Supervised Dictionary Learning

Julien Mairal
INRIA-Willow project
julien.mairal@inria.fr

Francis Bach
INRIA-Willow project
francis.bach@inria.fr

Jean Ponce
Ecole Normale Supérieure
jean.ponce@ens.fr

Guillermo Sapiro
University of Minnesota
guille@ece.umn.edu

Andrew Zisserman
University of Oxford
az@robots.ox.ac.uk

Abstract

It is now well established that sparse signal models are well suited for restoration tasks and can be effectively learned from audio, image, and video data. Recent research has been aimed at learning *discriminative* sparse models instead of purely *reconstructive* ones. This paper proposes a new step in that direction, with a novel sparse representation for signals belonging to different classes in terms of a shared dictionary and discriminative class models. The linear version of the proposed model admits a simple probabilistic interpretation, while its most general variant admits an interpretation in terms of kernels. An optimization framework for learning all the components of the proposed model is presented, along with experimental results on standard handwritten digit and texture classification tasks.

1 Introduction

Sparse and overcomplete image models were first introduced in [1] for modeling the spatial receptive fields of simple cells in the human visual system. The linear decomposition of a signal using a few atoms of a *learned* dictionary, instead of predefined ones–such as wavelets–has recently led to state-of-the-art results for numerous low-level image processing tasks such as denoising [2], showing that sparse models are well adapted to natural images. Unlike principal component analysis decompositions, these models are in general *overcomplete*, with a number of basis elements greater than the dimension of the data. Recent research has shown that sparsity helps to capture higher-order correlation in data. In [3, 4], sparse decompositions are used with predefined dictionaries for face and signal recognition. In [5], dictionaries are learned for a reconstruction task, and the corresponding sparse models are used as features in an SVM. In [6], a *discriminative* method is introduced for various classification tasks, learning one dictionary per class; the classification process itself is based on the corresponding reconstruction error, and does not exploit the actual decomposition coefficients. In [7], a generative model for documents is learned at the same time as the parameters of a deep network structure. In [8], multi-task learning is performed by learning features and tasks are selected using a sparsity criterion. The framework we present in this paper extends these approaches by learning simultaneously a single shared dictionary as well as models for different signal classes in a mixed generative and discriminative formulation (see also [9], where a different discriminative term is added to the classical reconstructive one). Similar joint generative/discriminative frameworks have started to appear in probabilistic approaches to learning, e.g., [10, 11, 12, 13, 14], and in neural networks [15], but not, to the best of our knowledge, in the sparse dictionary learning framework. Section 2 presents a formulation for learning a dictionary tuned for a classification task, which we call supervised dictionary learning, and Section 3 its interpretation in term of probability and kernel frameworks. The optimization procedure is detailed in Section 4, and experimental results are presented in Section 5.

2 Supervised dictionary learning

We present in this section the core of the proposed model. In classical *sparse coding* tasks, one considers a signal $\boldsymbol{x}$ in $\mathbb{R}^n$ and a *fixed* dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k]$ in $\mathbb{R}^{n \times k}$ (allowing $k > n$, making

the dictionary overcomplete). In this setting, sparse coding with an ℓ_1 regularization[1] amounts to computing

$$\mathcal{R}^\star(\boldsymbol{x}, \mathbf{D}) = \min_{\boldsymbol{\alpha} \in \mathbb{R}^k} ||\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 + \lambda_1 ||\boldsymbol{\alpha}||_1. \tag{1}$$

It is well known in the statistics, optimization, and compressed sensing communities that the ℓ_1 penalty yields a sparse solution, very few non-zero coefficients in $\boldsymbol{\alpha}$, although there is no explicit analytic link between the value of λ_1 and the effective sparsity that this model yields. Other sparsity penalties using the ℓ_0 regularization[2] can be used as well. Since it uses a proper norm, the ℓ_1 formulation of sparse coding is a convex problem, which makes the optimization tractable with algorithms such as those introduced in [16, 17], and has proven in practice to be more stable than its ℓ_0 counterpart, in the sense that the resulting decompositions are less sensitive to small perturbations of the input signal $\boldsymbol{x}$. Note that sparse coding with an ℓ_0 penalty is an NP-hard problem and is often approximated using greedy algorithms.

In this paper, we consider a setting, where the signal may belong to any of p different classes. We first consider the case of $p = 2$ classes and later discuss the multiclass extension. We consider a training set of m labeled signals $(\boldsymbol{x}_i)_{i=1}^m$ in $\mathbb{R}^n$, associated with binary labels $(y_i \in \{-1, +1\})_{i=1}^m$. Our goal is to learn *jointly* a single dictionary $\mathbf{D}$ *adapted to the classification task* and a function f which should be positive for any signal in class $+1$ and negative otherwise. We consider in this paper two different models to use the sparse code $\boldsymbol{\alpha}$ for the classification task:
(i) linear in $\boldsymbol{\alpha}$: $f(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{w}^T\boldsymbol{\alpha} + b$, where $\boldsymbol{\theta} = \{\mathbf{w} \in \mathbb{R}^k, b \in \mathbb{R}\}$ parametrizes the model.
(ii) bilinear in $\boldsymbol{x}$ and $\boldsymbol{\alpha}$: $f(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \boldsymbol{x}^T\mathbf{W}\boldsymbol{\alpha} + b$, where $\boldsymbol{\theta} = \{\mathbf{W} \in \mathbb{R}^{n \times k}, b \in \mathbb{R}\}$. In this case, the model is bilinear and f acts on both $\boldsymbol{x}$ and its sparse code $\boldsymbol{\alpha}$.

The number of parameters in (ii) is greater than in (i), which allows for richer models. Note that one can interpret $\mathbf{W}$ as a linear filter encoding the input signal $\boldsymbol{x}$ into a model for the coefficients $\boldsymbol{\alpha}$, which has a role similar to the encoder in [18] but for a discriminative task.

A classical approach to obtain $\boldsymbol{\alpha}$ for (i) or (ii) is to first adapt D to the data, solving

$$\min_{\mathbf{D}, \boldsymbol{\alpha}} \sum_{i=1}^m ||\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda_1 ||\boldsymbol{\alpha}_i||_1, \tag{2}$$

Note also that since the reconstruction errors $||\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2$ are invariant to scaling simultaneously $\mathbf{D}$ by a scalar and $\boldsymbol{\alpha}_i$ by its inverse, we need to constrain the ℓ_2 norm of the columns of $\mathbf{D}$. Such a constraint is classical in sparse coding [2]. This reconstructive approach (dubbed REC in this paper) provides sparse codes $\boldsymbol{\alpha}_i$ for each signal $\boldsymbol{x}_i$, which can be used a posteriori in a regular classifier such as logistic regression, which would require to solve

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^m \mathcal{C}\big(y_i f(\boldsymbol{x}_i, \boldsymbol{\alpha}_i, \boldsymbol{\theta})\big) + \lambda_2 ||\boldsymbol{\theta}||_2^2, \tag{3}$$

where $\mathcal{C}$ is the logistic loss function ($\mathcal{C}(x) = \log(1 + e^{-x})$), which enjoys properties similar to that of the hinge loss from the SVM literature, while being differentiable, and λ_2 is a regularization parameter, which prevents overfitting. This is the approach chosen in [5] (with SVMs). However, our goal is to learn jointly $\mathbf{D}$ and the model parameters $\boldsymbol{\theta}$. To that effect, we propose the formulation

$$\min_{\mathbf{D}, \boldsymbol{\theta}, \boldsymbol{\alpha}} \Big(\sum_{i=1}^m \mathcal{C}\big(y_i f(\boldsymbol{x}_i, \boldsymbol{\alpha}_i, \boldsymbol{\theta})\big) + \lambda_0 ||\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda_1 ||\boldsymbol{\alpha}_i||_1 \Big) + \lambda_2 ||\boldsymbol{\theta}||_2^2, \tag{4}$$

where λ_0 controls the importance of the reconstruction term, and the loss for a pair $(\boldsymbol{x}_i, y_i)$ is

$$\begin{aligned} &\mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i) = \min_{\boldsymbol{\alpha}} \mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i), \\ &\text{where } \ \mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i) = \mathcal{C}\big(y_i f(\boldsymbol{x}_i, \boldsymbol{\alpha}_i, \boldsymbol{\theta})\big) + \lambda_0 ||\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda_1 ||\boldsymbol{\alpha}_i||_1. \end{aligned} \tag{5}$$

In this setting, the classification procedure of a new signal $\boldsymbol{x}$ with an unknown label y, given a learned dictionary $\mathbf{D}$ and parameters $\boldsymbol{\theta}$, involves *supervised sparse coding*:

$$\min_{y \in \{-1; +1\}} \mathcal{S}^\star(\boldsymbol{x}, \mathbf{D}, \boldsymbol{\theta}, y), \tag{6}$$

The learning procedure of Eq. (4) minimizes the sum of the costs for the pairs $(\boldsymbol{x}_i, y_i)_{i=1}^m$ and corresponds to a generative model. We will refer later to this model as SDL-G (supervised dictionary

[1] The ℓ_1 norm of a vector $\boldsymbol{x}$ of size n is defined as $||\boldsymbol{x}||_1 = \sum_{i=1}^n |\boldsymbol{x}[i]|$.

[2] The ℓ_0 pseudo-norm of a vector $\boldsymbol{x}$ is the number of nonzeros coefficients of $\boldsymbol{x}$. Note that it is not a norm.

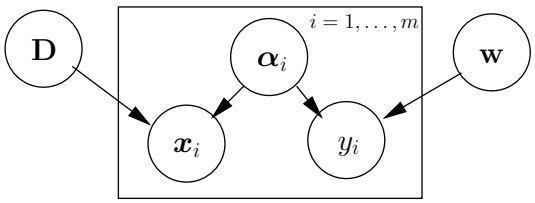


Figure 1: Graphical model for the proposed generative/discriminative learning framework.

learning, generative). Note the explicit incorporation of the reconstructive and discriminative component into sparse coding, in addition to the classical reconstructive term (see [9] for a different classification component).

However, since the classification procedure from Eq. (6) compares the different costs $\mathcal{S}^\star(\boldsymbol{x}, \mathbf{D}, \boldsymbol{\theta}, y)$ of a given signal for each class $y = -1, +1$, a more discriminative approach is to not only make the costs $\mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i)$ small, as in (4), but also make the value of $\mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i)$ greater than $\mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i)$, which is the purpose of the logistic loss function $\mathcal{C}$. This leads to:

$$\min_{\mathbf{D}, \boldsymbol{\theta}} \Big(\sum_{i=1}^{m} \mathcal{C}(\mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - \mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i)) \Big) + \lambda_2 ||\boldsymbol{\theta}||_2^2. \tag{7}$$

As detailed below, this problem is more difficult to solve than (4), and therefore we adopt instead a mixed formulation between the minimization of the generative Eq. (4) and its discriminative version (7), (see also [13])—that is,

$$\Big(\sum_{i=1}^{m} \mu \mathcal{C}(\mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - \mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i)) + (1-\mu) \mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i) \Big) + \lambda_2 ||\boldsymbol{\theta}||_2^2, \tag{8}$$

where μ controls the trade-off between the reconstruction from Eq. (4) and the discrimination from Eq. (7). This is the proposed generative/discriminative model for sparse signal representation and classification from learned dictionary $\mathbf{D}$ and model $\boldsymbol{\theta}$. We will refer to this mixed model as SDL-D, (supervised dictionary learning, discriminative). Note also that, again, we constrain the norm of the columns of $\mathbf{D}$ to be less than or equal to one.

All of these formulations admit a straightforward multiclass extension, using *softmax* discriminative cost functions $\mathcal{C}_i(x_1, ..., x_p) = \log(\sum_{j=1}^{p} e^{x_j - x_i})$, which are multiclass versions of the logistic function, and learning one model $\boldsymbol{\theta}_i$ per class. Other possible approaches such as one-vs-all or one-vs-one are of course possible, and the question of choosing the best approach among these possibilities is still open. Compared with earlier work using one dictionary per class [6], our model has the advantage of letting multiple classes share some features, and uses the coefficients $\boldsymbol{\alpha}$ of the sparse representations as part of the classification procedure, thereby following the works from [3, 4, 5], but with learned representations optimized for the classification task similar to [9, 10].

Before presenting the optimization procedure, we provide below two interpretations of the linear and bilinear versions of our formulation in terms of a probabilistic graphical model and a kernel.

3 Interpreting the model

3.1 A probabilistic interpretation of the linear model

Let us first construct a graphical model which gives a probabilistic interpretation to the training and classification criteria given above when using a linear model with zero bias (no constant term) on the coefficients—that is, $f(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{w}^T \boldsymbol{\alpha}$. It consists of the following components (Figure 1):

• The matrices $\mathbf{D}$ and the vector $\mathbf{w}$ are parameters of the problem, with a Gaussian prior on $\mathbf{w}$, $p(\mathbf{w}) \propto e^{-\lambda_2 ||\mathbf{w}||_2^2}$, and a constraint on the columns of $\mathbf{D}$–that is, $||\mathbf{d}_j||_2^2 = 1$ for all j. All the $\mathbf{d}_j$'s are considered independent of each other.
• The coefficients $\boldsymbol{\alpha}_i$ are latent variables with a Laplace prior, $p(\boldsymbol{\alpha}_i) \propto e^{-\lambda_1 ||\boldsymbol{\alpha}_i||_1}$.
• The signals $\boldsymbol{x}_i$ are generated according to a Gaussian probability distribution conditioned on $\mathbf{D}$ and $\boldsymbol{\alpha}_i$, $p(\boldsymbol{x}_i | \boldsymbol{\alpha}_i, \mathbf{D}) \propto e^{-\lambda_0 ||\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2}$. All the $\boldsymbol{x}_i$'s are considered independent from each other.

• The labels y_i are generated according to a probability distribution conditioned on $\mathbf{w}$ and $\boldsymbol{\alpha}_i$, and given by $p(y_i = \epsilon|\boldsymbol{\alpha}_i, \mathbf{W}) = e^{-\epsilon \mathbf{w}^T \boldsymbol{\alpha}_i}/\left(e^{-\mathbf{W}^T\boldsymbol{\alpha}_i} + e^{\mathbf{W}^T\boldsymbol{\alpha}_i}\right)$. Given $\mathbf{D}$ and $\mathbf{w}$, all the triplets $(\boldsymbol{\alpha}_i, \boldsymbol{x}_i, y_i)$ are independent.

What is commonly called "generative training" in the literature (e.g., [12, 13]), amounts to finding the maximum likelihood estimates for $\mathbf{D}$ and $\mathbf{w}$ according to the joint distribution $p(\{\boldsymbol{x}_i, y_i\}_{i=1}^m, \mathbf{D}, \mathbf{W})$, where the $\boldsymbol{x}_i$'s and the y_i's are the training signals and their labels respectively. It can easily be shown (details omitted due to space limitations) that there is an equivalence between this generative training and our formulation in Eq. (4) under MAP approximations.[3] Although joint generative modeling of $\boldsymbol{x}$ and y through a shared representation has shown great promise [10], we show in this paper that a more discriminative approach is desirable. "Discriminative training" is slightly different and amounts to maximizing $p(\{y_i\}_{i=1}^m, \mathbf{D}, \mathbf{w}|\{\boldsymbol{x}_i\}_{i=1}^m)$ with respect to $\mathbf{D}$ and $\mathbf{w}$: Given some input data, one finds the best parameters that will predict the labels of the data. The same kind of MAP approximation relates this discriminative training formulation to the discriminative model of Eq. (7) (again, details omitted due to space limitations). The mixed approach from Eq. (8) is a classical trade-off between generative and discriminative (e.g., [12, 13]), where generative components are often added to discriminative frameworks to add robustness, e.g., to noise and occlusions (see examples of this for the model in [9]).

3.2 A kernel interpretation of the bilinear model

Our bilinear model with $f(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \boldsymbol{x}^T \mathbf{W} \boldsymbol{\alpha} + b$ does not admit a straightforward probabilistic interpretation. On the other hand, it can easily be interpreted in terms of kernels: Given two signals $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, with coefficients $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$, using the kernel $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_2 \boldsymbol{x}_1^T \boldsymbol{x}_2$ in a logistic regression classifier amounts to finding a decision function of the same form as f. It is a product of two linear kernels, one on the $\boldsymbol{\alpha}$'s and one on the input signals $\boldsymbol{x}$. Interestingly, Raina et al. [5] learn a dictionary adapted to reconstruction on a training set, then train an SVM a posteriori on the decomposition coefficients $\boldsymbol{\alpha}$. They derive and use a Fisher kernel, which can be written as $K'(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{\alpha}_1^T \boldsymbol{\alpha}_2 \mathbf{r}_1^T \mathbf{r}_2$ in this setting, where the $\mathbf{r}$'s are the residuals of the decompositions. In simple experiments, which are not reported in this paper, we have observed that the kernel K, where the signals $\boldsymbol{x}$ replace the residuals $\mathbf{r}$, generally yields a level of performance similar to K' and often actually does better when the number of training samples is small or the data are noisy.

4 Optimization procedure

Classical dictionary learning techniques (e.g., [1, 5, 19]), address the problem of learning a reconstructive dictionary $\mathbf{D}$ in $\mathbb{R}^{n \times k}$ well adapted to a training set, which is presented in Eq. (3). It can be seen as an optimization problem with respect to the dictionary $\mathbf{D}$ and the coefficients $\boldsymbol{\alpha}$. Altough not jointly convex in $(\mathbf{D}, \boldsymbol{\alpha})$, it is convex with respect to each unknown when the other one is fixed. This is why block coordinate descent on $\mathbf{D}$ and $\boldsymbol{\alpha}$ performs reasonably well [1, 5, 19], although not necessarily providing the global optimum. Training when $\mu = 0$ (generative case), i.e., from Eq. (4), enjoys similar properties and can be addressed with the same optimization procedure. Equation (4) can be rewritten as:

$$\min_{\mathbf{D}, \boldsymbol{\theta}, \boldsymbol{\alpha}} \Big(\sum_{i=1}^{m} \mathcal{S}(\boldsymbol{x}_j, \boldsymbol{\alpha}_j, \mathbf{D}, \boldsymbol{\theta}, y_i) \Big) + \lambda_2 ||\boldsymbol{\theta}||_2^2, \;\; \text{s.t. } \forall\, j = 1, \ldots, k, \;\; ||\mathbf{d}_j||_2 \leq 1. \tag{9}$$

Block coordinate descent consists therefore of iterating between *supervised sparse coding*, where $\mathbf{D}$ and $\boldsymbol{\theta}$ are fixed and one optimizes with respect to the $\boldsymbol{\alpha}$'s and *supervised dictionary update*, where the coefficients $\boldsymbol{\alpha}_i$'s are fixed, but $\mathbf{D}$ and $\boldsymbol{\theta}$ are updated. Details on how to solve these two problems are given in sections 4.1 and 4.2. The discriminative version SDL-D from Eq. (7) is more problematic. To reach a local minimum for this difficult non-convex optimization problem, we have chosen a continuation method, starting from the generative case and ending with the discriminative one as in [6]. The algorithm is presented in Figure 2, and details on the hyperparameters' settings are given in Section 5.

4.1 Supervised sparse coding

The supervised sparse coding problem from Eq. (6) ($\mathbf{D}$ and $\boldsymbol{\theta}$ are fixed in this step) amounts to minimizing a convex function under an ℓ_1 penalty. The *fixed-point continuation method* (FPC) from

[3] We are also investigating how to properly estimate $\mathbf{D}$ by marginalizing over $\boldsymbol{\alpha}$ instead of maximizing with respect to $\boldsymbol{\alpha}$.

Input: n (signal dimensions); $(\boldsymbol{x}_i, y_i)_{i=1}^m$ (training signals); k (size of the dictionary); $\lambda_0, \lambda_1, \lambda_2$ (parameters); $0 \leq \mu_1 \leq \mu_2 \leq \ldots \leq \mu_m \leq 1$ (increasing sequence).
Output: $\mathbf{D} \in \mathbb{R}^{n \times k}$ (dictionary); $\boldsymbol{\theta}$ (parameters).
Initialization: Set $\mathbf{D}$ to a random Gaussian matrix with normalized columns. Set $\boldsymbol{\theta}$ to zero.
Loop: For $\mu = \mu_1, \ldots, \mu_m$,
Loop: Repeat until convergence (or a fixed number of iterations),

• *Supervised sparse coding:* Solve, for all $i = 1, \ldots, m$,

$$\left\{ \begin{array}{l} \boldsymbol{\alpha}^\star_{i,-} = \arg\min_{\boldsymbol{\alpha}} \mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -1) \\ \boldsymbol{\alpha}^\star_{i,+} = \arg\min_{\boldsymbol{\alpha}} \mathcal{S}(\boldsymbol{\alpha}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, +1) \end{array} \right. . \tag{10}$$

• *Dictionary and parameters update:* Solve

$$\min_{\mathbf{D}, \boldsymbol{\theta}} \Big(\sum_{i=1}^m \mu \mathcal{C}\big((\mathcal{S}(\boldsymbol{\alpha}^\star_{i,-}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - \mathcal{S}(\boldsymbol{\alpha}^\star_{i,+}, \boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta}, y_i))\big) + (1-\mu)\mathcal{S}(\boldsymbol{\alpha}^\star_{i,y_i}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i) + \lambda_2 ||\boldsymbol{\theta}||_2^2 \Big) \text{ s.t. } \forall j, ||\mathbf{d}_j||_2 \leq 1. \tag{11}$$

Figure 2: *SDL:* Supervised dictionary learning algorithm.

[17] achieves good results in terms of convergence speed for this class of problems. For our specific problem, denoting by g the convex function to minimize, this method only requires ∇g and a bound on the spectral norm of its Hessian $\mathcal{H}_g$. Since the we have chosen models g which are both linear in $\boldsymbol{\alpha}$, there exists, for each supervised sparse coding problem, a vector $\mathbf{a}$ in $\mathbb{R}^k$ and a scalar c in $\mathbb{R}$ such that

$$\left\{ \begin{array}{ll} g(\boldsymbol{\alpha}) = & \mathcal{C}(\mathbf{a}^T\boldsymbol{\alpha} + c) + \lambda_0 ||\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2, \\ \nabla g(\boldsymbol{\alpha}) = & \nabla \mathcal{C}(\mathbf{a}^T\boldsymbol{\alpha} + c)\mathbf{a} - 2\lambda_0 \mathbf{D}^T(\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}), \end{array} \right.$$

and it can be shown that, if $||\mathbf{U}||_2$ denotes the spectral norm of a matrix $\mathbf{U}$ (which is the magnitude of its largest eigenvalue), then we can obtain the following bound, $||\mathcal{H}_g(\boldsymbol{\alpha})||_2 \leq |\mathcal{H}_\mathcal{C}(\mathbf{a}^T\boldsymbol{\alpha}+c)|||\mathbf{a}||_2^2 + 2\lambda_0 ||\mathbf{D}^T\mathbf{D}||_2$.

4.2 Dictionary update

The problem of updating $\mathbf{D}$ and $\boldsymbol{\theta}$ in Eq. (11) is not convex in general (except when μ is close to 0), but a local minimum can be obtained using projected gradient descent (as in the general literature on dictionary learning, this local minimum has experimentally been found to be good enough in terms of classification performance).). Denoting $E(\mathbf{D}, \boldsymbol{\theta})$ the function we want to minimize in Eq. (11), we just need the partial derivatives of E with respect to $\mathbf{D}$ and the parameters $\boldsymbol{\theta}$. When considering the linear model for the $\boldsymbol{\alpha}$'s, $f(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{w}^T\boldsymbol{\alpha} + b$, and $\boldsymbol{\theta} = \{\mathbf{w} \in \mathbb{R}^k, b \in \mathbb{R}\}$, we obtain

$$\left\{ \begin{array}{ll} \dfrac{\partial E}{\partial \mathbf{D}} = & -2\lambda_0 \big(\displaystyle\sum_{i=1}^m \sum_{z=\{-1,+1\}} \omega_{i,z}(\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}^\star_{i,z})\boldsymbol{\alpha}^{\star T}_{i,z} \big), \\ \dfrac{\partial E}{\partial \mathbf{w}} = & \displaystyle\sum_{i=1}^m \sum_{z=\{-1,+1\}} \omega_{i,z} z \nabla \mathcal{C}(\mathbf{w}^T\boldsymbol{\alpha}^\star_{i,z} + b)\boldsymbol{\alpha}^\star_{i,z}, \\ \dfrac{\partial E}{\partial b} = & \displaystyle\sum_{i=1}^m \sum_{z=\{-1,+1\}} \omega_{i,z} z \nabla \mathcal{C}(\mathbf{w}^T\boldsymbol{\alpha}^\star_{i,z} + b), \end{array} \right. \tag{12}$$

where $\omega_{i,z} = -\mu z \nabla \mathcal{C}\big(\mathcal{S}(\boldsymbol{\alpha}^\star_{i,-}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - \mathcal{S}(\boldsymbol{\alpha}^\star_{i,+}, \boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i)\big) + (1-\mu)\mathbf{1}_{z=y_i}$.
Partial derivatives when using our model with multiple classes or with the bilinear models $f(\boldsymbol{x}, \boldsymbol{\alpha}, \theta) = \boldsymbol{x}^T\mathbf{W}\boldsymbol{\alpha} + b$ are not presented in this paper due to space limitations.

5 Experimental validation

We compare in this section the reconstructive approach, dubbed REC, which consists of learning a reconstructive dictionary $\mathbf{D}$ as in [5] and then learning the parameters $\boldsymbol{\theta}$ a posteriori; SDL with generative training (dubbed SDL-G); and SDL with discriminative learning (dubbed SDL-D). We also compare the performance of the linear (L) and bilinear (BL) models.

	REC L	SDL-G L	SDL-D L	REC BL	k-NN, ℓ_2	SVM-Gauss
MNIST	4.33	3.56	**1.05**	3.41	5.0	1.4
USPS	6.83	6.67	**3.54**	4.38	5.2	4.2

Table 1: Error rates on the MNIST and USPS datasets in percents for the REC, SDL-G L and SDL-D L approaches, compared with k-nearest neighbor and SVM with a Gaussian kernel [20].

Before presenting experimental results, let us briefly discuss the choice of the five model parameters λ_0, λ_1, λ_2, μ and k (size of the dictionary). Tuning all of them using cross-validation is cumbersome and unnecessary since some simple choices can be made, some of which can be made sequentially. We define first the *sparsity* parameter $\kappa = \frac{\lambda_1}{\lambda_0}$, which dictates how sparse the decompositions are. When the input data points have unit ℓ_2 norm, choosing $\kappa = 0.15$ was empirically found to be a good choice. For reconstructive tasks, a typical value often used in the literature (e.g., [19]) is $k = 256$ for $m = 100\,000$ signals. Nevertheless, for discriminative tasks, increasing the number of parameters is likely to lead to overfitting, and smaller values like $k = 64$ or $k = 32$ are preferred. The scalar λ_2 is a regularization parameter for preventing the model to overfit the input data. As in logistic regression or support vector machines, this parameter is crucial when the number of training samples is small. Performing cross validation with the fast method REC quickly provides a reasonable value for this parameter, which can be used afterward for SDL-G or SDL-D.

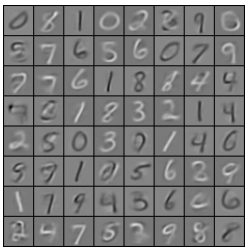
Once κ, k and λ_2 are chosen, let us see how to find λ_0, which plays the important role of controlling the trade-off between reconstruction and discrimination. First, we perform cross-validation for a few iterations with $\mu = 0$ to find a good value for SDL-G. Then, a scale factor making the costs $\mathcal{S}^\star$ discriminative for $\mu > 0$ can be chosen during the optimization process: Given a set of computed costs $\mathcal{S}^\star$, one can compute a scale factor $\gamma^\star$ such that $\gamma^\star = \arg\min_\gamma \sum_{i=1}^m \mathcal{C}(\{\gamma(\mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - \mathcal{S}^\star(\boldsymbol{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i))$. We therefore propose the following strategy, which has proven to be effective in our experiments: Starting from small values for λ_0 and a fixed κ, we apply the algorithm in Figure 2, and after a supervised sparse coding step, we compute the best scale factor $\gamma^\star$, and replace λ_0 and λ_1 by $\gamma^\star\lambda_0$ and $\gamma\lambda_1$. Typically, applying this procedure during the first 10 iterations has proven to lead to reasonable values for these parameters. Since we are following a continuation path from $\mu = 0$ to $\mu = 1$, the optimal value of μ is found along the path by measuring the classification performance of the model on a validation set during the optimization.

5.1 Digits recognition

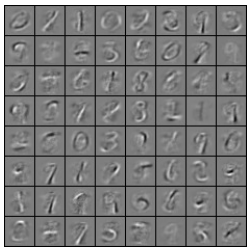
In this section, we present experiments on the popular MNIST [20] and USPS handwritten digit datasets. MNIST is composed of $70\,000$ 28×28 images, $60\,000$ for training, $10\,000$ for testing, each of them containing one handwritten digit. USPS is composed of 7291 training images and 2007 test images of size 16×16. As is often done in classification, we have chosen to learn pairwise binary classifiers, one for each pair of digits. Although our framework extends to a multiclass formulation, pairwise binary classifiers have resulted in slightly better performance in practice. Five-fold cross validation is performed to find the best pair (k, κ). The tested values for k are $\{24, 32, 48, 64, 96\}$, and for κ, $\{0.13, 0.14, 0.15, 0.16, 0.17\}$. We keep the three best pairs of parameters and use them to train three sets of pairwise classifiers. For a given image $\boldsymbol{x}$, the test procedure consists of selecting the class which receives the most votes from the pairwise classifiers. All the other parameters are obtained using the procedure explained above. Classification results are presented on Table 1 using the linear model. We see that for the linear model L, SDL-D L performs the best. REC BL offers a larger feature space and performs better than REC L, but we have observed no gain by using SDL-G BL or SDL-D BL instead of REC BL (this results are not reported in this table). Since the linear model is already performing very well, one side effect of using BL instead of L is to increase the number of free parameters and thus to cause overfitting. Note that our method is competitive since the best error rates published on these datasets (without any modification of the training set) are 0.60% [18] for MNIST and 2.4% [21] for USPS, using methods tailored to these tasks, whereas ours is generic and has not been tuned for the handwritten digit classification domain.

The purpose of our second experiment is not to measure the raw performance of our algorithm, but to answer the question *"are the obtained dictionaries* $\mathbf{D}$ *discriminative per se?"*. To do so, we have trained on the USPS dataset 10 binary classifiers, one per digit in a one vs all fashion on the training set. For a given value of μ, we obtain 10 dictionaries $\mathbf{D}$ and 10 sets of parameters $\boldsymbol{\theta}$, learned by the SDL-D L model.

To evaluate the discriminative power of the dictionaries $\mathbf{D}$, we discard the learned parameters $\boldsymbol{\theta}$ and use the dictionaries as if they had been learned in a reconstructive REC model: For each dictionary,



(a) REC, MNIST



(b) SDL-D, MNIST

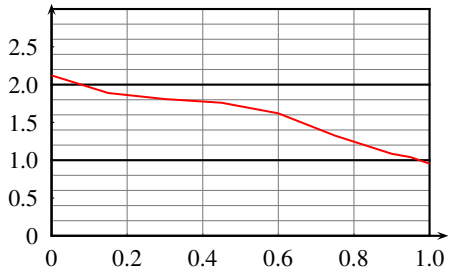


Figure 3: On the left, a reconstructive and a discriminative dictionary. On the right, average error rate in percents obtained by our dictionaries learned in a discriminative framework (SDL-D L) for various values of μ, when used at test time in a reconstructive framework (REC-L).

m	REC L	SDL-G L	SDL-D L	REC BL	SDL-G BL	SDL-D BL	Gain
300	48.84	47.34	44.84	26.34	26.34	26.34	0%
1 500	46.8	46.3	42	22.7	22.3	22.3	2%
3 000	45.17	45.1	40.6	21.99	21.22	21.22	4%
6 000	45.71	43.68	39.77	19.77	18.75	18.61	6%
15 000	47.54	46.15	38.99	18.2	17.26	15.48	15%
30 000	47.28	45.1	38.3	18.99	16.84	14.26	25%

Table 2: Error rates for the texture classification task using various methods and sizes m of the training set. The last column indicates the gain between the error rate of REC BL and SDL-D BL.

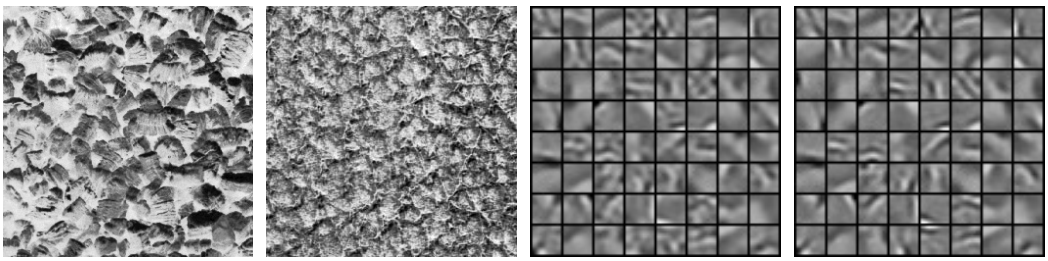
we decompose each image from the training set by solving the simple sparse reconstruction problem from Eq. (1) instead of using supervised sparse coding. This provides us with some coefficients $\boldsymbol{\alpha}$, which we use as features in a linear SVM. Repeating the sparse decomposition procedure on the test set permits us to evaluate the performance of these learned linear SVMs. We plot the average error rate of these classifiers on Figure 3 for each value of μ. We see that using the dictionaries obtained with discriminative learning ($\mu > 0$, SDL-D L) dramatically improves the performance of the basic linear classifier learned a posteriori on the $\boldsymbol{\alpha}$'s, showing that our learned dictionaries are discriminative per se. Figure 3 also shows a dictionary adapted to the reconstruction of the MNIST dataset and a discriminative one, adapted to "9 vs all".

5.2 Texture classification

In the digit recognition task, our bilinear framework did not perform better than the linear one L. We believe that one of the main reasons is due to the simplicity of the task, where a linear model is rich enough. The purpose of our next experiment is to answer the question *"When is BL worth using?"*. We have chosen to consider two texture images from the Brodatz dataset, presented in Figure 4, and to build two classes, composed of 12×12 patches taken from these two textures. We have compared the classification performance of all our methods, including BL, for a dictionary of size $k = 64$ and $\kappa = 0.15$. The training set was composed of patches from the left half of each texture and the test sets of patches from the right half, so that there is no overlap between them in the training and test set. Error rates are reported in Table 2 for varying sizes of the training set. This experiment shows that in some cases, the linear model performs very poorly where BL does better. Discrimination helps especially when the size of the training set is large. Note that we did not perform any cross-validation to optimize the parameters k and κ for this experiment. Dictionaries obtained with REC and SDL-D BL are presented in Figure 4. Note that though they are visually quite similar, they lead to very different performances.

6 Conclusion

we have introduced in this paper a discriminative approach to supervised dictionary learning that effectively exploits the corresponding sparse signal decompositions in image classification tasks, and have proposed an effective method for learning a shared dictionary and multiple (linear or bilinear) models. Future work will be devoted to adapting the proposed framework to shift-invariant models that are standard in image processing tasks, but not readily generalized to the sparse dictionary learning setting. We are also investigating extensions to unsupervised and semi-supervised learning and applications to natural image classification.



(a) Texture 1 (b) Texture 2 (c) REC (d) SDL-D BL

Figure 4: Left: test textures. Right: reconstructive and discriminative dictionaries

Acknowledgments

This paper was supported in part by ANR under grant MGA. Guillermo Sapiro would like to thank Fernando Rodriguez for insights into the learning of discriminatory sparsity patterns. His work is partially supported by NSF, NGA, ONR, ARO, and DARPA.

References

[1] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37, 1997.

[2] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. IP*, 54(12), 2006.

[3] K. Huang and S. Aviyente. Sparse representation for signal classification. In *NIPS*, 2006.

[4] J. Wright, A. Y. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. In *PAMI*, 2008. to appear.

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

[6] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Learning discriminative dictionaries for local image analysis. In *CVPR*, 2008.

[7] M. Ranzato and M. Szummer. Semi-supervised learning of compact document representations with deep networks. In *ICML*, 2008.

[8] A. Argyriou and T. Evgeniou and M. Pontil Multi-Task Feature Learning. In *NIPS*, 2006.

[9] F. Rodriguez and G. Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries. IMA Preprint 2213, 2007.

[10] D. Blei and J. McAuliffe. Supervised topic models. In *NIPS*, 2007.

[11] A. Holub and P. Perona. A discriminative framework for modeling object classes. In *CVPR*, 2005.

[12] J.A. Lasserre, C.M. Bishop, and T.P. Minka. Principled hybrids of generative and discriminative models. In *CVPR*, 2006.

[13] R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. Classification with hybrid generative/discriminative models. In *NIPS*, 2004.

[14] R. R. Salakhutdinov and G. E. Hinton. Learning a non-linear embedding by preserving class neighbourhood structure. In *AI and Statistics*, 2007.

[15] H. Larochelle, and Y. Bengio. Classification using discriminative restricted boltzmann machines. in *ICML*, 2008.

[16] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Stat.*, 32(2), 2004.

[17] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. CAAM Tech Report TR07-07, 2007.

[18] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In *NIPS*, 2006.

[19] M. Aharon, M. Elad, and A. M. Bruckstein. The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations. *IEEE Trans. SP*, 54(11), 2006.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11), 1998.

[21] B. Haasdonk and D. Keysers. Tangent distant kernels for support vector machines. In *ICPR*, 2002.