity of lasso algorithms. When p is much larger than n, marginal regression provides two orders of magnitude speedup over Lasso based formulations. Note that in sparse coding, the above speedup occurs for each iteration of the outer loop, thus enabling sparse coding for significantly larger dictionary sizes. Recent studies have suggested that marginal regression is a viable alternative for Lasso given its computational advantage over lasso. A comparison of the statistical properties of marginal regression and lasso is available in (Fan & Lv, 2008; Genovese et al., 2012).

Code update (step (i)): Applying marginal regression to smooth sparse coding, we obtain the following scheme. The marginal least squares coefficients are

$$\hat{\alpha}_i^{(k)} = \sum_{j=1}^{n} \frac{w(x_j, x_i)}{\|d_k\|_2} d_k^T x_j.$$

We sort these coefficient in terms of their absolute values, and select the top s coefficients whose L_1 norm is bounded by λ:

$$\hat{\beta}_i^{(k)} = \begin{cases} \hat{\alpha}_i^{(k)} & k \in S \\ 0 & k \notin S \end{cases}, \quad \text{where}$$

$$S = \left\{1, \dots, s \,:\, s \le d : \sum_{k=1}^{s} |\hat{\alpha}_i^{(k)}| \le \lambda \right\}$$

We select the thresholding parameter using cross validation in each of the sparse coding iterations. Note that the same approach could be used with structured regularizers too, for example (Bronstein et al., 2012; Jenatton et al., 2010).

Dictionary update (step (ii)): Marginal regression works well when there is minimal correlation between the different dictionary atoms. In the linear regression setting, marginal regression performs much better with orthogonal data (Genovese et al., 2012). In the context of sparse coding, this corresponds to having uncorrelated or incoherent dictionaries (Tropp, 2004). One way to measure such incoherence is using the babel function, which bounds the maximum inner product between two different columns d_i, d_j:

$$\mu_s(D) = \max_{i\in\{1,\ldots,d\}} \max_{\Lambda\subset\{1,\ldots,d\}\setminus\{i\};|\Lambda|=s} \sum_{j\in\Lambda} |d_j^\top d_i|.$$

An alternative, which leads to easier computation is by adding the term $\|D^TD - I_{K\times K}\|_F^2$ to the reconstruction objective, when optimizing over the dictionary matrix D. This leads to the following optimization problem for dictionary update step:

$$\hat{D} = \arg\min_{D\in\mathcal{D}} F(D) \quad \text{where}$$
$$F(D) = \sum_{i=1}^{n} \|x_i - D\hat{\beta}_i\|_2^2 + \gamma\|D^\top D - I\|_F^2$$

and $\mathcal{D} = \{D \in \mathbb{R}^{d\times K} : \|d_j\|_2 \le 1\}$. The regularization term γ controls the level of incoherence enforced.

This optimization problem is of the form of minimizing a differentiable function over a closed convex set. We use the gradient projection method (Bertsekas, 1976; Solodov, 1997) for solving the above optimization problem. The gradient (cf. (Magnus & Neudecker, 1988)) of the above expression with respect to D at each iteration is given by $\nabla F(D) = 2\left(D\hat{B}\hat{B}^\top - X\hat{B}^\top\right) + 4\gamma\left(DD^\top D - D\right)$, where $\hat{B} = [\hat{\beta}_1, \ldots, \hat{\beta}_n]$ is the matrix of codes from the previous code update step, $X \in \mathbb{R}^{p\times n}$ is the data in matrix format. The gradient projection descent iterations are given by

$$D(t+1) = \Pi_{\mathcal{D}}\left(D(t) - \eta_t \nabla F(D(t))\right).$$

where by $\Pi_{\mathcal{D}}$, we denote column-wise projection of the dictionary matrix on to the unit ball and t is the index for sub-iteration count for each dictionary update step. Specifically, for each dictionary update step, we run the gradient projected descent algorithm untill convergence (more details about this in experimental section). Note that projection of a vector onto the l_2 ball is straightforward since we only need to rescale the vector towards the origin, i.e., normalize the vectors with length greater than 1.

Convergence to local point of gradient projection methods for minimizing differentiable functions over convex set have been analyzed in (Solodov, 1997). Similar guarantees could be provided for each of the dictionary update steps. A heuristic approach for dictionary update with incoherence constraint was proposed in (Ramırez et al., 2009) and more recently in (Sigg et al., 2012)(where the L-BFGS method was used for the unconstrained problem and the norm constraint was enforced at the final step). We found that the proposed gradient projected descent method performed empirically better than both the approaches. Furthermore both approaches are heuristic and do not guarantee local convergence for the dictionary update step.

Finally, a sequence of such updates corresponding to step (i) and step (ii) converges to a stationary point of the optimization problem (this can be shown using Zangwill's theorem (Zangwill, 1969)). But no provable algorithm that converges (under certain assumptions) to the global minimum of the smooth sparse coding (or standard sparse coding) exists yet. Nevertheless, the main idea of this section is to speed-up the existing alternating minimization procedure for obtaining sparse representations, by using marginal regression. We leave a detailed theoretical analysis of the individual dictionary update steps and the overall alternating procedure (for codes and dictionary) as future work.

5. Sample Complexity of Smooth sparse coding

In this section, we analyze the sample complexity of the proposed smooth sparse coding framework. Specifically, since there does not exist a provable algorithm that converges to the global minimum of the optimization problem in Equation (1), we provide uniform convergence bounds over the dictionary space and thereby prove a sample complexity result for dictionary learning under smooth spare coding setting. We leverage the analysis for dictionary learning in the standard sparse coding setting by (Vainsencher et al., 2011) and extend it to the smooth sparse coding setting. The main difficulty for the smooth sparse coding setting is obtaining a covering number bound for an appropriately defined class of functions (see Theorem 1 for more details).

We begin by re-representing the smooth sparse coding problem in a convenient form for analysis. Let $x_1, \ldots, x_n$ be independent random variables with a common probability measure $\mathbb{P}$ with a density p. We denote by $\mathbb{P}_n$ the empirical measure over the n samples, and the kernel density estimate of p is defined

Algorithm 1 Smooth Sparse Coding via Marginal Regression

```
**Input:** Data $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ and kernel/similarity measure $\mathcal{K}_1$ and $d_1$.
**Precompute:** Compute the weight matrix $w(i, j)$ using the kernel/similarity measure and
**Initialize:** Set the dictionary at time zero to be $D_0$.
**Algorithm:**
**repeat**
    **Step (i):** For all $i = 1, \ldots, n$, solve marginal regression:
        $\hat{\alpha}_i^{(k)} = \sum_{j=1}^{n} \frac{w(x_j, x_i)}{\|d_k\|_2} d_k^T x_j$
        $\hat{\beta}_j^{(k)} = \begin{cases} \hat{\alpha}_j^{(k)} & j \in S \\ 0 & j \notin S \end{cases},$
        $S = \{1, \ldots, s; s \leq d : \sum_{k=1}^{s} |\hat{\alpha}_i^{(k)}| \leq \lambda\}.$
    **Step (ii):** Update the dictionary based on codes from previous step.
        $\hat{D} = \arg\min_{D \in \mathcal{D}} \sum_{i=1}^{n} \|x_i - D\hat{\beta}_i\|_2^2 + \gamma \|D^\top D - I\|_F^2$
        where $\mathcal{D} = \{D \in \mathbb{R}^{d \times K} : \|d_j\|_2 \leq 1\}$
**until** convergence
**Output:** Return the learned codes and dictionary.
```

by

$$p_{n,h}(x) = \frac{1}{nh} \sum_{i=1}^{n} \mathcal{K}\left(\frac{\|x - X_i\|_2}{h}\right).$$

Let $\mathcal{K}_{h_1}(\cdot) = \frac{1}{h_1}\mathcal{K}_1(\frac{\cdot}{h})$. With the above notations, the reconstruction error at the point x is given by

$$r_\lambda(x) = \int \min_{\beta \in \mathcal{S}_\lambda} \|x' - D\beta\|_2 \mathcal{K}_{h_1}(\rho(x, x'))\, d\mathbb{P}_n(x')$$

where

$$\mathcal{S}_\lambda = \{\beta : \|\beta\|_1 \leq \lambda\}.$$

The empirical reconstruction error is

$$\mathsf{E}_{\mathbb{P}_n}(r) = \iint \min_{\beta \in \mathcal{S}_\lambda} \|x' - D\beta\|_2 \mathcal{K}_{h_1}(\rho(x, x'))\, d\mathbb{P}_n(x')\, dx$$

and its population version is

$$\mathsf{E}_{\mathbb{P}}(r) = \iint \min_{\beta \in \mathcal{S}_\lambda} \|x' - D\beta\|_2 \mathcal{K}_{h_1}(\rho(x, x'))\, d\mathbb{P}(x')\, dx.$$

Our goal is to show that the sample reconstruction error is close to the true reconstruction error. Specifically, to show $\mathsf{E}_{\mathbb{P}}(r_\lambda) \leq (1 + \kappa)\, \mathsf{E}_{\mathbb{P}_n}(r_\lambda) + \epsilon$ where $\epsilon, \kappa \geq 0$, we bound the covering number of the class of functions corresponding to the reconstruction error. We assume a dictionary of bounded babel function, which holds as a result of the relaxed orthogonality constraint used in the Algorithm 1 (see also (Ramırez et al., 2009)). We define the set of r functions with respect the the dictionary D (assuming data lies in the unit d-dimensional ball $\mathbb{S}^{d-1}$) by

$$\mathcal{F}_\lambda = \{r_\lambda : \mathbb{S}^{d-1} \to \mathbb{R} : D \in \mathbb{R}^{d \times K}, \|d_i\|_2 \leq 1, \mu_s(D) \leq \gamma\}.$$

The following theorem bounds the covering number of the above function class.

Theorem 1. *For every $\epsilon > 0$, the metric space $(\mathcal{F}_\lambda, |\cdot|_\infty)$ has a subset of cardinality at most $\left(\frac{4\lambda|\mathcal{K}_{h_1}(\cdot)|_1}{\epsilon(1-\gamma)}\right)^{dK}$, such that every element from the class is at a distance of at most ϵ from the subset, where $|\mathcal{K}_{h_1}(\cdot)|_1 = \int |\mathcal{K}_{h_1}(x)|\, d\mathbb{P}$.*

Proof. Let $\mathcal{F}'_\lambda = \{r'_\lambda : \mathbb{S}^{d-1} \to \mathbb{R} : D \in d \times K, \|d_i\|_2 \leq 1\}$, where $r'_\lambda(x) = \min_{\beta \in \mathcal{S}_\lambda} \|D\beta - x\|$. With this definition we note that $\mathcal{F}_\lambda$ is just $\mathcal{F}'_\lambda$ convolved with the kernel $\mathcal{K}_{h_1}(\cdot)$. By Young's inequality (Devroye & Lugosi, 2001) we have,

$$|\mathcal{K}_{h_1} * (s_1 - s_2)|_p \leq |\mathcal{K}_{h_1}|_1 |s_1 - s_2|_p, \quad 1 \leq p \leq \infty$$

for any L_p integrable functions s_1 and s_2. Using this fact, we see that convolution mapping between metric spaces $\mathcal{F}'$ and $\mathcal{F}$ converts $\frac{\epsilon}{|\mathcal{K}_{h_1}(\cdot)|_1}$ covers into ϵ covers. From (Vainsencher et al., 2011), we have that the the class $\mathcal{F}'_\lambda$ has ϵ covers of size at most $\left(\frac{4\lambda}{\epsilon(1-\gamma)}\right)^{dK}$. This proves the the statement of the theorem. □

The above theorem could be used in conjunction with standard statements in the literature for bounding the generalization error of empirical risk minimization algorithms based on covering numbers. We have provided the general statements in the appendix for completeness of this paper. The proofs of the general statements could be found in the references cited. Below, we provide two such generalization bounds for smooth sparse coding problem, corresponding to slow rates and fast rates.

Slow rates: When the theorem on covering numbers for the function class $\mathcal{F}_\lambda$ (Theorem 1) is used along with Lemma 1 stated in the appendix (corresponding to slow rate generalization bounds) it is straightforward to obtain the following generalization bounds with slow rates for the smooth sparse coding problem.

Theorem 2. *Let $\gamma < 1$, $\lambda > e/4$ with distribution $\mathbb{P}$ on $\mathbb{S}^{d-1}$. Then with probability at least $1 - e^{-t}$ over* *the n samples drawn according to $\mathbb{P}$, for all the D with unit length columns and $\mu_s(D) \leq \gamma$, we have:*

$$\mathsf{E}_{\mathbb{P}}(r_\lambda) \leq \mathsf{E}_{\mathbb{P}_n}(r_\lambda) + \sqrt{\frac{dK \ln\left(\frac{4\sqrt{n}\lambda|\mathcal{K}_{h_1}(\cdot)|_1}{(1-\gamma)}\right)}{2n}} + \sqrt{\frac{t}{2n}} + \sqrt{\frac{4}{n}}$$

The above theorem, establishes that the generalization error scales as $O(n^{-1/2})$ (assuming the other problem parameters are fixed).

Fast rates: Under further assumptions ($\kappa > 0$), it is possible to obtain faster rates of $O(n^{-1})$ for smooth sparse coding, similar to the ones obtained for general learning problems in (Bartlett et al., 2005). The following theorem gives the precise statement.

Theorem 3. *Let $\gamma < 1$, $\lambda > e/4$, $dK > 20$ and $n \geq 5000$. Then with probability at least $1 - e^{-t}$, we have for all D with unit length and $\mu_s(D) \leq \gamma$,*

$$\mathsf{E}_{\mathbb{P}}(r_\lambda) \leq 1.1\, \mathsf{E}_{\mathbb{P}_n}(r_\lambda) + 9\frac{dK \ln\left(\frac{4n\lambda|\mathcal{K}_{h_1}(\cdot)|_1}{(1-\gamma)}\right) + t}{n}.$$

The above theorem follows from the theorem on covering number bound (Theorem 1) above and Lemma 2 from the appendix. In both statements the definition of $r_\lambda(x)$ differs from (1) by a square term, but it could easily be incorporated into the above bounds resulting in an additive factor of 2 inside the logarithm term as is done in (Vainsencher et al., 2011).

6. Experiments

We demonstrate the advantage of the proposed approach both in terms of speed-up and accuracy, over standard sparse coding. A detailed description of all real-world data sets used in the experiments are given in the appendix. As discussed before, the overall optimization procedure is non-convex. The stopping criterion was chosen as when the value of the reconstruction error did not change by more than 0.001%. Though this does not gaurantee convergence to a global optimum, according to the experimental results, we see that the points of convegence invariably resulted in a good local optimum (as reflected by the good empirical performance). Furthermore, in all the experiments, we ran 10 iterations of the projected gradient descent algorithm for each dictionary update step. We fixed the learning rate for all iterations of gradient projection descent algorithm as $\eta = \eta_t = 0.01$ as it was found to performed well in the experiments. The parameters γ and t are set for each experiment based on cross-validation (we first tuned for γ and then for t) for classification results on training set as is done in the literature (Yang et al., 2010).

6.1. Speed comparison

We conducted synthetic experiments to examine the speed-up provided by sparse coding with marginal regression. The data was generated from a a 100 dimensional mixture of two Gaussian distribution that satisfies $\|\mu_1 - \mu_2\|_2 = 3$ (with identity covariance matrices). The dictionary size was fixed at 1024.

We compare the proposed smooth sparse coding algorithm, standard sparse coding with lasso (Lee et al., 2007) and marginal regression updates respectively, with a relative reconstruction error $\|X - \hat{D}\hat{B}\|_F / \|X\|_F$ convergence criterion. We experimented with different values of the relative reconstruction error (less than 10%) and report the average time. From Table 1, we see that smooth sparse coding with marginal regression takes significantly less time to achieve a fixed reconstruction error. This is due to the fact that it takes advantage of the spatial structure and use marginal regression updates. It is worth mentioning that standard sparse coding with marginal regression updates performs faster compared to the other two methods that uses lasso updates, as expected (but does not take into account the spatial structure).

Method	time (sec)
SC+LASSO	524.5 $\pm$12
SC+MR	242.2$\pm$10
SSC+LASSO	560.2$\pm$12
SSC+MR	184.4 $\pm$19

Table 1. Time comparison of coefficient learning in SC and SSC with either Lasso or Marginal regression updates. The dictionary update step was same for all methods.

6.2. Experiments with Kernel in Feature space

We conducted several experiments demonstrating the advantage of the proposed coding scheme in different settings. Concentrating on face and object recognition from static images, we evaluated the performance of the proposed approach along with standard sparse coding and LLC (Yu et al., 2009), another method for obtaining sparse features based on locality. Also, we performed experiments on activity recognition from videos based on both space and time based kernels. As mentioned before all results are reported using tricube kernel.

6.2.1. Image classification

We conducted image classification experiments on CMU-multipie, 15 Scene and Caltech-101 data sets. Following (Yang et al., 2010) , we used the following approach for generating sparse image representation: we densely sampled 16×16 patches from images at the pixel level on a gird with step size 8 pixels, computed SIFT features, and then computed the corresponding sparse codes over a 1024-size dictionary. We used max pooling to get the final representation of the image based on the codes for the patches. The process was repeated with different randomly selected training and testing images and we report the average per-class recognition rates (together with its standard deviation estimate) based on one-vs-all SVM classification. As Table 2 indicates, our smooth sparse coding algorithm resulted in significantly higher classification accuracy than standard sparse coding and LLC. In fact, the reported performance is better than previous reported results using unsupervised sparse coding techniques (Yang et al., 2010).

	CMU-multipie	15 scene	Caltech-101
SC	92.70±1.21	80.28±2.12	73.20±1.14
LLC	93.70±2.22	82.28±1.98	74.82±1.65
SSC	95.05 ±2.33	84.53±2.57	77.54±2.59

Table 2. Test set error accuracy for face recognition on CMU-multipie data set (left) 15 scene (middle) and Caltech-101 (right) respectively. The performance of the smooth sparse coding approach is better than the standard sparse coding and LLC in all cases.

Dictionary size	15 scene	Caltech-101
1024	84.42±2.01	77.14 ±2.23
2048	87.92±2.35	79.75±1.44
4096	90.22±2.91	81.01±1.17

Table 3. Effect of dictionary size on classification accuracy using smooth sparse coding and marginal regression on 15 scene and Caltech -101 data set.

Dictionary size: In order to demonstrate the use of scalability of the proposed method with respect to dictionary size, we report classification accuracy with increasing dictionary sizes using smooth sparse coding. The main advantage of the proposed marginal regression training method is that one could easily run experiments with larger dictionary sizes, which typically takes a significantly longer time for other algorithms. For both the Caltech-101 and 15-scene data set, classification accuracy increases significantly with increasing dictionary sizes as seen in Table 3.

6.2.2. Action recognition:

We further conducted an experiment on activity recognition from videos with KTH action and YouTube data set (see Appendix). Similar to the static image case, we follow the standard approach for generating sparse representations for videos as in (Wang et al., 2009). We densely sample $16 \times 16 \times 10$ blocks from the video and extract HoG-3d (Kläser et al., 2008) features from the sampled blocks. We then use smooth sparse coding and max-pooling to generate the video representation (dictionary size was fixed at 1024 and cross-validation was used to select the regularization and bandwidth parameters). Previous approaches include sparse coding, vector quantization, and k-means on top of the HoG-3d feature set (see (Wang et al., 2009) for a comprehensive evaluation). As indicated by Table 4, smooth sparse coding results in higher classification accuracy than previously reported state-of-the-art and standard sparse coding on both datasets (see (Wang et al., 2009; Liu et al., 2009) for a description of the alternative techniques).

Cited method	SC	SSC
92.10 (Wang et al., 2009)	92.423	94.393
71.2 (Liu et al., 2009)	72.640	75.022

Table 4. Action recognition (accuracy) for cited method (left), Hog3d+ SC (middle) and Hog3d+ SSC (right): KTH data set(top) YouTube action dataset (bottom).

6.2.3. Discriminatory power

In this section, we describe another experiment that contrasts the codes obtained by sparse coding and smooth sparse coding in the context of a subsequent classification task. As in (Yu et al., 2011), we first compute the codes in both case based on patches and combine it with max-pooling to obtain the image level representation. We then compute the fisher discriminant score (ratio of within-class variance to between-class variance) for each dimension as measures of the discrimination power realized by the representations.

Figure 1, graphs a histogram of the ratio of smooth sparse coding Fisher score over standard sparse coding Fisher score $R(d) = F_1(d)/F_2(d)$ for 15-scene dataset (left) and Youtube dataset (right). Both histograms demonstrate the improved discriminatory power of smooth sparse coding over regular sparse coding.

7. Semi-supervised smooth sparse coding

One of the primary difficulties in some image classification tasks is the lack of availability of labeled data and in some cases, both labeled and unlabeled data (for particular domains). This motivated semi-supervised learning and transfer learning without labels (Raina et al., 2007) respectively. The motivation for such approaches is that data from a related domain might have some visual patterns that might be similar to the problem at hand. Hence, learning a high-level dictionary based on data from a different domains aids the classification task of interest.

We propose that the smooth sparse coding approach might be useful in this setting. The motivation is as

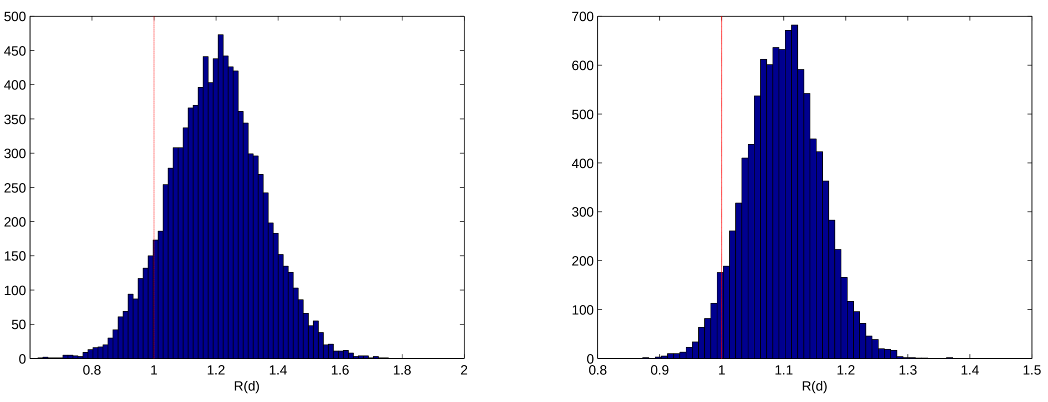

Figure 1. Comparison between the histograms of Fisher discriminant score realized by sparse coding and smooth sparse coding. The images represent the histogram of the ratio of smooth sparse coding Fisher score over standard sparse coding Fisher score (left: image data set; right: video). A value greater than 1 implies that smooth sparse coding is more discriminatory.

follows: in semi-supervised, typically not all samples from a different data set might be useful for the task at hand. Using smooth sparse coding, one can weigh the useful points more than the other points (the weights being calculated based on feature/time similarity kernel) to obtain better dictionaries and sparse representations. Other approach to handle a lower number of labeled samples include collaborative modeling or multi-task approaches which impose a shared structure on the codes for several tasks and use data from all the tasks simultaneously, for example group sparse coding (Bengio et al., 2009). The proposed approach provides an alternative when such collaborative modeling assumptions do not hold, by using relevant unlabeled data samples that might help the task at hand via appropriate weighting.

We now describe an experiment that examines the proposed smoothed sparse coding approach in the context of semi-supervised dictionary learning. We use data from both CMU multi-pie dataset (session 1) and faces-on-tv dataset (treated as frames) to learn a dictionary using a feature similarity kernel. We follow the same procedure described in the previous experiments to construct the dictionary. In the test stage we use the obtained dictionary for coding data from sessions 2, 3, 4 of CMU-multipie data set, using smooth sparse coding. Note that semi-supervision was used only in the dictionary learning stage (the classification stage used supervised SVM).

Table 5 shows the test set error rate and compares it to standard sparse coding and LLC (Yu et al., 2009). Smooth sparse coding achieves significantly lower test error rate than the two alternative techniques. We conclude that the smoothing approach described in this paper may be useful in cases where there is a small set of labeled data, such as semisupervised learning and transfer learning.

Method	SC	LLC	SSC-tricube
Test errror	6.345	6.003	4.975

Table 5. Semi-supervised learning test set error: Dictionary learned from both CMU multi-pie and faces-on-tv data set using feature similarity kernel, used to construct sparse codes for CMU multipie data set.

8. Discussion and Future work

We proposed a simple framework for incorporating similarity in feature space and space or time into sparse coding. We also propose in this paper modifying sparse coding by replacing the lasso optimization stage by marginal regression and adding a constraint to enforce incoherent dictionaries. The resulting algorithm is significantly faster (speedup of about two-orders of magnitude over standard sparse coding). This facilitates scaling up the sparse coding framework to large dictionaries, an area which is usually restricted due to intractable computation.

This work leads to several interesting follow-up directions. On the theoritical side: (i) local convergence of Lasso-based sparse coding has been analyzed in (Jenatton et al., 2012)- preliminary examination suggests that the proposed marginal-regression based sparse coding algorithm might be more favorable for the local convergence analysis and (ii) it is also interesting to explore tighter generalization error bounds by directly analyzing the solutions of the marginal regression iterative algorithm. Methodologically, it is interesting to explore: (i) using an adaptive or non-constant kernel bandwidth to get higher accuracy, and (iv) alternative incoherence constraints that may lead to easier optimization and scaling up.

References

Bartlett, P.L., Bousquet, O., and Mendelson, S. Local rademacher complexities. *The Annals of Statistics*, 2005.

Bengio, S., Pereira, F., Singer, Y., and Strelow, D. Group sparse coding. In *NIPS*, 2009.

Bertsekas, D. On the goldstein-levitin-polyak gradient projection method. *IEEE Transactions on Automatic Control*, 1976.

Bronstein, A., Sprechmann, P., and Sapiro, G. Learning efficient structured sparse models. *ICML*, 2012.

Devroye, L. and Lugosi, G. Combinatorial methods in density estimation. 2001.

Fan, J. and Lv, J. Sure independence screening for ultrahigh dimensional feature space. *JRSS: B(Statistical Methodology)*, 2008.

Genovese, C. R., Jin, J., Wasserman, L., and Yao, Z. A comparison of the lasso and marginal regression. *JMLR*, 2012.

Hastie, T. and Loader, C. Local regression: Automatic kernel carpentry. *Statistical Science*, pp. 120–129, 1993.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. Proximal methods for sparse hierarchical dictionary learning. *ICML*, 2010.

Jenatton, R., Gribonval, R., and Bach, F. Local stability and robustness of sparse dictionary learning in the presence of noise. *arXiv preprint arXiv:1210.0685*, 2012.

Kläser, A., Marszałek, M., and Schmid, C. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

Lee, H., Battle, A., Raina, R., and Ng, A.Y. Efficient sparse coding algorithms. In *NIPS*, 2007.

Liu, J., Luo, J., and Shah, M. Recognizing realistic actions from videos in the wild. In *CVPR*, 2009.

Loader, C. *Local regression and likelihood*. Springer Verlag, 1999.

Lowe, D. G. Object recognition from local scale-invariant features. *CVPR*, 1999.

Magnus, J. R. and Neudecker, H. Matrix differential calculus with applications in statistics and econometrics. 1988.

Meier, L. and Bühlmann, P. Smoothing l1-penalized estimators for high-dimensional time-course data. *Electronic Journal of Statistics*, 2007.

Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A.Y. Self-taught learning: transfer learning from unlabeled data. In *ICML*, 2007.

Ramırez, I., Lecumberry, F., and Sapiro, G. Sparse modeling with universal priors and learned incoherent dictionaries. *Tech Report, IMA, University of Minnesota*, 2009.

Sigg, C. D., Dikk, T., and Buhmann, J. M. Learning dictionaries with bounded self-coherence. *IEEE Transactions on Signal Processing*, 2012.

Solodov, M. V. Convergence analysis of perturbed feasible descent methods. *Journal of Optimization Theory and Applications*, 1997.

Tropp, J.A. Greed is good: Algorithmic results for sparse approximation. *Information Theory, IEEE*, 2004.

Vainsencher, D., Mannor, S., and Bruckstein, A.M. The sample complexity of dictionary learning. *JMLR*, 2011.

Wang, H., Ullah, M.M., Klaser, A., Laptev, I., and Schmid, C. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

Yang, J., Yu, K., and Huang, T. Supervised translation-invariant sparse coding. In *CVPR*, 2010.

Yu, K., Zhang, T., and Gong, Y. Nonlinear learning using local coordinate coding. *NIPS*, 2009.

Yu, K., Lin, Y., and Lafferty, J. Learning image representations from the pixel level via hierarchical sparse coding. In *CVPR*, 2011.

Zangwill, W.I. Nonlinear programming: a unified approach. 1969.