Fast approximations to structured sparse coding and applications to object classification

Arthur Szlam, Karol Gregor, and Yann LeCun

1 Introduction

We describe a method for fast approximation of sparse coding. The input space is subdivided by a binary decision tree, and we simultaneously learn a dictionary and assignment of allowed dictionary elements for each leaf of the tree. We store a lookup table with the assignments and the pseudoinverses for each node, allowing for very fast inference. In the process of describing this algorithm, we discuss the more general problem of learning the groups in group structured sparse modelling. We show that our method creates good sparse representations by using it in the object recognition framework of [1, 2]. Implementing our own fast version of the SIFT descriptor the whole system runs at 20 frames per second on 321×481 sized images on a laptop with a quad-core cpu, while sacrificing very little accuracy on the Caltech 101 and 15 scenes benchmarks.

1.1 Structured sparse models

Sparse modeling has lead to state of the art algorithms in image denoising, inpainting, supervised learning, and of particular interest here, object recognition. The systems described in [1, 2, 3, 4, 5] use sparse coding as an integral element. Since the coding is done densely in an image with relatively large dictionaries, this is a computationally expensive part of the recognition system, and a barrier to real time application. One standard formulation of sparse coding is to consider N d-dimensional real vectors $X = \{x_1, \dots, x_N\}$ and represent them using N K-dimensional real vectors $Z = \{z_1, \dots, z_N\}$ using a $k \times d$ dictionary matrix W by solving

$$\text{argmin}_{Z,W} \sum_k ||Wz_k - x_k||^2, \text{ s.t. } ||z_k||_0 \leq q. \tag{1}$$

Each input vector x is thus represented as a vector z with at most q nonzero coefficients. While this problem is not convex, and in fact the problem in the Z variable is NP-hard, there exist algorithms for solving both the problem in Z (e.g. Orthogonal Matching Pursuit, OMP) and the problem in both variables (e.g. K-SVD [6]) that work well in many practical situations.

It is sometimes appropriate to enforce more structure on Z than just sparsity. A simple form of structured sparsity is given by specifying a list of L allowable active sets, and some function $g : \mathbb{R}^d \mapsto \{1, ..., L\}$ associating to each x to one of the L configurations. An example of this is the output of many subspace clustering algorithms. There, X is reordered and partitioned into $PX = [X_1 \; X_2 ... \; X_L]$ (where P is a permutation matrix), so that each block X_j is near a low dimensional subspace spanned by B_j. Supposing for simplicity that each of the B_j

are of the same dimension q, then if we set $W = [B_1 ... B_L]$, the allowable active sets are given by $\{1, ..., q\}$, $\{q + 1, ..., 2q\}$, etc. By setting the allowable active sets to the blocks, and the function g to simply map each point to its nearest subspace (say in the standard sense of Euclidean projections), then we get an example of structured sparsity as described above; this sort of method is used in object recognition in [4].

In this work we will try to learn the L configurations as well as the dictionary. We introduce a LLoyd-like algorithm that alternates between updating the dictionary, updating the assignments of each data point to the groups, and updating the dictionary elements associated to a group via Simultaneous OMP [7] (SOMP).

At inference time, we need a fast method for determining which group an x belongs to. This is computationally expensive if there is a large number of groups and one needs check the projection onto each group. However, by specializing the Lloyd type algorithm to the case when each group is composed of a union of (perhaps only one) leaves of a binary decision tree, we will build a fast inference scheme into the learned dictionary. The key idea is that by using SOMP, we can learn which leaves should use which dictionary elements as we train the dictionary. To code an input, we march it down the tree until we arrive at the appropriate leaf. In addition to the decision vectors and thresholds, we will store a lookup table with the active set of each leaf as learned above, and the pseudoinverse of the columns of W corresponding to that active set. Thus after following x down the tree we need only make one matrix multiplication to get the coefficients.

Finally, we would like use these algorithms to build an accurate real time recognition system. We focus on a particular architecture studied in [1, 2, 4, 5]. We use this pipeline with two modifications. First we write our own fast implementation of the SIFT descriptor. Second we use our fast algorithm for the sparse coding step. The resulting system achieves nearly the same performance as exact sparse coding calculation on Caltech 101 and 256, and 15 scenes, but processes 321 × 481 size images at the rate of 20 frames per second on a laptop computer with a quad core cpu.

References

[1] C. Schmid S. Lazebnik and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories", in *CVPR'06*, 2006.

[2] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang, "Linear spatial pyramid matching using sparse coding for image classification", in *CVPR'09*, 2009.

[3] Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun, "Fast inference in sparse coding algorithms with applications to object recognition", Tech. Rep. CBLL-TR-2008-12-01, Computational and Biological Learning Lab, Courant Institute, NYU, 2008.

[4] K. Yu J. Yang and T. Huang., "Efcient highly overcomplete sparse coding using a mixture model.", in *European Conference on Computer Vision*, 2010.

[5] Y. Boureau, N. La Roux, F. Bach, J. Ponce, and Y. LeCun, "Ask the locals: multi-way local pooling for image recognition", in *International Conference on Computer Vision*, 2011.

[6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation", *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[7] Anna C. Gilbert, Martin J. Strauss, and Joel A. Tropp, "Simultaneous Sparse Approximation via Greedy Pursuit", *IEEE Trans. Acoust. Speech Signal Process.*, vol. 5, pp. 721–724, 2005.