Application: Sparse coding / dictionary learning

The *dictionary learning / sparse coding* problem is defined as follows: there is an unknown $n \times m$ matrix $A = (a_1 | \cdots | a_m)$ (think of $m = 10n$). We are given access to many examples of the form

$$y = Ax + e \tag{1}$$

for some distribution $\{x\}$ over sparse vectors and distribution $\{e\}$ over noise vectors with low magnitude.

Our goal is to learn the matrix A, which is called a *dictionary*.

The intuition behind this problem is that natural data elements are sparse when represented in the "right" basis, in which every coordinate corresponds to some meaningful features. For example while natural images are always dense in the pixel basis, they are sparse in other bases such as wavelet bases, where coordinates corresponds to edges etc.. and for this reason these bases are actually much better to work with for image recognition and manipulation. (And the coordinates of such bases are sometimes in a non-linear way to get even more meaningful features that eventually correspond to things such as being a picture of a cat or a picture of my grandmother etc. or at least that's the theory behind deep neural networks.) While we can simply guess some basis such as the Fourier or Wavelet to work with, it is best to learn the right basis directly from the data. Moreover, it seems that in many cases it is actually better to learn an *overcomplete* basis: a set of $m > n$ vectors $a_1, \ldots, a_m \in \mathbb{R}^n$ so that every example from our data is a sparse linear combination the a_k's.[1]

[1] Even considering the case that the a_m's are a union of two orthonormal bases, such as the standard and Fourier one, already gives rise to many of the representational advantages and computational challenges.

Olshausen and Field [1997] were the first to define this problem - they used a heuristic to learn such a basis for some natural images, and argued that representing images via such an dictionary is somewhat similar to what is done in the human visual cortex. Since then this problem has been used in a great many applications in computational neuroscience, machine learning, computer vision and image processing. Most of the time people use heuristics without rigorous analysis of running time or correctness.

There has been some rigorous work using a method known as "Independent Component Analysis" (Comon [1992]), but that method makes quite strong assumptions on the distribution $\{x\}$ (namely independence). The work of Wang et al. [2015] has given rise to a different type of rigorously analyzed algorithms based on linear programming, but these all required the vector x to be *very sparse*—less than $\sqrt{n}$ nonzero coordinates. The sos method allows recovery in

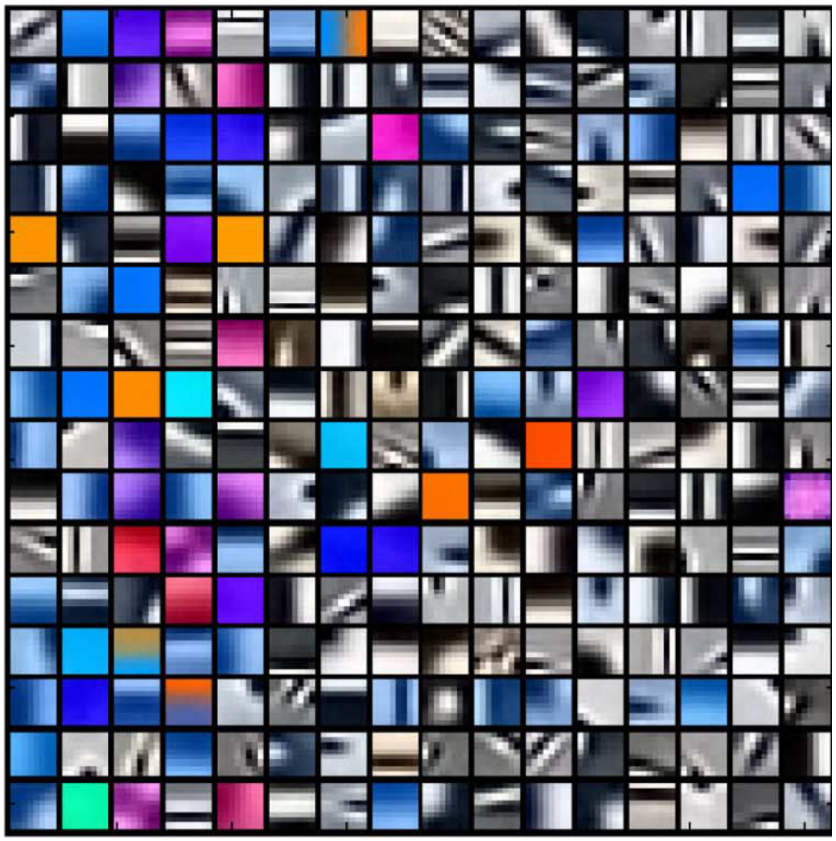

Figure 1: A dictionary for natural images. This is a set of 256 vectors that offer sparse representation of 8 by 8 pixel patches of natural image. Image taken from Mairal et al. [2008]

Figure 2: Using dictionary learning to remove overlaid text from images. The authors used a dictionary learned from natural images and then removed text from the image by making a "low pass filter" over this dictionary, see Mairal et al. [2009]

the much denser case where x has up to εn nonzero coordinates for some $\varepsilon > 0$.

From tensor decomposition to dictionary learning

Suppose that $A = (a_1|\cdots|a_m)$ is a dictionary. For simplicity, we assume that $\|a_i\| = 1$ for all i, and that the a_i's are *incoherent* in the sense that $\langle a_i, a_j \rangle = o(1)$ for all $i \neq j$.[2] Now suppose that we are given many examples $y_1, \ldots, y_M$ of the form $y_i = Ax_i + e$ where $x_1, \ldots, x_M$ are independently sampled from some distribution $\mathcal{D}$ over sparse (or nearly sparse) vectors in $\mathbb{R}^m$ and e is sampled from a noise distribution with small magnitude. In fact, to simplify things further, for the sake of the current discussion, we will ignore the noise and assume that $y_i = Ax_i + e$. We will also assume that the distribution on the coefficients x is *symmetric*, in the sense that $\mathbb{P}[x] = \mathbb{P}[-x]$ and in particular $\mathbb{E}\, m(x) = 0$ for every square-free monomial m. This is not without loss of generality but is a fairly natural assumption on this disrtribution.

[2] The assumption on the norm is without loss of generality while some type of incoherence can be shown to be *necessary* for recovery of the dictionary, regardless of computational issues.

Consider the empirical tensor $\hat{T} = \frac{1}{M} y_i^{\otimes 4}$. If M is large enough, we can assume that it is essentially the same as the expected tensor

$$T = \mathbb{E}\, y^{\otimes 4} = \mathbb{E}(Ax)^{\otimes 4}\,. \tag{2}$$

We will attempt to recover the vectors $a_1, \ldots, a_m$ by doing a tensor decomposition for T. A priori this might seem like a strange approach since the tensor T is *not* going to be proportional to $\sum_{i=1}^m a_i^{\otimes 4}$. Indeed, by expanding out we can see that this tensor is going to be of the form

$$T = \sum_{i=1}^m (\mathbb{E}\, x_i^4) a_i^{\otimes 4} + O(1) \sum_{1 \leq i,j \leq m} (\mathbb{E}\, x_i^2 x_j^2) a_i^{\otimes 2} a_j^{\otimes 2} \tag{3}$$

where we used the fact that the odd moments of x vanish.

Now since x is supposed to be a vector with εm nonzero coordinates, let's consider the simple case where

x_i is equal to ± 1 with probability ε and is equal to 0 otherwise and that the distribution is *pairwise independent*.[3] In this case, $\mathbb{E}\, x_i^4 = \epsilon$ and $\mathbb{E}\, x_i^2 x_j^2 = \epsilon^2$.

[3] It can be shown that some bound on the correlation between two coefficients x_i and x_j is necessary for recovery. For example, it is a good **exercise** to show that if $x_i = x_j$ always then there are two dictionaries A, A' with distinct set of columns such that the distributions $y = Ax$ and $y' = A'x$ are identical.

Now the incoherence assumption implies that the $n^2 \times n^2$ matrix

$$(\sum_i a_i^{\otimes 2})^{\otimes 2} \tag{4}$$

will have spectral norm which is $O(1)$. Hence the tensor T will have the form

$$T = \epsilon \sum_{i=1}^{m} a_i^{\otimes 4} + O(\epsilon^2) T' \tag{5}$$

where T' is a 4-tensor that when considered as an $n^2 \times n^2$ matrix has $O(1)$ spectral norm. Note that the spectral norm of $\sum_{i=1}^{m} a_i^{\otimes 4}$ So if $\epsilon \ll 1$ then what we need is a tensor decomposition algorithm that allows for noise that is small in *spectral* norm. Note that this is a much stricter condition (in the sense of allowing more noise) than the standard notion of the noise being small when considered as a vector (which corresponds to being small in *Frobenius norm*). Luckily, the sos based tensor decomposition algorithms can handle this type of noise.

References

Pierre Comon. Independent component analysis. *Higher-order statistics*, pages 29–38, 1992.

Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Trans. Image Process.*, 17(1): 53–69, 2008. ISSN 1057-7149. doi: 10.1109/TIP.2007.911828. URL http://dx.doi.org/10.1109/TIP.2007.911828.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23): 3311–3325, 1997.

Huan Wang, John Wright, and Daniel A. Spielman. A batchwise monotone algorithm for dictionary learning. *CoRR*, abs/1502.00064, 2015.