Supervised Dictionary Learning

Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, Andrew Zisserman

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Supervised Dictionary Learning

Julien Mairal — Francis Bach — Jean Ponce — Guillermo Sapiro — Andrew Zisserman

N° 6652

September 2008

Thème COG

ISSN 0249-6399 ISRN INRIA/RR--6652--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
PARIS - ROCQUENCOURT

Supervised Dictionary Learning

Julien Mairal*†, Francis Bach*† , Jean Ponce‡† , Guillermo Sapiro§ , Andrew Zisserman¶†

Thème COG — Systèmes cognitifs
Équipes-Projets Willow

Rapport de recherche n° 6652 — September 2008 — 15 pages

Abstract: It is now well established that sparse signal models are well suited to restoration tasks and can effectively be 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 multiple class-decision functions. The linear variant 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.

Key-words: sparsity, classification

* INRIA
† WILLOW project-team, Laboratoire d'Informatique de l'Ecole Normale Supérieure, ENS/INRIA/CNRS UMR 8548
‡ Ecole Normale Supérieure
§ University of Minnesota, Department of Electrical Engineering
¶ University of Oxford

Centre de recherche INRIA Paris – Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex
Téléphone : +33 1 39 63 55 11 — Télécopie : +33 1 39 63 53 30

Apprentissage de dictionnaires supervisé

Résumé : Il est maintenant bien établi que les représentations parcimonieuses de signaux sont bien adaptées à des taches de restauration d'image, de sons ou de video. De recherches récentes ont eu pour but d'apprendre des représentations discriminantes au lieu de seulement reconstructives. Ce travail propose un nouveau cadre pour représenter des signaux appartenant à plusieurs classes différentes, en apprenant de façon simultanée un dictionnaire partagé et de multiples fonctions de décision. On montre que la variante linéaire de ce cadre admet une interprétation probabilistique simple, tandis que la version plus générale peut s'interpréter en terme de noyaux. Nous proposons une méthode d'optimisation efficace et nous évaluons le modèle sur un problème de reconnaissance de chiffres manuscrits et de classification de textures.

Mots-clés : parcimonie, sparsité, classification

1 Introduction

Sparse and overcomplete image models were first introduced in [13] 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 [5], showing that sparse models are well adapted to natural images. Unlike principal component analysis decompositions, these models are most ofen 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 [9, 21], sparse decompositions are used with predefined dictionaries for face and signal recognition. In [14], dictionaries are learned for a reconstruction task, and the sparse decompositions are then used a posteriori within a classifier. In [12], 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 [17], a generative model for document representation is learned at the same time as the parameters of a deep network structure. The framework we present in this paper extends these approaches by learning simultaneously a single shared dictionary as well as multiple decision functions for different signal classes in a mixed generative and discriminative formulation (see also [18], where a different discrimination term is added to the classical reconstructive one for supervised dictionary learning via class supervised simultaneous orthogonal matching pursuit).. Similar joint generative/discriminative frameworks have started to appear in probabilistic approaches to learning, e.g., [2, 8, 10, 15, 19, 20], but not, to the best of our knowledge, in the sparse dictionary learning framework. Section 2 presents the formulation 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. We start by describing how to perform sparse coding in a supervised fashion, then show how to simultaneously learn a discriminative/reconstructive dictionary and a classifier.

2.1 Supervised Sparse Coding

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

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

[1]The ℓ_p regularization term of a vector $\boldsymbol{x}$ for $p \geq 0$ is defined as $||\boldsymbol{x}||_p^p = (\sum_{i=1}^n |\boldsymbol{x}[i]|^p)$. $||.||_p$ is a norm when $p \geq 1$. When $p = 0$, it counts the number of non-zeros elements in the vector.

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 $\boldsymbol{\alpha}$, [3], 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 (or more generally ℓ_p) regularization 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 [4, 7], and has proven in our proposed framework 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 different setting, where the signal may belong to any of p different classes. We model the signal $\boldsymbol{x}$ using a single shared dictionary $\mathbf{D}$ and a set of p decision functions $g_i(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta})$ $(i = 1, \ldots, p)$ acting on $\boldsymbol{x}$ and its sparse code $\boldsymbol{\alpha}$ over $\mathbf{D}$. The function g_i should be positive for any signal in class i and negative otherwise. The vector $\boldsymbol{\theta}$ parametrizes the model and will be *jointly* learned with $\mathbf{D}$. In the following, we will consider two kinds of decision functions:
(i) linear in $\boldsymbol{\alpha}$: $g_i(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{w}_i^T \boldsymbol{\alpha} + b_i$, where $\boldsymbol{\theta} = \{\mathbf{w}_i \in \mathbb{R}^k, b_i \in \mathbb{R}\}_{i=1}^p$, and the vectors $\mathbf{w}_i$ $(i = 1, \ldots, p)$ can be thought of as p linear models for the coefficients $\boldsymbol{\alpha}$, with the scalars b_i acting as biases;
(ii) bilinear in $\boldsymbol{x}$ and $\boldsymbol{\alpha}$: $g_i(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \boldsymbol{x}^T \mathbf{W}_i \boldsymbol{\alpha} + b_i$, where $\boldsymbol{\theta} = \{\mathbf{W}_i \in \mathbb{R}^{n \times k}, b_i \in \mathbb{R}\}_{i=1}^p$. Note that the number of parameters in (ii) is greater than in (i), which allows for richer models. One can interpret $\mathbf{W}_i$ as a filter encoding the input signal $\boldsymbol{x}$ into a model for the coefficients $\boldsymbol{\alpha}$, which has a role similar to the encoder in [16] but for a discriminative task.

Let us define *softmax* discriminative cost functions as

$$\mathcal{C}_i(x_1, ..., x_p) = \log(\sum_{j=1}^{p} e^{x_j - x_i})$$

for $i = 1, \ldots, p$. These are multiclass versions of the logistic function, enjoying properties similar to that of the hinge loss from the SVM literature, while being differentiable. Given some input signal $\boldsymbol{x}$ and fixed (for now) dictionary $\mathbf{D}$ and parameters $\boldsymbol{\theta}$, the *supervised sparse coding* problem for the class p can be defined as computing

$$\mathcal{S}_i^\star(\boldsymbol{x}, \mathbf{D}, \boldsymbol{\theta}) = \min_{\boldsymbol{\alpha}} \mathcal{S}_i(\boldsymbol{\alpha}, \boldsymbol{x}, \mathbf{D}, \boldsymbol{\theta}), \tag{2}$$

where

$$\mathcal{S}_i(\boldsymbol{\alpha}, \boldsymbol{x}, \mathbf{D}, \boldsymbol{\theta}) = \mathcal{C}_i(\{g_j(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta})\}_{j=1}^p) + \lambda_0 ||\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2 + \lambda_1 ||\boldsymbol{\alpha}||_1. \tag{3}$$

Note the explicit incorporation of the classification and discriminative component into sparse coding, in addition to the classical reconstructive term (see [18] for a different classificaiton component). In turn, any solution to this problem provides a straightforward classification procedure, namely:

$$i^\star(\boldsymbol{x}, \mathbf{D}, \boldsymbol{\theta}) = \arg\min_{i=1,\ldots,p} \mathcal{S}_i^\star(\boldsymbol{x}, \mathbf{D}, \boldsymbol{\theta}). \tag{4}$$

Compared with earlier work using one dictionary per class [12], this 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 [9, 14, 21], but with learned representations optimized for the classification task similar to [2, 18]. As shown in Section 3, this formulation has a straightforward probabilistic interpretation, but let us first see how to learn the dictionary $\mathbf{D}$ and the parameters $\boldsymbol{\theta}$ from training data.

2.2 SDL: Supervised Dictionary Learning

Let us assume that we are given p sets of training data T_i, $i = 1, \ldots, p$, such that all samples in T_i belong to class i. The most direct method for learning $\mathbf{D}$ and $\boldsymbol{\theta}$ is to minimize with respect to these variables the mean value of $\mathcal{S}_i^\star$, with an ℓ_2 regularization term to prevent overfitting:

$$\min_{\mathbf{D},\boldsymbol{\theta}} \Big(\sum_{i=1}^{p} \sum_{j \in T_i} \mathcal{S}_i^\star(\boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta}) \Big) + \lambda_2 ||\boldsymbol{\theta}||_2^2, \quad \text{s.t.} \quad \forall\, i = 1, \ldots, k, \quad ||\mathbf{d}_i||_2 \leq 1. \tag{5}$$

Since the reconstruction errors $||\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2$ are invariant to scaling simultaneously $\mathbf{D}$ by a scalar and $\boldsymbol{\alpha}$ by its inverse, constraining the ℓ_2 norm of columns of $\mathbf{D}$ prevents any transfer of energy between these two variables, which would have the effect of overcoming the sparsity penalty. Such a constraint is classical in sparse coding [5]. We will refer later to this model as SDL-G (supervised dictionary learning, generative).

Nevertheless, since the classification procedure from Eq. (4) will compare the different residuals $\mathcal{S}_i^\star$ of a given signal for $i = 1, \ldots, p$, a more discriminative approach is to not only make the $\mathcal{S}_i^\star$ small for signals with label i, as in (5), but also make the value of $\mathcal{S}_j^\star$ greater than $\mathcal{S}_i^\star$ for j different than i, which is the purpose of the softmax function $\mathcal{C}_i$. This leads to:

$$\min_{\mathbf{D},\boldsymbol{\theta}} \Big(\sum_{i=1}^{p} \sum_{j \in T_i} \mathcal{C}_i(\{\mathcal{S}_l^\star(\boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta})\}_{l=1}^p) \Big) + \lambda_2 ||\boldsymbol{\theta}||_2^2 \quad \text{s.t.} \quad \forall\, i = 1, \ldots, k, \quad ||\mathbf{d}_i||_2 \leq 1. \tag{6}$$

As detailed below, this problem is more difficult to solve than Eq. (5), and therefore we adopt instead a mixed formulation between the minimization of the generative Eq. (5) and its discriminative version (6), [15]—that is,

$$\begin{aligned} \min_{\mathbf{D},\boldsymbol{\theta}} \Big(\sum_{i=1}^{p} \sum_{j \in T_i} \mu \mathcal{C}_i(\{\mathcal{S}_l^\star(\boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta})\}_{l=1}^p) + (1-\mu) S_i^\star(\boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta}) \Big) + \lambda_2 ||\boldsymbol{\theta}||_2^2 \\ \text{s.t.} \quad \forall i, \quad ||\mathbf{d}_i||_2 \leq 1, \end{aligned} \tag{7}$$

where μ controls the trade-off between reconstruction from Eq. (5) and discrimination from Eq. (6). 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).

Before presenting the proposed 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.

.

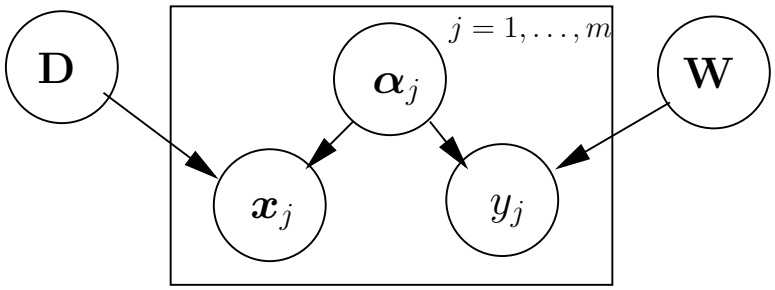


Figure 1: Graphical model for the proposed generative/discriminative learning framework.

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, $g_i(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{w}_i^T \boldsymbol{\alpha}$. This model consists of the following components (Figure 1):

• The matrices $\mathbf{D}$ and $\mathbf{W}$ are parameters of the problem, with a Gaussian prior on $\mathbf{W}$, $p(\mathbf{W}) \propto e^{-\lambda_2 ||\mathbf{W}||_2^2}$, and on the columns of $\mathbf{D}$, $p(\mathbf{D}) \propto \prod_{l=1}^{k} e^{-\gamma_l ||\mathbf{d}_l||_2^2}$, where the γ_l's are the Gaussian parameters. All the $\mathbf{d}_l$'s are considered independent of each other.

• The coefficients $\boldsymbol{\alpha}_j$ are latent variables with a Laplace prior, $p(\boldsymbol{\alpha}_j) \propto e^{-\lambda_1 ||\boldsymbol{\alpha}_j||_1}$.

• The signals $\boldsymbol{x}_j$ are generated according to a Gaussian probability distribution conditioned on $\mathbf{D}$ and $\boldsymbol{\alpha}_j$, $p(\boldsymbol{x}_j | \boldsymbol{\alpha}_j, \mathbf{D}) \propto e^{-\lambda_0 ||\boldsymbol{x}_j - \mathbf{D}\boldsymbol{\alpha}_j||_2^2}$. All the $\boldsymbol{x}_j$'s are considered independent from each other.

• The labels y_j are generated according to a probability distribution conditioned on $\mathbf{W}$ and $\boldsymbol{\alpha}_j$, and given by $p(y_j = i | \boldsymbol{\alpha}_j, \mathbf{W}) = e^{-\mathbf{W}_i^T \boldsymbol{\alpha}_j} / \sum_{l=1}^{p} e^{-\mathbf{W}_l^T \boldsymbol{\alpha}_j}$. Given $\mathbf{D}$ and $\mathbf{W}$, all the triplets $(\boldsymbol{\alpha}_j, \boldsymbol{x}_j, y_j)$ are independent.

What is commonly called "generative training" in the literature (e.g., [10, 15]), amounts to finding the maximum likelihood for $\mathbf{D}$ and $\mathbf{W}$ according to the joint distribution $p(\{\boldsymbol{x}_j, y_j\}_{j=1}^m, \mathbf{D}, \mathbf{W})$, where the $\boldsymbol{x}_j$'s and the y_j's are respectively the training signals and their labels. 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. (5) under MAP approximations.[2] Although joint generative modeling of $\boldsymbol{x}$ and y through a shared representation, e.g., [2], has shown great promise, we show in this paper that a more discriminative approach is desirable. "Discriminative training" is slightly different and amounts to maximizing $p(\{y_j\}_{j=1}^m, \mathbf{D}, \mathbf{W} | \{\boldsymbol{x}_j\}_{j=1}^m)$ 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. (6)

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

(again, details omitted due to space limitations). The mixed approach from Eq. (7) is a classical trade-off between generative and discriminative (e.g., [10, 15]), 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 [18]).

3.2 A kernel interpretation of the bilinear model

Our bilinear model with $g_i(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \boldsymbol{x}^T \mathbf{W}_i \boldsymbol{\alpha} + b_i$ 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 (ii). 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. [14] 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. Experimentally, 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, 13, 14]), address the problem of learning a reconstructive dictionary $\mathbf{D}$ in $\mathbb{R}^{n \times k}$ well adapted to a training set T as

$$\min_{\mathbf{D}, \boldsymbol{\alpha}} \sum_{j \in T} ||\boldsymbol{x}_j - \mathbf{D}\boldsymbol{\alpha}_j||_2^2 + \lambda_1 ||\boldsymbol{\alpha}_j||_1, \tag{8}$$

which is not jointly convex in $(\mathbf{D}, \boldsymbol{\alpha})$, but 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, 13, 14], although not necessarily providing the global optimum. Training when $\mu = 0$ (generative case), i.e., from Eq. (5), enjoys similar properties and can be addressed with the same optimization procedure. Equation (5) can be rewritten as:

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

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 $\boldsymbol{\alpha}_j$'s are fixed, but $\mathbf{D}$ and $\boldsymbol{\theta}$ are updated. Details on how to solve these two problems are given in Section 4.1 and 4.2.

The discriminative version of SDL from Eq. (6) is more problematic. The minimization of the term $\mathcal{C}_i(\{\mathcal{S}_l(\boldsymbol{\alpha}_{jl}, \boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta})\}_{l=1}^p)$ with respect to $\mathbf{D}$ and $\boldsymbol{\theta}$ when the $\boldsymbol{\alpha}_{jl}$'s are fixed, is not convex in general, and does not necessarily decrease the first term of Eq. (6), i.e., $\mathcal{C}_i(\{\mathcal{S}_l^\star(\boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta})\}_{l=1}^p)$. To reach a local minimum for this difficult problem, we have chosen a continuation method,

Input: p (number of classes); n (signal dimensions); $\{T_i\}_{i=1}^p$ (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. 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:* Compute, for all $i = 1, \ldots, p$, all j in T_i, and all $l = 1, \ldots, p$,

$$\boldsymbol{\alpha}_{jl}^\star = \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^k} \mathcal{S}_l(\boldsymbol{\alpha}, \boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta}). \tag{10}$$

• *Dictionary update:* Solve, under the constraint $||\mathbf{d}_l|| \leq 1$ for all $l = 1, \ldots, k$

$$\min_{\mathbf{D}, \boldsymbol{\theta}} \Big(\sum_{i=1}^{p} \sum_{j \in T_i} \mu \mathcal{C}_i(\{\mathcal{S}_l(\boldsymbol{\alpha}_{jl}^\star, \boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta})\}_{l=1}^p) + (1-\mu) S_i(\boldsymbol{\alpha}_{ji}^\star, \boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta}) \Big) + \lambda_2 ||\boldsymbol{\theta}||_2^2. \tag{11}$$

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

starting from the generative case and ending with the discriminative one as in [12]. The algorithm is presented on 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. (10) ($\mathbf{D}$ 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 [7] achieves state-of-the-art results in terms of convergence speed for this class of problems. It has proven in our experiments to be simple, efficient, and well adapted to our supervised sparse coding problem. Algorithmic details are given in [7]. For our specific problem, denoting by f the convex function to minimize, this method only requires ∇f and a bound on the spectral norm of its Hessian $\mathcal{H}_f$. Since the we have chosen decision functions g_i in Eq. (10) which are linear in $\boldsymbol{\alpha}$, there exists, for each signal $\boldsymbol{x}$ to be sparsely represented, a matrix $\mathbf{A}$ in $\mathbb{R}^{k \times p}$ and a vector $\mathbf{b}$ in $\mathbb{R}^p$ such that

$$\begin{cases} f(\boldsymbol{\alpha}) = & \mathcal{C}_i(\mathbf{A}^T\boldsymbol{\alpha} + \mathbf{b}) + \lambda_0 ||\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}||_2^2, \\ \nabla f(\boldsymbol{\alpha}) = & \mathbf{A}\nabla\mathcal{C}_i(\mathbf{A}^T\boldsymbol{\alpha} + \mathbf{b}) - 2\lambda_0 \mathbf{D}^T(\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}), \end{cases}$$

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 $||\mathcal{H}_f||_2 \leq (1 - \frac{1}{p})||\mathbf{A}^T\mathbf{A}||_2^2 + 2\lambda_0||\mathbf{D}^T\mathbf{D}||_2$. In the case where $p = 2$ (only two classes), we can obtain a tighter bound, $||\mathcal{H}_f(\boldsymbol{\alpha})||_2 \leq e^{-\mathcal{C}_1(\mathbf{A}^T\boldsymbol{\alpha}) - \mathcal{C}_2(\mathbf{A}^T\boldsymbol{\alpha})} ||\mathbf{a}_2 - \mathbf{a}_1||_2^2 + 2\lambda_0||\mathbf{D}^T\mathbf{D}||_2$, where $\mathbf{a}_1$ and $\mathbf{a}_2$ are the first and second columns of $\mathbf{A}$.

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 for our formulation). 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 $\boldsymbol{\theta}$. Details when using the linear model for the $\boldsymbol{\alpha}$'s, $g_i(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \mathbf{w}_i^T \boldsymbol{\alpha} + b_i$, and $\boldsymbol{\theta} = \{\mathbf{W} \in \mathbb{R}^{k \times p}, \mathbf{b} \in \mathbb{R}^p\}$, are

$$\left\{ \begin{array}{rcl} \dfrac{\partial E}{\partial \mathbf{D}} = & -2\lambda_0 \big(\displaystyle\sum_{i=1}^{p} \sum_{j \in T_i} \sum_{l=1}^{p} \omega_{jl} (\boldsymbol{x}_j - \mathbf{D}\boldsymbol{\alpha}_{jl}^\star) \boldsymbol{\alpha}_{jl}^{\star T} \big), \\ \dfrac{\partial E}{\partial \mathbf{W}} = & \displaystyle\sum_{i=1}^{p} \sum_{j \in T_i} \sum_{l=1}^{p} \omega_{jl} \boldsymbol{\alpha}_{jl}^\star \nabla \mathcal{C}_l^T (\mathbf{W}^T \boldsymbol{\alpha}_{jl}^\star + \mathbf{b}), \\ \dfrac{\partial E}{\partial \mathbf{b}} = & \displaystyle\sum_{i=1}^{p} \sum_{j \in T_i} \sum_{l=1}^{p} \omega_{jl} \nabla \mathcal{C}_l (\mathbf{W}^T \boldsymbol{\alpha}_{jl}^\star + \mathbf{b}), \end{array} \right. \tag{12}$$

where

$$\omega_{jl} = \mu \nabla \mathcal{C}_i(\{\mathcal{S}_m(\boldsymbol{\alpha}_{jm}^\star, \boldsymbol{x}_j, \mathbf{D}, \boldsymbol{\theta})\}_{m=1}^p)[l] + (1-\mu)\mathbf{1}_{l=i}. \tag{13}$$

Partial derivatives when using our model with the bilinear decision functions $g_i(\boldsymbol{x}, \boldsymbol{\alpha}, \theta) = \boldsymbol{x}^T \mathbf{W}_i \boldsymbol{\alpha} + b_i$ are not given in this paper because of space limitations.

5 Experimental validation

We compare in this section a reconstructive approach, dubbed REC, which consists of learning a reconstructive dictionary $\mathbf{D}$ as in [14] 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) decision functions.

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 done 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. The number of parameters to learn is linear in k, the number of elements in the dictionary $\mathbf{D}$. For reconstructive tasks, $k = 256$ is a typical value often used in the literature (e.g., [1]). Nevertheless, for discriminative tasks, increasing the number of parameters is likely to allow 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. In logistic regression, a projection matrix maps input data onto a softmax function, and its shape and scale are adapted so that it becomes discriminative according to an underlying probabilistic model. In the model we are proposing, the functions $\mathcal{S}_i^\star$ are also mapped onto a softmax function, and the parameters $\mathbf{D}$ and $\boldsymbol{\theta}$ are adapted (learned) in such a way that $\mathcal{S}_i^\star$ becomes discriminative. However, for a fixed κ, the second and third terms of $\mathcal{S}_i^\star$, namely $\lambda_0||\boldsymbol{x}-\mathbf{D}\boldsymbol{\alpha}||_2^2$ and $\lambda_0\kappa||\boldsymbol{\alpha}||_1$, are not freely scalable when adapting $\mathbf{D}$ and $\boldsymbol{\theta}$, since their magnitudes are bounded. λ_0 plays the important role of controlling the trade-off between reconstruction and discrimination in Eq. (3). 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 $\mathcal{S}_i^\star$'s discriminative for $\mu > 0$ can be chosen during the optimization process: Given a set of $\mathcal{S}_i^\star$'s, one can compute a scale factor γ such that $\gamma = \arg\min_\gamma \sum_{i=1}^p \sum_{j\in T_i} \mathcal{C}_i(\{\gamma \mathcal{S}_l^\star(\boldsymbol{x}_j, \mathbf{D}, \mathbf{W})\})$. We therefore propose the following strategy, which has proven to be efficient during 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 γ, and replace λ_0 and λ_1 by $\gamma\lambda_0$ and $\gamma\lambda_1$. Typically, applying this procedure during the first 10 iterations has proven to lead to reasonable values for this parameter.

Since we are following a continuation path starting 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 [11] and USPS handwritten digit datasets. MNIST is composed of 70 000 images of 28×28 pixels, 60 000 for training, 10 000 for testing, each of them containing a handwritten digit. USPS is composed of 7291 training images and 2007 test images. As it is often done in classification, we have chosen to learn pairwise binary classifiers, one for each pair of digits. Although we have presented a multiclass framework, pairwise binary classifiers have proven to offer a slightly better performance in practice. Five-fold cross validation has been performed to find the best pair (k, κ). The tested values for k are $\{24, 32, 48, 64, 96\}$, and for κ, $\{0.13, 0.14, 0.15, 0.16, 0.17\}$. Then, we have kept the three best pairs of parameters and used them to train three sets of pairwise classifiers. For a given patch $\boldsymbol{x}$, 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 when 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. Nevertheless, we have observed no gain by using SDL-G BL or SDL-D BL instead of REC BL. 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 the best error rates published on these datasets (without any modification of the training set) are 0.60% [16] for MNIST and 2.4% [6] for USPS, using methods tailored to these tasks, whereas ours is generic and has not been tuned to the handwritten digit classification domain.

	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 MNIST and USPS datasets in percents from the REC, SDL-G L and SDL-D L approaches, compared with k-nearest neighbor and SVM with a Gaussian kernel [11].

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 or is the pair* $(\mathbf{D},\boldsymbol{\theta})$ *discriminative?"*. 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, 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 SVM. 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 discrimative 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 4 shows a dictionary adapted to the reconstruction of the MNIST dataset and a discriminative one, adapted to "9 vs all".

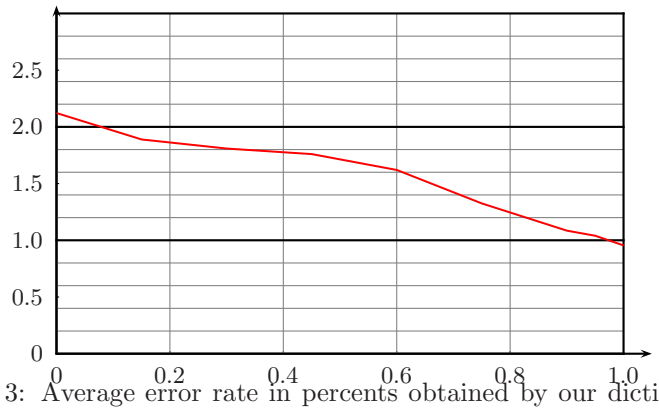
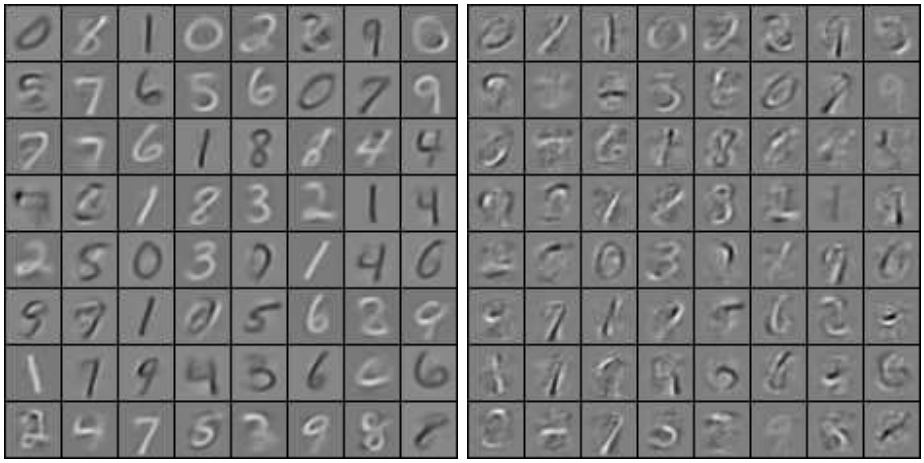


Figure 3: Average error rate in percents obtained by our dictionaries learned in a discriminative framework (SDL-D L) for various values of μ, when used in used at test time in a reconstructive framework (REC-L). See text for details.

5.2 Texture classification

In the digit recognition task, our BL bilinear framework did not perform better than L and we believe that one of the main reasons is due to the simplicity of the



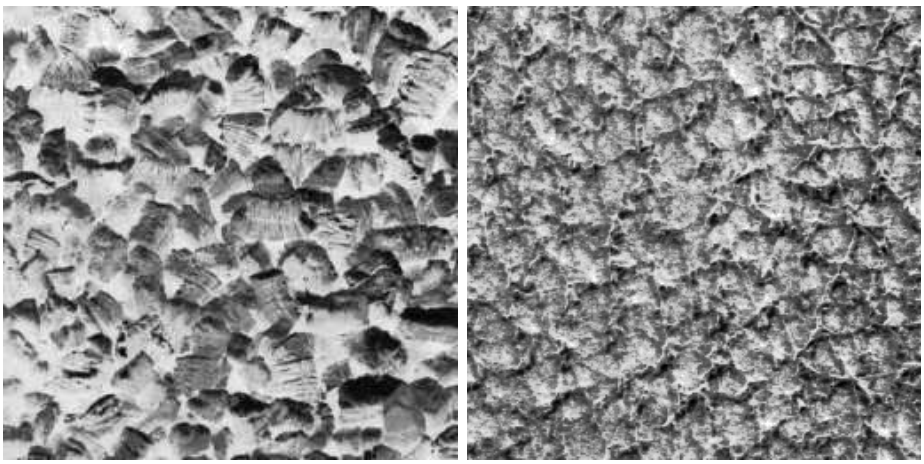
(a) REC, MNIST (b) SDL-D, MNIST

Figure 4: On the left, a reconstructive dictionary, on the right a discriminative one for the task "9 vs all".

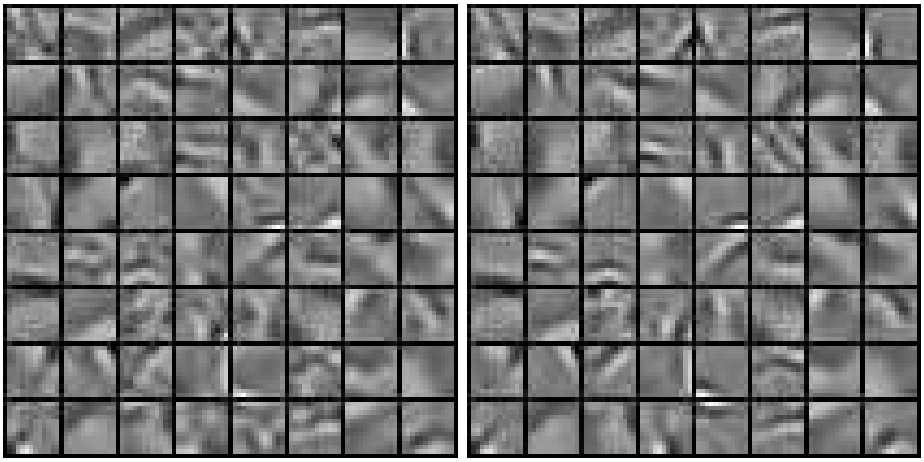
M	REC L	SDL-G L	SDL-D L	REC BL	SDL-G BL	SDL-D BL	Gain
300	48.84	47.34	44.84	26.34	26.34	26.34	0%
1500	46.8	46.3	42	22.7	22.3	22.3	2%
3000	45.17	45.1	40.6	21.99	21.22	21.22	4%
6000	45.71	43.68	39.77	19.77	18.75	18.61	6%
15000	47.54	46.15	38.99	18.2	17.26	15.48	15%
30000	47.28	45.1	38.3	18.99	16.84	14.26	25%

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

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 5, 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 for varying sizes of the training set. This experiment shows that in some cases, the linear model completely fails and BL is necessary. Discrimination helps especially when the size of the training set is particularly valuable for large training sets. 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 5. Note that though they are visually quite similar, they lead to very different performance.



(a) Texture 1 (b) Texture 2



(c) REC (d) SDL-D BL

Figure 5: Top: Test textures. Bottom left: reconstructive dictionary. Bottom right: discriminative dictionary.

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 affords an effective method for learning a shared dictionary and multiple (linear or bilinear) decision functions. 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 into natural image classification.

References

[1] 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):4311–4322, November 2006.

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

[3] D. L. Donoho. Compressive sampling. *IEEE Trans. IT*, 52(4):1289–1306, April 2006.

[4] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.

[5] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. IP*, 54(12):3736–3745, December 2006.

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

[7] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for l1-regularized minimization with applications to compressed sensing. Technical report, Rice University,, 2007. CAAM Technical Report TR07-07, http://www.caam.rice.edu/∼optimization/L1/fpc/.

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

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

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

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

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

[13] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 37:3311–3325, 1997.

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

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

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

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

[18] F. Rodriguez and G. Sapiro. Sparse representations for image classification: Learning discriminative and reconstructive non-parametric dictionaries. Technical report, University of Minnesota, December 2007. IMA Preprint 2213.

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

[20] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proc. IEEE ICCV*, 2005.

[21] J. Wright, A. Y. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. PAMI*, 2008. to appear, http://perception.csl.uiuc.edu/recognition/Home.html.

Centre de recherche INRIA Paris – Rocquencourt
Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399