Sparse coding for multitask and transfer learning

Andreas Maurer AM@ANDREAS-MAURER.EU
Adalbertstrasse 55, D-80799, Munchen, Germany

Massimiliano Pontil M.PONTIL@CS.UCL.AC.UK
Department of Computer Science and Centre for Computational Statistics and Machine Learning
University College London, Malet Place, London WC1E 6BT, UK

Bernardino Romera-Paredes BERNARDINO.PAREDES.09@UCL.AC.UK
Department of Computer Science and UCL Interactive Centre
University College London, Malet Place, London WC1E 6BT, UK

Abstract

We investigate the use of sparse coding and dictionary learning in the context of multitask and transfer learning. The central assumption of our learning method is that the tasks parameters are well approximated by sparse linear combinations of the atoms of a dictionary on a high or infinite dimensional space. This assumption, together with the large quantity of available data in the multitask and transfer learning settings, allows a principled choice of the dictionary. We provide bounds on the generalization error of this approach, for both settings. Numerical experiments on one synthetic and two real datasets show the advantage of our method over single task learning, a previous method based on orthogonal and dense representation of the tasks and a related method learning task grouping.

1. Introduction

The last decade has witnessed many efforts of the machine learning community to exploit assumptions of sparsity in the design of algorithms. A central development in this respect is the Lasso (Tibshirani, 1996), which estimates a linear predictor in a high dimensional space under a regularizing ℓ_1-penalty. Theoretical results guarantee a good performance of this method under the assumption that the vector corresponding to the underlying predictor is sparse, or at least has a small ℓ_1-norm, see e.g. (Bühlmann & van de Geer, 2011) and references therein.

In this work we consider the case where the predictors are linear combinations of the atoms of a dictionary of linear functions on a high or infinite dimensional space, and we assume that we are free to choose the dictionary. We will show that a principled choice is possible, if there are many learning problems, or "tasks", and there exists a dictionary allowing sparse, or nearly sparse representations of all or most of the underlying predictors. In such a case we can exploit the larger quantity of available data to estimate the "good" dictionary and still reap the benefits of the Lasso for the individual tasks. This paper gives theoretical and experimental justification of this claim, both in the domain of multitask learning, where the new representation is applied to the tasks from which it was generated, and in the domain of learning to learn, where the dictionary is applied to new tasks of the same environment.

Our work combines ideas from sparse coding (Olshausen & Field, 1996), multitask learning (Ando & Zhang, 2005; Argyriou, Evgeniou, Pontil, 2008; Argyriou, Maurer, Pontil, 2008; Ben-David & Schuller, 2003; Caruana, 1997; Evgeniou, Micchelli, Pontil, 2005; Maurer, 2009) and learning to learn (Baxter, 2000; Thrun & Pratt, 1998). There is a vast literature on these subjects and the list of papers provided here is necessarily incomplete. Learning to learn (also called inductive bias learning or transfer learning) has been proposed by Baxter (2000) and an error analysis is provided therein, showing that a common representation which performs well on the training tasks will also generalize to new tasks obtained from the same "environment". The precursors of the analysis pre-

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

sented here are (Maurer & Pontil, 2010) and (Maurer, 2009). The first paper provides a bound on the reconstruction error of sparse coding and may be seen as a special case of the ideas presented here when the sample size is infinite. The second paper provides a learning to learn analysis of the multitask feature learning method in (Argyriou, Evgeniou, Pontil, 2008).

We note that a method similar to the one presented in this paper has been recently proposed within the multitask learning setting (Kumar & Daumé III, 2012). Here we highlight the connection between sparse coding and multitask learning and present a probabilistic analysis which complements well with the practical insights in the above work. We also address the different problem of learning to learn, demonstrating the utility of our approach in this setting by means of both learning bounds and numerical experiments. A further novelty of our approach is that it applies to a Hilbert spaces setting, thereby providing the possibility of learning nonlinear predictors using reproducing kernel Hilbert spaces.

The paper is organized in the following manner. In Section 2, we set up our notation and introduce the learning problem. In Section 3, we present our learning bounds for multitask learning and learning to learn. In Section 4 we report on numerical experiments. Section 5 contains concluding remarks.

2. Method

In this section, we turn to a technical exposition of the proposed method, introducing some necessary notation on the way.

Let H be a finite or infinite dimensional Hilbert space with inner product $\langle \cdot, \cdot \rangle$, norm $\|\cdot\|$, and fix an integer K. We study the problem

$$\min_{D \in \mathcal{D}_K} \frac{1}{T} \sum_{t=1}^{T} \min_{\gamma \in \mathcal{C}_\alpha} \frac{1}{m} \sum_{i=1}^{m} \ell\left(\langle D\gamma, x_{ti} \rangle, y_{ti}\right), \qquad (1)$$

where

- $\mathcal{D}_K$ is the set of K-dimensional dictionaries (or simply dictionaries), which means that every $D \in \mathcal{D}_K$ is a linear map $D : \mathbb{R}^K \to H$, such that $\|De_k\| \leq 1$ for every one of the canonical basis vectors e_k of $\mathbb{R}^K$. The number K can be regarded as one of the regularization parameters of our method.
- $\mathcal{C}_\alpha$ is the set of code vectors γ in $\mathbb{R}^K$ satisfying $\|\gamma\|_1 \leq \alpha$. The ℓ_1-norm constraint implements the assumption of sparsity and α is the other regularization parameter. Different sets $\mathcal{C}_\alpha$ could be readily used in our method, such as those associated with ℓ_p-norms.
- $\mathbf{Z} = ((x_{ti}, y_{ti}) : 1 \leq i \leq m, 1 \leq t \leq T)$ is a dataset on which our algorithm operates. Each $x_{ti} \in H$ represents an input vector, and y_{ti} is a corresponding real valued label. We also write $\mathbf{Z} = (\mathbf{X}, \mathbf{Y}) = (\mathbf{z}_1, \ldots, \mathbf{z}_T) = ((\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_T, \mathbf{y}_T))$ with $\mathbf{x}_t = (x_{t1}, \ldots, x_{tm})$ and $\mathbf{y}_t = (y_{t1}, \ldots, y_{tm})$. The index t identifies a learning task, and $\mathbf{z}_t$ are the corresponding training points, so the algorithm operates on T tasks, each of which is represented by m example pairs.
- ℓ is a loss function where $\ell(y, y')$ measures the loss incurred by predicting y when the true label is y'. We assume that ℓ has values in $[0, 1]$ and has Lipschitz constant L in the first argument for all values of the second argument.

The minimum in (1) is zero if the data is generated according to a noise-less model which postulates that there is a "true" dictionary $D^* \in \mathcal{D}_{K^*}$ with K^* atoms and vectors $\gamma_1^*, \ldots, \gamma_T^*$ satisfying $\|\gamma_t^*\|_1 \leq \alpha^*$, such that an input $x \in H$ generates the label $y = \langle D^* \gamma_t^*, x \rangle$ in the context of task t. If $K \geq K^*$ and $\alpha \geq \alpha^*$ then the minimum in (1) is zero. In Section 4, we will present experiments with such a generative model, when noise is added to the labels, that is $y = \langle D^* \gamma_t^*, x \rangle + \zeta$ with $\zeta \sim \mathcal{N}(0, \sigma)$, the standard normal distribution.

The method (1) should output a minimizing $D(\mathbf{Z}) \in \mathcal{D}_K$ as well as a minimizing $\gamma_1(\mathbf{Z}), \ldots, \gamma_T(\mathbf{Z})$ corresponding to the different tasks. Our implementation, described in Section 4.1, does not guarantee exact minimization, because of the non-convexity of the problem. Below predictors are always linear, specified by a vector $w \in H$, predicting the label $\langle w, x \rangle$ for an input $x \in H$, and a learning algorithm is a rule which assigns a predictor $A(\mathbf{z})$ to a given data set $\mathbf{z} = ((x_i, y_i) : 1 \leq i \leq m) \in (H \times \mathbb{R})^m$.

3. Learning bounds

In this section, we present learning bounds for method (1), both in the multitask learning and learning to learn settings, and discuss the special case of sparse coding.

3.1. Multitask learning

Let $\mu_1, \ldots, \mu_T$ be probability measures on $H \times \mathbb{R}$. We interpret $\mu_t(x, y)$ as the probability of observing the input/output pair (x, y) in the context of task t. For each of these tasks an i.i.d. training

sample $\mathbf{z}_t = ((x_{ti}, y_{ti}) : 1 \le i \le m)$ is drawn from $(\mu_t)^m$ and the ensemble $\mathbf{Z} \sim \prod_{t=1}^T \mu_t^m$ is input to algorithm (1). Upon returning of a minimizing $D(\mathbf{Z})$ and $\gamma_1(\mathbf{Z}), \ldots, \gamma_T(\mathbf{Z})$, we will use the predictor $D(\mathbf{Z})\gamma_t(\mathbf{Z})$ on the t-th task. The average over all tasks of the expected error incurred by these predictors is

$$\frac{1}{T}\sum_{t=1}^T \mathbb{E}_{(x,y)\sim\mu_t}\left[\ell\left(\langle D(\mathbf{Z})\gamma_t(\mathbf{Z}), x\rangle, y\right)\right].$$

We compare this *task-average risk* to the minimal analogous risk obtainable by any dictionary $D \in \mathcal{D}_K$ and any set of vectors $\gamma_1, \ldots, \gamma_T \in \mathcal{C}_\alpha$. Our first result is a bound on the excess risk.

Theorem 1. *Let $\delta > 0$ and let $\mu_1, \ldots, \mu_T$ be probability measures on $H \times \mathbb{R}$. With probability at least $1-\delta$ in the draw of $\mathbf{Z} \sim \prod_{t=1}^T \mu_t^m$ we have*

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^T \mathbb{E}_{(x,y)\sim\mu_t}\left[\ell\left(\langle D(\mathbf{Z})\gamma_t(\mathbf{Z}), x\rangle, y\right)\right] \\
&\quad - \inf_{D\in\mathcal{D}_K}\frac{1}{T}\sum_{t=1}^T \inf_{\gamma\in\mathcal{C}_\alpha}\mathbb{E}_{(x,y)\sim\mu_t}\left[\ell\left(\langle D\gamma, x\rangle, y\right)\right] \\
&\qquad \le L\alpha\sqrt{\frac{2S_1(\mathbf{X})(K+12)}{mT}} \\
&\qquad + L\alpha\sqrt{\frac{8S_\infty(\mathbf{X})\ln(2K)}{m}} + \sqrt{\frac{8\ln 4/\delta}{mT}},
\end{aligned}$$

where $S_1(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^T \operatorname{tr}\left(\hat{\Sigma}(\mathbf{x}_t)\right)$ and $S_\infty(\mathbf{X}) = \frac{1}{T}\sum_{t=1}^T \lambda_{\max}\left(\hat{\Sigma}(\mathbf{x}_t)\right)$. Here $\hat{\Sigma}(\mathbf{x}_t)$ is the empirical covariance of the input data for the t-th task, $\operatorname{tr}(\cdot)$ denotes the trace and $\lambda_{\max}(\cdot)$ the largest eigenvalue.

We state several implications of this theorem.

1. The quantity $S_1(\mathbf{X})$ appearing in the bound is just the average square norm of the input data points, while $S_\infty(\mathbf{X})$ is roughly the average inverse of the observed dimension of the data for each task. Suppose that $H = \mathbb{R}^d$ and that the data-distribution is uniform on the surface of the unit ball. Then $S_1(\mathbf{X}) = 1$ and for $m \ll d$ it follows from Levy's isoperimetric inequality (see e.g. (Ledoux & Talagrand, 1991)) that $S_\infty(\mathbf{X}) \approx 1/m$, so the corresponding term behaves like $\sqrt{\ln K}/m$. If the minimum in (1) is small and T is large enough for this term to become dominant then there is a significant advantage of the method over learning the tasks independently. If the data is essentially low dimensional, then $S_\infty(\mathbf{X})$ will be large, and in the extreme case, if the data is one-dimensional for all tasks then $S_\infty(\mathbf{X}) = S_1(\mathbf{X})$ and our bound will always be worse by a factor of $\ln K$ than standard bounds for independent single task learning as in (Bartlett & Mendelson, 2002). This makes sense, because for low dimensional data there can be little advantage to multitask learning.

2. In the regime $T < K$ the bound is dominated by the term of order $\sqrt{S_1(\mathbf{X})K/mT} > \sqrt{S_1(\mathbf{X})/m}$. This is easy to understand, because the dictionary atoms De_k can be chosen independently, separately for each task, so we could at best recover the usual bound for linear models and there is no benefit from multitask learning.

3. Consider the noiseless generative model mentioned in Section 2. If $K \ge K^*$ and $\alpha \ge \alpha^*$ then the minimum in (1) is zero. In the bound the overestimation of K^* can be compensated by a proportional increase in the number of tasks considered and an only very minor increase of the sample size m, namely $m \to (\ln K^*/\ln K)\, m$.

4. Suppose that we concatenate two sets of tasks. If the tasks are generated by the model described in Section 2 then the resulting set of tasks is also generated by such a model, obtained by concatenating the lists of atoms of the two true dictionaries D_1^* and D_2^* to obtain the new dictionary D^* of length $K^* = K_1^* + K_2^*$ and taking the union of the set of generating vectors $\{\gamma_t^{*1}\}_{t=1}^T$ and $\{\gamma_t^{*2}\}_{t=1}^T$, extending them to $\mathbb{R}^{K_1^*+K_2^*}$ so that the supports of the first group are disjoint from the supports of the second group. If $T_1 = T_2$, $K_1^* = K_2^*$ and we train with the correct parameters, then the excess risk for the total task set increases only by the order of $1/\sqrt{m}$, independent of K, despite the fact that the tasks in the second group are in no way related to those in the first group. Our method has the property of finding the right clusters of mutually related tasks.

5. Consider the alternative method of subspace learning (SL) where $\mathcal{C}_\alpha$ is replaced by an euclidean ball of radius α. With similar methods one can prove a bound for SL where, apart from slightly different constants, $\sqrt{\ln K}$ above is replaced by K. SL will be successful and outperform the proposed method, whenever K can be chosen small, with $K < m$ and the vector γ_t^* utilize the entire span of the dictionary. For large values of K, a correspondingly large number of tasks and sparse γ_t^* the proposed method will be superior.

The proof of Theorem 1, which is given in Section B.1

of the supplementary appendix, uses standard methods of empirical process theory, but also employs a concentration result related to Talagrand's convex distance inequality to obtain the crucial dependence on $S_\infty(\mathbf{X})$. At the end of Section B.1 we sketch applications of the proof method to other regularization schemes, such as the one presented in (Kumar & Daumé III, 2012), in which the Frobenius norm on the dictionary D is used in place of the ℓ_2/ℓ_∞-norm employed here and the ℓ_1/ℓ_1 norm on the coefficient matrix $[\gamma_1, \ldots, \gamma_T]$ is used in place of the ℓ_1/ℓ_∞.

3.2. Learning to learn

There is no absolute way to assess the quality of a learning algorithm. Algorithms may perform well on one kind of task, but poorly on another kind. It is important that an algorithm performs well on those tasks which it is likely to be applied to. To formalize this, Baxter (2000) introduced the notion of an *environment*, which is a probability measure $\mathcal{E}$ on the set of tasks. Thus $\mathcal{E}(\tau)$ is the probability of encountering the task τ in the environment $\mathcal{E}$, and $\mu_\tau(x, y)$ is the probability of finding the pair (x, y) in the context of the task τ.

Given $\mathcal{E}$, the *transfer risk* (or simply risk) of a learning algorithm A is defined as follows. We draw a task from the environment, $\tau \sim \mathcal{E}$, which fixes a corresponding distribution μ_τ on $H \times \mathbb{R}$. Then we draw a training sample $\mathbf{z} \sim \mu_\tau^m$ and use the algorithm to compute the predictor $A(\mathbf{z})$. Finally we measure the performance of this predictor on test points $(x, y) \sim \mu_\tau$. The corresponding definition of the transfer risk of A reads as

$$R_{\mathcal{E}}(A) = \mathbb{E}_{\tau\sim\mathcal{E}}\mathbb{E}_{\mathbf{z}\sim\mu_\tau^m}\mathbb{E}_{(x,y)\sim\mu_\tau}\left[\ell\left(\langle A(\mathbf{z}), x\rangle, y\right)\right] \quad (2)$$

which is simply the expected loss incurred by the use of the algorithm A on tasks drawn from the environment $\mathcal{E}$.

For any given dictionary $D \in \mathcal{D}_K$ we consider the learning algorithm A_D, which for $\mathbf{z} \in \mathcal{Z}^m$ computes the predictor

$$A_D(\mathbf{z}) = D \ \arg\min_{\gamma\in\mathcal{C}_\alpha} \frac{1}{m}\sum_{i=1}^m \ell\left(\langle D\gamma, x_i\rangle, y_i\right). \quad (3)$$

Equivalently, we can regard A_D as the Lasso operating on data preprocessed by the linear map $D^\top$, the adjoint of D.

We can make a single observation of the environment $\mathcal{E}$ in the following way: one first draws a task $\tau \sim \mathcal{E}$. This task and the corresponding distribution μ_τ are then observed by drawing an i.i.d. sample $\mathbf{z}$ from μ_τ, that is $\mathbf{z} \sim \mu_\tau^m$. For simplicity the sample size m will be fixed. Such an observation corresponds to the draw of a sample $\mathbf{z}$ from a probability distribution $\rho_{\mathcal{E}}$ on $(H \times \mathbb{R})^m$ which is defined by

$$\rho_{\mathcal{E}}(\mathbf{z}) := \mathbb{E}_{\tau\sim\mathcal{E}}\left[(\mu_\tau)^m(\mathbf{z})\right]. \quad (4)$$

To estimate an environment a large number T of independent observations is needed, corresponding to a vector $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_T) \in ((H \times \mathbb{R})^m)^T$ drawn i.i.d. from $\rho_{\mathcal{E}}$, that is $\mathbf{Z} \sim (\rho_{\mathcal{E}})^T$.

We now propose to solve the problem (1) with the data $\mathbf{Z}$, ignore the resulting $\gamma_i(\mathbf{Z})$, but retain the dictionary $D(\mathbf{Z})$ and use the algorithm $A_{D(\mathbf{Z})}$ on future tasks drawn from the same environment. The performance of this method can be quantified as the transfer risk $R_{\mathcal{E}}\left(A_{D(\mathbf{Z})}\right)$ as defined in equation (2) and again we are interested in comparing this to the risk of an ideal solution based on complete knowledge of the environment. For any fixed dictionary D and task τ the best we can do is to choose $\gamma \in \mathcal{C}$ so as to minimize $\mathbb{E}_{(x,y)\sim\mu_\tau}\left[\ell\left(\langle D\gamma, x\rangle, y\right)\right]$, so the best is to choose D so as to minimize the average of this over $\tau \sim \mathcal{E}$. The quantity

$$R_{\mathrm{opt}} = \min_{D\in\mathcal{D}_K} \mathbb{E}_{\tau\sim\mathcal{E}} \min_{\gamma\in\mathcal{C}_\alpha} \mathbb{E}_{(x,y)\sim\mu_\tau}\ell\left[\left(\langle D\gamma, x\rangle, y\right)\right]$$

thus describes the optimal performance achievable under the given constraint. Our second result is

Theorem 2. *With probability at least* $1-\delta$ *in the multisample* $\mathbf{Z} = (\mathbf{X}, \mathbf{Y}) \sim \rho_{\mathcal{E}}^T$ *we have*

$$R_{\mathcal{E}}\left(A_{D(\mathbf{Z})}\right) - R_{\mathrm{opt}} \le L\alpha K\sqrt{\frac{2\pi S_1(\mathbf{X})}{T}}$$

$$+4L\alpha\sqrt{\frac{S_\infty(\mathcal{E})(2+\ln K)}{m}} + \sqrt{\frac{8\ln 4/\delta}{T}},$$

where $S_1(\mathbf{X})$ *is as in Theorem 1 and* $S_\infty(\mathcal{E}) := \mathbb{E}_{\tau\sim\mathcal{E}}\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu_\tau^m}\lambda_{\max}\left(\hat{\Sigma}(\mathbf{x})\right)$.

We discuss some implications of the above theorem. 1.

1. The interpretation of $S_\infty(\mathcal{E})$ is analogous to that of $S_\infty(\mathbf{X})$ in the bound for Theorem 1. The same applies to Remark 6 following Theorem 1.

2. In the regime $T \le K^2$ the result does not imply any useful behaviour. On the other and, if $T \gg K^2$ the dominant term in the bound is of order $\sqrt{S_\infty(\mathcal{E})/m}$.

3. There is an important difference with the multitask learning bound, namely in Theorem 2 we have $\sqrt{T}$ in the denominator of the first term of

the excess risk, and not $\sqrt{mT}$ as in Theorem 1. This is because in the setting of learning to learn there is always a possibility of being misled by the draw of the training tasks. This possibility can only decrease as T increases – increasing m does not help.

The proof of Theorem 2 is given in Section B.2 of the supplementary appendix and follows the method outlined in (Maurer, 2009): one first bounds the estimation error for the expected empirical risk on future tasks, and then combines this with a bound of the expected true risk by said expected empirical risk. The term $K/\sqrt{T}$ may be an artefact of our method of proof and the conjecture that it can be replaced by $\sqrt{K/T}$ seems plausible.

3.3. Connection to sparse coding

We discuss a special case of Theorem 2 in the limit $m \to \infty$, showing that it subsumes the sparse coding result in (Maurer & Pontil, 2010). To this end, we assume the noiseless generative model $y_{ti} = \langle w_t, x_{ti} \rangle$ described in Section 2, that is $\mu(x, y) = p(x)\delta(y, \langle w, x \rangle)$, where p is the uniform distribution on the sphere in $\mathbb{R}^d$ (i.e. the Haar measure). In this case the environment of tasks is fully specified by a measure ρ on the unit ball in $\mathbb{R}^d$ from which a task $w \in \mathbb{R}^d$ is drawn and the measure μ is identified with the vector w. Note that we do not assume that these tasks are obtained as sparse combinations of some dictionary. Under the above assumptions and choosing ℓ to be the square loss, we have that $\mathbb{E}_{(x,y)\sim\mu_t} \ell(\langle w, x \rangle, y) = \|w_t - w\|^2$. Consequently, in the limit of $m \to \infty$ method (1) reduces to a constrained version of sparse coding (Olshausen & Field, 1996), namely

$$\min_{D \in \mathcal{D}_K} \frac{1}{T} \sum_{t=1}^{T} \min_{\gamma \in \mathcal{C}_\alpha} \|D\gamma - w_t\|^2.$$

In turn, the transfer error of a dictionary D is given by the quantity $R(D) := \min_{\gamma \in \mathcal{C}_\alpha} \|D\gamma - w\|^2$ and $R_{\text{opt}} = \min_{D \in \mathcal{D}_K} \mathbb{E}_{w\sim\rho} \min_{\gamma \in \mathcal{C}_\alpha} \|D\gamma - w\|^2$. Given the constraints $D \in \mathcal{D}_K$, $\gamma \in \mathcal{C}_\alpha$ and $\|x\| \leq 1$, the square loss $\ell(y, y') = (y - y')^2$, evaluated at $y = \langle D\gamma, x \rangle$, can be restricted to the interval $y \in [-\alpha, \alpha]$, where it has the Lipschitz constant $2(1 + \alpha)$ for any $y' \in [-1, 1]$, as is easily verified. Since $S_1(\mathbf{X}) = 1$ and $S_\infty(\mathcal{E}) < \infty$, the bound in Theorem 2 becomes

$$R(D) - R_{\text{opt}} \leq 2\alpha(1 + \alpha)K\sqrt{\frac{2\pi}{T}} + 8\sqrt{\frac{\ln 4/\delta}{T}} \quad (5)$$

in the limit $m \to \infty$. The typical choice for α is $\alpha \leq 1$, which ensures that $\|D\gamma\| \leq 1$. In this case inequality (5) provides an improvement over the sparse coding bound in (Maurer & Pontil, 2010) (cf. Theorem 2 and Section 2.4 therein), which contains an additional term of the order of $\sqrt{(\ln T)/T}$ and the same leading term in K as in (5) but with slightly worse constant (14 instead of $4\sqrt{2\pi}$). The connection of our method to sparse coding is experimentally demonstrated in Section 4.4 and illustrated in Figure 6.

4. Experiments

In this section, we present experiments on a synthetic and two real datasets. The aim of the experiments is to study the statistical performance of the proposed method, in both settings of multitask learning and learning to learn. We compare our method, denoted as Sparse Coding Multi Task Learning (SC-MTL), with independent ridge regression (RR) as a base line and multitask feature learning (MTFL) (Argyriou, Evgeniou, Pontil, 2008) and GO-MTL (Kumar & Daumé III, 2012). We also report on sensitivity analysis of the proposed method versus different number of parameters involved.

4.1. Optimization algorithm

We solve problem (1) by alternating minimization over the dictionary matrix D and the code vectors $\boldsymbol{\gamma}$. The techniques we use are very similar to standard methods for sparse coding and dictionary learning, see e.g. (Jenatton et al., 2011) and references therein for more information. Briefly, assuming that the loss function ℓ is convex and has Lipschitz continuous gradient, either minimization problem is convex and can be solved efficiently by proximal gradient methods, see e.g. (Beck & Teboulle, 2009; Combettes & Wajs, 2006). The key ingredient in each step is the computation of the proximity operator, which in either problem has a closed form expression.

4.2. Toy experiment

We generated a synthetic environment of tasks as follows. We choose a $d \times K$ matrix D by sampling its columns independently from the uniform distribution on the unit sphere in $\mathbb{R}^d$. Once D is created, a generic task in the environment is given by $w = D\gamma$, where γ is an s-sparse vector obtained as follows. First, we generate a set $J \subseteq \{1, \ldots, K\}$ of cardinality s, whose elements (indices) are sampled uniformly without replacement from the set $\{1, \ldots, K\}$. We then set $\gamma_j = 0$ if $j \notin J$ and otherwise sample $\gamma_j \sim \mathcal{N}(0, 0.1)$. Finally, we normalize γ so that it has ℓ_1-norm equal to some prescribed value α. Using the above procedure we generated T tasks $w_t = D\gamma_t$, $t = 1, \ldots, T$.

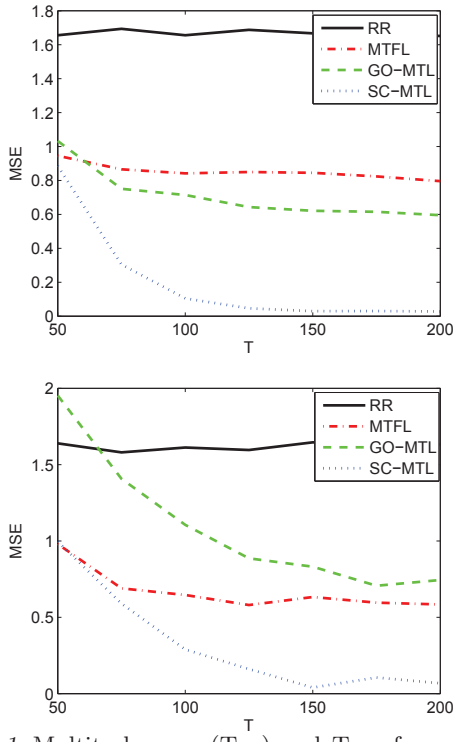

Figure 1. Multitask error (Top) and Transfer error (Bottom) vs. number of training tasks T.

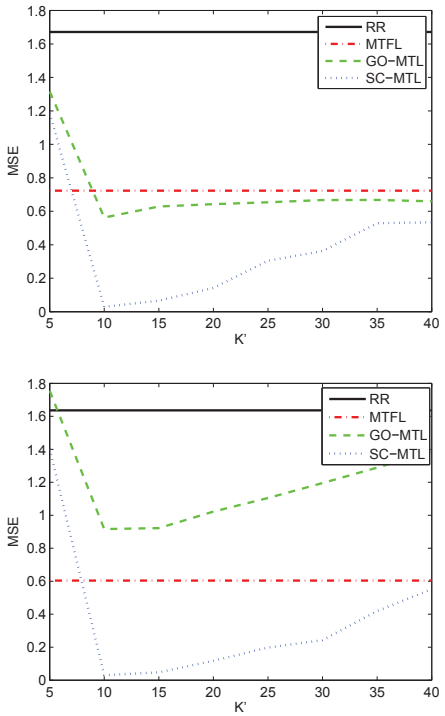

Figure 2. Multitask error (Top) and Transfer error (Bottom) vs. number of atoms K' used by dictionary-based methods.

Further, for each task t we generated a training set $\mathbf{z}_t = \{(x_{ti}, y_{ti})\}_{i=1}^m$, sampling x_{ti} i.i.d. from the uniform distribution on the unit sphere in $\mathbb{R}^d$. We then set $y_{ti} = \langle w_t, x_{ti}\rangle + \xi_{ti}$, with $\xi_{ti} \sim \mathcal{N}(0, \sigma^2)$, where σ is the variance of the noise. This procedure also defines the generation of new tasks in the transfer learning experiments below.

The above model depends on seven parameters: the number K and the dimension d of the atoms, the sparsity s and the ℓ_1-norm α of the codes, the noise level σ, the sample size per task m and the number of training tasks T. In all experiments we report both the multitask learning (MTL) and learning to learn (LTL) performance of the methods. For MTL, we measure performance by the estimation error $1/T \sum_{t=1}^T \|w_t - \hat{w}_t\|^2$, where $\hat{w}_1, \ldots, \hat{w}_T$ are the estimated task vectors (in the case of SC-MTL, $\hat{w}_t = D(\mathbf{Z})\gamma(\mathbf{Z})_t$ – see the discussion in Section 2. For LTL, we use the same quantity but with a new set of tasks generated by the environment (in the experiment below we generate 100 new tasks). The regularization parameter of each method is chosen by cross validation. Finally, all experiments are repeated 50 times, and the average performance results are reported in the plots below.

In the first experiment, we fix $K = 10, d = 20, s = 2, \alpha = 10, m = 10, \sigma = 0.1$ and study the statistical performance of the methods as a function of the number of tasks. The results, shown in Figure 1, clearly indicate that the proposed method outperforms the remaining approaches. In this experiment the number of atoms used by dictionary-based approaches, which here we denote by K' to avoid confusion with the number of atoms K of the target dictionary, was equal to $K = 10$. This gives an advantage to both GO-MTL and SC-MTL. We therefore also studied the performance of those methods in dependence on K'. Figure 2, reporting this result, is in qualitative agreement with our theoretical analysis: the performance of SC-MTL is not too sensitive to K' if $K' \geq K$, and the method still outperforms independent RR and MTFL if $K' = 4K$. On the other hand if $K' < K$ the performance of the method quickly degrades. In the last experiment we study performance vs. the sparsity ratio s/K. Intuitively we would expect our method to have greater advantage over MTL if $s \ll K$. The results, shown in Figure 3, confirm this fact, also indicating that SC-MTL is outperformed by both GO-MTL and MTFL as sparsity becomes less pronounced ($s/K > 0.6$).

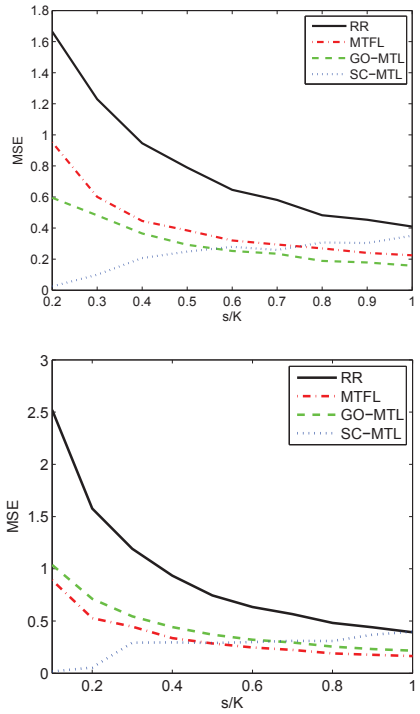

Figure 3. Multitask error (Top) and Transfer error (Bottom) vs. sparsity ratio s/K.

4.3. Learning to learn optical character recognition

We have conducted experiments on real data to study the performance of our method in a learning to learn / transfer learning setting. To this end, we employed the NIST dataset[1], which is composed of a set of 14×14 pixels images of handwritten characters (digits and lower and capital case letters, for a total of 52 characters).

We considered the following experimental protocol. First, a set of 20 characters are chosen randomly as well as n instances for each character. These are used to learn all possibilities of 1-vs-1 train tasks, which makes $T = 190$, each of which having $m = 2n$ instances. The knowledge learned in this stage is employed to learn another set of target tasks. In our approach, the assumption that is made is that some of the components in the dictionary learned from the training tasks, can also be useful for representing the target tasks. In order to create the target tasks, another set of 10 characters are chosen among the remaining set of characters in the dataset, inducing a set of 45 1-vs-1 classification tasks. Since we are interested in the case where the training set size of the target tasks is small, we sample only 3 instances for each character, hence 6 examples per task.

[1]The NIST dataset is available at http://www.nist.gov/srd/nistsd19.cfm

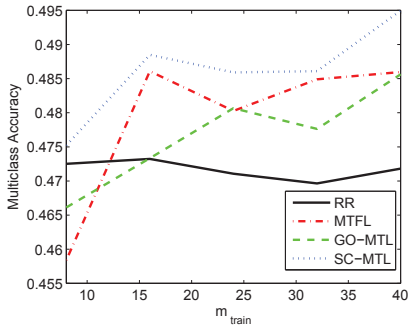

Figure 4. Multiclassification accuracy of RR, MTFL GO-MTL and SC-MTL vs. the number of training instances in the transfer tasks, m.

In order to tune the hyperparameters of all compared approaches, we have also created another set of 45 validation tasks by following the process previously described, simulating the target set of tasks. Note that there is not overlapping between the digits associated to the train, target and validation tasks.

We have run 50 trials of the above process for different values of m and the average multiclass accuracy on the target tasks is reported in Figure 4.

4.4. Sparse coding of images with missing pixels

In the last experiment we consider a sparse coding problem (Olshausen & Field, 1996) of optical character images, with missing pixels. We employ the Binary Alphadigits dataset[2], which is composed of a set of binary 20×16 images of all digits and capital letters (39 images for each character). In the following experiment only the digits are used. We regard each image as a task, hence the input space is the set of 320 possible pixels indices, while the output space is the real interval $[0, 1]$, representing the gray level. We sample $T = 100, 130, 160, 190, 220, 250$ images, equally divided among the 10 possible digits. For each of these, a corresponding random set of $m = 160$ pixel values are sampled (so the set of sample pixels varies from one image to another).

We test the performance of the dictionary learned by method (1) in a learning to learn setting, by choosing 100 new images. The regularization parameter for each approach is tuned using cross validation. The results, shown in Figure 5, indicate some advantage of the pro-

[2]Available at *http://www.cs.nyu.edu/ roweis/data.html*.

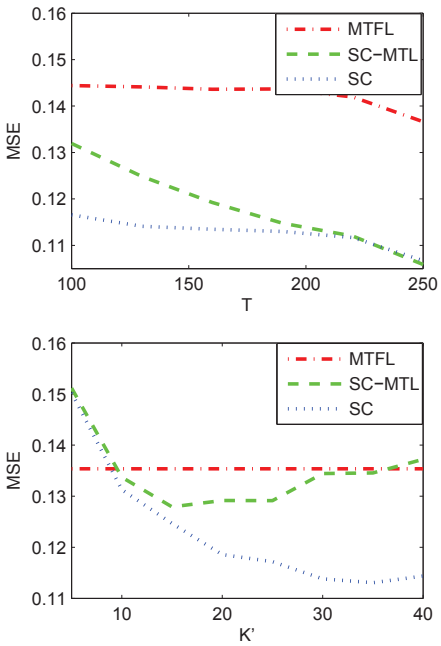

Figure 5. Transfer error vs. number of tasks T (Top) and vs. number of atoms K (Bottom) on the Binary Alphadigits dataset.

posed method over trace norm regularization. A similar trend, not reported here due to space constraints, is obtained in the multitask setting. Ridge regression performed significantly worse and is not shown in the figure. We also show as a reference the performance of sparse coding (SC) applied when all pixels are known.

With the aim of analyzing the atoms learned by the algorithm, we have carried out another experiment where we assume that there are 10 underlying atoms (one for each digit). We compare the resultant dictionary to that obtained by sparse coding, where all pixels are known. The results are shown in Figure 6.

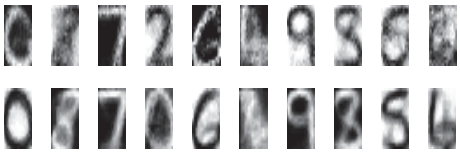

Figure 6. Dictionaries found by SC-MTL using $m = 240$ pixels (missing 25% pixels) per image (top) and by Sparse Coding employing all pixels (bottom).

5. Summary

In this paper, we have explored an application of sparse coding, which has been widely used in unsupervised learning and signal processing, to the domains of multitask learning and learning to learn. Our learning bounds provide a justification of this method and offer insights into its advantage over independent task learning and learning dense representation of the tasks. The bounds, which hold in a Hilbert space setting, depend on data dependent quantities which measure the intrinsic dimensionality of the data. Numerical simulations presented here indicate that sparse coding is a promising approach to multitask learning and can lead to significant improvements over competing methods.

In the future, it would be valuable to study extensions of our analysis to more general classes of code vectors. For example, we could use code sets $\mathcal{C}_\alpha$ which arise from structured sparsity norms, such as the group Lasso, see e.g. (Jenatton et al., 2011; Lounici et al., 2011) or other families of regularizers. A concrete example which comes to mind is to choose $K = Qr$, $Q, r \in \mathbb{N}$ and a partition $\mathcal{J} = \{\{(q-1)r+1, \ldots, qr\} : q = 1, \ldots, Q\}$ of the index set $\{1, \ldots, K\}$ into contiguous index sets of size r. Then using a norm of the type $\|\gamma\| = \|\gamma\|_1 + \sum_{J\in\mathcal{J}} \|\gamma_J\|_2$ will encourage codes which are sparse and use only few of the groups in $\mathcal{J}$. Using the ball associated with this norm as our set of codes would allow to model sets of tasks which are divided into groups. A further natural extension of our method is nonlinear dictionary learning in which the dictionary columns correspond to functions in a reproducing kernel Hilbert space and the tasks are expressed as sparse linear combinations of such functions.

Acknowledgments

This work was supported in part by EPSRC Grant EP/H027203/1 and Royal Society International Joint Project 2012/R2.

References

Ando, R.K. and Zhang, T. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. of Machine Learning Research*, 6:1817–1853, 2005.

Argyriou, A., Evgeniou, T., and Pontil, M. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

Argyriou, A., Maurer, A., and Pontil, M. An algorithm for transfer learning in a heterogeneous environment. *Proc. European Conf. Machine Learning*, pp. 71–85, 2008.

Bartlett, P.L. and Mendelson, S. Rademacher and gaussian complexities: risk bounds and structural results. *J. of Machine Learning Research*, 3:463–482, 2002.

Baxter, J. A model for inductive bias learning. *J. of Artificial Intelligence Research*, 12:149–198, 2000.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal of Imaging Sciences*, 2(1):183–202, 2009.

Ben-David, S. and Schuller, R. Exploiting task relatedness for multiple task learning. *Proceedings of Computational Learning Theory (COLT)*, 2003.

Bühlmann, P. and van de Geer, S. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, 2011.

Caruana, R. Multi-task learning. *Machine Learning*, 28:41–75, 1997.

Combettes, P.L. and Wajs, V.R. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2006.

Evgeniou, T., Micchelli, C.A., and Pontil, M. Learning multiple tasks with kernel methods. *J. of Machine Learning Research*, 6:615–637, 2005.

Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. Proximal methods for hierarchical sparse coding. *J. of Machine Learning Research*, 12:2297–2334, 2011.

Koltchinskii, V. and Panchenko, D. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30(1):1–50, 2002.

Kumar, A. and Daumé III, H. Learning task grouping and overlap in multitask learning. *International Conference on Machine Learning (ICML)*, 2012.

Ledoux, M. and Talagrand, M. *Probability in Banach Spaces*. Springer, 1991.

Lounici, K., Pontil, M., Tsybakov, A.B. and van de Geer, S. Oracle inequalities and optimal inference under group sparsity *Annals of Statistics*, 39(4): 2164-2204, 2011.

Maurer, A. Concentration inequalities for functions of independent variables. *Random Structures and Algorithms*, 29:121–138, 2006.

Maurer, A. Transfer bounds for linear feature learning. *Machine Learning*, 75(3):327–350, 2009.

Maurer, A. and Pontil, M. K-dimensional coding schemes in Hilbert spaces. *IEEE Transactions on Information Theory*, 56(11):5839–5846, 2010.

McDiarmid, C. *Probabilistic Methods of Algorithmic Discrete Mathematics*. Springer, 1998.

Olshausen, B.A. and Field, D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

Slepian, D. The one-sided barrier problem for gaussian noise. *Bell System Tech. J.*, 41:463–501, 1962.

Thrun, S. and Pratt, L. *Learning to Learn*. Springer, 1998.

Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58(1):267–288, 1996.