	natural image 196×512	speech 500×200	stereo 288×400	video 512×200
Feature-sign	2.16 (0)	0.58 (0)	1.72 (0)	0.83 (0)
LARS	3.62 (0)	1.28 (0)	4.02 (0)	1.98 (0)
Grafting	13.39 (7e-4)	4.69 (4e-6)	11.12 (5e-4)	5.88 (2e-4)
Chen et al.'s	88.61 (8e-5)	47.49 (8e-5)	66.62 (3e-4)	47.00 (2e-4)
QP solver (CVX)	387.90 (4e-9)	1,108.71 (1e-8)	538.72 (7e-9)	1,219.80 (1e-8)

Table 1: The running time in seconds (and the relative error in parentheses) for coefficient learning algorithms applied to different natural stimulus datasets. For each dataset, the input dimension k and the number of bases n are specified as $k \times n$. The relative error for an algorithm was defined as $(f_{obj} - f^*)/f^*$, where f_{obj} is the final objective value attained by that algorithm, and f^* is the best objective value attained among all the algorithms.

where $e_i \in \mathbb{R}^n$ is the i-th unit vector. Now, we can optimize the Lagrange dual (8) using Newton's method or conjugate gradient. After maximizing $\mathcal{D}(\vec{\lambda})$, we obtain the optimal bases B as follows:

$$B^\top = (SS^\top + \Lambda)^{-1}(XS^\top)^\top. \tag{11}$$

The advantage of solving the dual is that it uses significantly fewer optimization variables than the primal. For example, optimizing $B \in \mathbb{R}^{1,000\times1,000}$ requires only 1,000 dual variables. Note that the dual formulation is independent of the sparsity function (e.g., L_1, $\mathrm{epsilonL}_1$, or other sparsity function), and can be extended to other similar models such as "topographic" cells [14].[6]

5 Experimental results

5.1 The feature-sign search algorithm

We evaluated the performance of our algorithms on four natural stimulus datasets: natural images, speech, stereo images, and natural image videos. All experiments were conducted on a Linux machine with AMD Opteron 2GHz CPU and 2GB RAM.

First, we evaluated the feature-sign search algorithm for learning coefficients with the L_1 sparsity function. We compared the running time and accuracy to previous state-of-the-art algorithms: a generic QP solver,[7] a modified version of LARS [12] with early stopping,[8] grafting [13], and Chen et al.'s interior point method [11];[9] all the algorithms were implemented in MATLAB. For each dataset, we used a test set of 100 input vectors and measured the running time[10] and the objective function at convergence. Table 1 shows both the running time and accuracy (measured by the relative error in the final objective value) of different coefficient learning algorithms. Over all datasets, feature-sign search achieved the best objective values as well as the shortest running times. Feature-sign search and modified LARS produced more accurate solutions than the other methods.[11] Feature-sign search was an order of magnitude faster than both Chen et al.'s algorithm and the generic QP solver, and it was also significantly faster than modified LARS and grafting. Moreover, feature-sign search has the crucial advantage that it can be initialized with arbitrary starting coefficients (unlike LARS); we will demonstrate that feature-sign search leads to even further speedup over LARS when applied to iterative coefficient learning.

5.2 Total time for learning bases

The Lagrange dual method for one basis learning iteration was much faster than gradient descent with iterative projections, and we omit discussion of those results due to space constraints. Below, we directly present results for the overall time taken by sparse coding for learning bases from natural stimulus datasets.

---

[6] The sparsity penalty for topographic cells can be written as $\sum_l \phi((\sum_{j\in \text{cell } l} s_j^2)^{\frac{1}{2}})$, where $\phi(\cdot)$ is a sparsity function and cell l is a topographic cell (e.g., group of 'neighboring' bases in 2-D torus representation).

[7] We used the CVX package available at `http://www.stanford.edu/~boyd/cvx/`.

[8] LARS (with LASSO modification) provides the entire regularization path with discrete L_1-norm constraints; we further modified the algorithm so that it stops upon finding the optimal solution of the Equation (4).

[9] MATLAB code is available at `http://www-stat.stanford.edu/~atomizer/`.

[10] For each dataset/algorithm combination, we report the average running time over 20 trials.

[11] A general-purpose QP package (such as CVX) does not explicitly take the sparsity of the solutions into account. Thus, its solution tends to have many very small nonzero coefficients; as a result, the objective values obtained from CVX were always slightly worse than those obtained from feature-sign search or LARS.