Recitation 12

DS-GA 1013 Mathematical Tools for Data Science

Sparse Coding

Solution: Credits: This document is produced (sometimes word to word) from this webpage

We saw that techniques such as Principal Component Analysis (PCA) allow us to learn a complete set of basis vectors efficiently. If the basis vectors are $\varphi \in \mathbb{R}^n$ then we can represent an input vector $x \in \mathbb{R}^n$ as a linear combination of these basis vectors:

$$x = \sum_{i=1}^{k} a_i \varphi_i \tag{1}$$

1. For a given $x \in \mathbb{R}^k$ if φ_is are the vectors found through PCA, are the coefficients a_is in 1 unique?

Solution: yes

In PCA, we get $k = n$. But, we wish to learn an **over-complete** (i.e, $k > n$) set of basis vectors to represent input vectors $x \in \mathbb{R}^n$. The advantage of having an over-complete basis is that our basis vectors are better able to capture structures and patterns inherent in the input data.

2. For a given $x \in \mathbb{R}^k$ if φ_is form an overcomplete basis, are the coefficients a_is in 1 unique?

Solution: No

In sparse coding, we introduce the additional criterion of **sparsity** to resolve the degeneracy introduced by over-completeness.

Here, we define sparsity as having few non-zero components (or having few components not close to zero). The requirement that our coefficients a_i be sparse means that given a input vector, we would like as few of our coefficients to be far from zero as possible. The choice of sparsity as a desired characteristic of our representation of the input data can be motivated by the observation that most sensory data such as natural images may be described as the superposition of a small number of atomic elements such as surfaces or edges. Other justifications such as comparisons to the properties of the primary visual cortex have also been advanced (Highly recommended Olshausen and Field 1996)

We define the sparse coding objective function on a set of m inputs as:

$$\text{minimize}_{a_i^{(j)}, \varphi_i} \sum_{j=1}^{m} \left|\left| \mathbf{x}^{(j)} - \sum_{i=1}^{k} a_i^{(j)} \varphi_i \right|\right|^2 + \lambda \sum_{i=1}^{k} S(a_i^{(j)}) \tag{2}$$

where $S(.)$ is a sparsity cost function which penalizes a_i for being far from zero. We can interpret the first term of the sparse coding objective as a reconstruction term which tries to force the algorithm to provide a good representation of x and the second term as a sparsity penalty which forces our representation of x to be sparse.

3. Let the sparsity cost be L_1 penalty. If φ_i, a_i is a solution which fits the reconstruction part of the objective well, can you derive a new φ_i', a_i' from φ_i, a_i that retains the reconstruction but minimizes the sparsity cost? How can we fix this degeneracy?

Solution: Scale the basis vector by $\alpha > 1$ and scale down the coefficient by the same α. Fix L_2 norm of φ_is.

4. Assume that φ_is are known to us and the sparsity cost function is the L_0 norm. To solve for a_is of a given x we would like to solve the equation:

$$\min_{a_i} \left|\left| \mathbf{x} - \sum_{i=1}^{k} a_i \varphi_i \right|\right|^2 \quad \text{subject to} \quad \|\mathbf{a}\|_0 \leq N \tag{3}$$

   1. How can you solve the optimization problem exactly?
   2. If $N = 1$, how would you solve the problem?
   3. If $N = 2$, can you build on top of your $N = 1$ solution to find the new solution?
   4. Generalize this to give a greedy algorithm to solve the optimization problem.

Solution: To solve exactly, we have to look at all combinations of non-zero coefficients and do least squares with only those coefficients. The combination with the lowest objective will be the solution.

Algorithm: Orthogonal Matching Pursuit OMP

5. The over-complete basis vectors φ is often called atom and set all φs is called a Dictionary. Suggest a way to learn φ and a_i using a training set in 2. After you've learned φs, during inference, how can you find coefficients for a new example?

Solution: Reproduced verbatim from here

Learning a set of basis vectors φ using sparse coding consists of performing two separate optimizations, the first being an optimization over coefficients a_i for each training example $\mathbf{x}$ and the second an optimization over basis vectors φ across many training examples at once. Assuming an L_1 sparsity penalty, learning $a_i^{(j)}$ reduces to solving a L_1 regularized least squares problem which is convex in $a_i^{(0)}$ for which several techniques have been developed (convex optimization software such as CVX can also be used to perform L1 regularized least squares). Assuming a differentiable S(.) such as the log penalty, gradient-based methods such as conjugate gradient methods can also be used. Learning a set of basis vectors with a L_2 norm constraint also reduces to a least squares problem with quadratic constraints which is convex in φ. Standard convex optimization software (e.g. CVX) or other iterative methods can be used to solve for φ although significantly more efficient methods such as solving the Lagrange dual have also been developed.

As described above, a significant limitation of sparse coding is that even after a set of basis vectors have been learnt, in order to "encode" a new data example, optimization must be performed to obtain the required coefficients. This significant "runtime" cost means that sparse coding is computationally expensive to implement even at test time especially compared to typical feedforward architectures.