CIS 6930 – Assignment 2
Nonlinear Approximation

Due on October 20, 2016, at the *beginning* of class

- Please submit your homework in hard copy. For computer problems, submit a written report documenting your experiments. Moreover, you need to submit a printout of the source code you used for running the experiments. If you discuss exercises/problems with others or use resources for any part of the assignment (including ideas, source code), make sure to include proper citation and clear acknowledgments in your report.
- Make use of the Discussion forum on canvas. Asking questions as well as helping others count as class participation.
- Late submissions are penalized by 10% of the grade for each day (maximum of four) past the due date.

1 Best K-Term Approximation

In the previous assignment we learned about the linear approximation where we decide, a priori, which dimensions of the data to keep and which ones to throw away: in the polynomial case, we kept the subspace spanned by the *first* three monomials (i.e., t^0, t^1 and t^2), and in the trigonometric case, we kept the *first* K frequencies in the DCT basis.

The problem is that this approach blindly picks the subspace for approximation (i.e., spanned by the first K basis vectors) regardless of what data we are dealing with. It only works if we are lucky enough that the most of the energy is concentrated in those first few transform coefficients. If we have enough computing resources, we can do better with nonlinear approximation. Nonlinear approximation *adapts* to the data to provide the best low-dimensional approximation possible. Given the expansion in an orthobasis:

$$\mathbf{x} = \sum_{n=0}^{N-1} a_n \mathbf{v}_n, \tag{1}$$

we first sort the coefficients $\{a_n\}$ to obtain another sequence $\{a_n^*\}$ where $|a_0^*| \geq |a_1^*| \geq |a_2^*| \geq \cdots \geq |a_{N-1}^*|$. Let p be the mapping between the indices in the original and the sorted list. The n^{th} term in the sorted coefficients is the $p(n)^{\text{th}}$ term in the original: $a_n^* = a_{p(n)}$. Then the *best K-term approximation* is obtained by keeping the first K terms in the latter sequence and throwing away (i.e., setting to zero) all terms starting from a_K^*. This guarantees that the approximation is using the most important (i.e., largest magnitude) transform coefficients. The best K-term approximation is obtained by:

$$\hat{\mathbf{x}}_K^* = \sum_{n=0}^{K-1} a_{p(n)} \mathbf{v}_{p(n)}. \tag{2}$$

The error in this approximation (via Parseval) is:

$$\|\mathbf{x} - \hat{\mathbf{x}}_K^*\|_2^2 = \sum_{n=K}^{N-1} |a_n^*|^2. \tag{3}$$

The error in this nonlinear approximation will not be bigger than the linear case (often much smaller) since $\{a_n^*\}$ is a descending sequence.

Depending on what type of data we are dealing with and what basis vectors $\mathbf{v}_n$ we are using, the rate of decay of the sorted transform coefficients varies. For a given point $\mathbf{x}$ and a choice of basis, assume we have a model for the decay of sorted coefficients. In other words, we learn about two constants $C > 0$ and $\beta > \frac{1}{2}$ that:

$$|a_n^*| \leq C n^{-\beta} \tag{4}$$

a. Using techniques from calculus (i.e., series) prove that for large N (tends to ∞) the error in this approximation follows:

$$\|\mathbf{x} - \hat{\mathbf{x}}_K^*\|_2^2 \leq \frac{C^2}{2\beta - 1} K^{-2\beta+1}. \tag{5}$$

so that the length of the error, $\|\mathbf{x} - \hat{\mathbf{x}}_K^*\|_2$, decays like $K^{-\beta+\frac{1}{2}}$.

b. If we want to approximate $\mathbf{x}$ with the error tolerance given by ϵ, how many terms do we need to recover in best K-term approximation (i.e., given ϵ how do we find K)?

c. Choose a data point, 1D (e.g., audio signal, temperature variations) or a 2D image (e.g., $a_{m,n}$ from the previous assignment) and establish the rate of decay for it: Experimentally find a pair of numbers C and β that best models the rate of decay in sorted transform coefficients.

2 Sparse Representation and Matching Pursuits

In this exercise we work with a dictionary $\mathbf{D} = [\mathbf{I} \;\; \mathbf{C}]$ that is composed of a $d \times d$ identity matrix $\mathbf{I}$ and a $d \times d$ DCT matrix $\mathbf{C}$. The columns of the DCT matrix (as we have seen in the previous assignment) are discretized cosine vectors. For a sparse representation of a point $\mathbf{x} \in \mathbb{R}^d$, the coefficient vector $\mathbf{a} \in \mathbb{R}^N$ with $N = 2d$ in this dictionary.

a. Implement Matching Pursuit (MP) and Orthogonal Matching Pursuit (OMP) algorithms.

b. Compute the coherence of $\mathbf{D}$: $\quad \mu = \max_{\mathbf{d},\mathbf{d}' \in \mathbf{D}} |\langle \mathbf{d}, \mathbf{d}' \rangle|$

c. Write a procedure that generates a vector $\mathbf{x} = \mathbf{D}\mathbf{a}$ by picking K random positions in $\mathbf{a}$ and sets those positions to random numbers (normal distribution). Then write a procedure that reports the error in recovering $\mathbf{a}$ from $\mathbf{x}$ by using the MP and OMP algorithms.

d. Vary the level of sparsity in a (subset) of the range $1 \leq K \leq d$. For each sparsity K repeat the experiment in part c. several iterations (e.g., 30). Report the probability of success and the mean relative error versus K. In particular, at what sparsity level do these algorithms begin to fail?

e. Repeat this process for a few dimensions, d, and comment on the relationship between the failing sparsity level and d for each of these pursuit algorithms.

f. *Bonus Credits*: Implement the basis pursuit (BP) algorithm (e.g., `CVX` in MATLAB or any package with linear programming or convex programmint) and run a similar analysis for BP. Compare and contrast the performance (chances for accurate recovery and computational cost) the matching pursuit algorithms with basis pursuit.t