Complementary material

Sparse representation

matching pursuit (MP) algorithm

The case of $m < n$

- When $m < n$, the number of bases m is smaller than the signal dimension n.
- In this case, it is impossible to find the coefficients c such that $x = Ac$.
- We usually find $\hat{c}$ instead,

$$\hat{c} = \underset{c}{\operatorname{argmin}} \|x - Ac\|^2 .$$

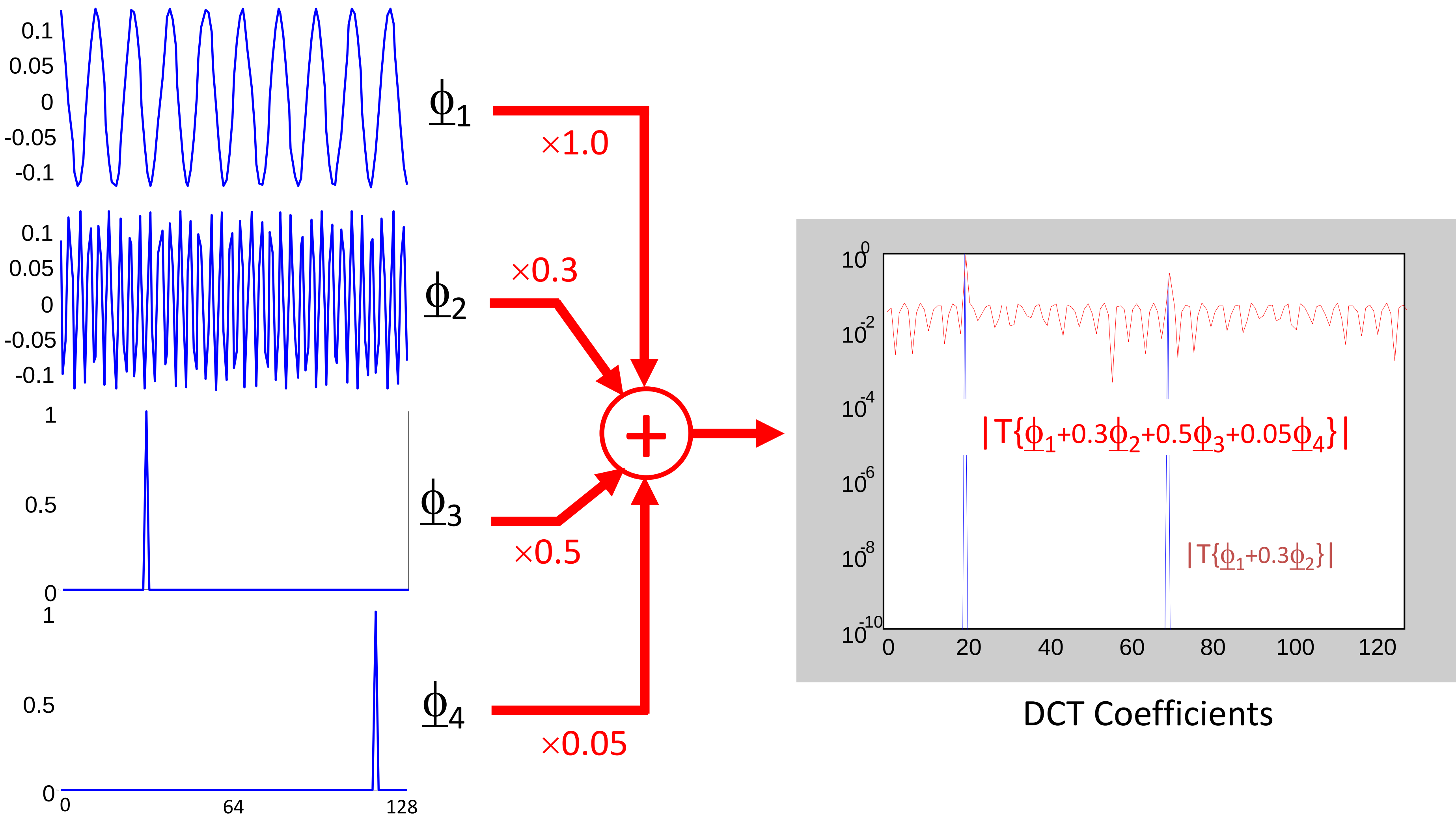
The case of $m > n$

- When $m > n$, the number of bases is greater than that of the signal dimension.
- Therefore, we can find infinite solutions that satisfy $x = Ac$.
- Note that in this case, the bases A are redundant (i.e., the columns of A are linear dependent) or over-complete.
  - The bases can't be orthonormal in this case.

Why Over-Completeness?

- An example is union of bases:
  - Some bases are suitable for harmonic analysis (i.e., frequency component finding), such as the Fourier bases A_F described before.
  - Some bases are suitable for modeling the spikes, such as the natural bases $A_N = I_N$ (where I_N is the $N \times N$ identity matrix).
- Signals could consist of both harmonic parts and spike parts. It is more effective to represent them with the union of different bases, $A_F \cup A_N$.

Union of bases



DCT Coefficients

Over-complete bases (or dictionary)

- Another way to get over-complete bases (also known as *dictionary*) is to learn them from pre-collected training data.

- Sometimes, the training data themselves can serve as the bases (dictionary).

- A basis is also called a codeword in the dictionary.

Coefficients finding under the sparse assumption

- When $m > n$, we usually consider the sparse constraint to avoid the infinite solution problem (i.e., make the solution unique and well posed).

$$x = Ac = [a_1, a_2, \dots, a_m]\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix}$$

- Sparse constraint: Assume that only few coefficients in $\{c_1, c_2, \dots, c_m\}$ are non-zero, and the others are all zeros.
- That is, for a signal x, only few bases are allowed to be activated to represent it.

Sparse solution

- The problem becomes

$$\boxed{\begin{aligned} &\min_{c} \ \|c\|_0, \\ &\text{s.t. } Ac = x. \end{aligned}} \qquad \textbf{P1}$$

where $\|c\|_0$ is the **zero norm** of c, or equivalently, the number of nonzero elements in c. It is also called the **sparse degree** of c.

Sparse solution

- Another related problem that we could consider is (for handling the noisy case)

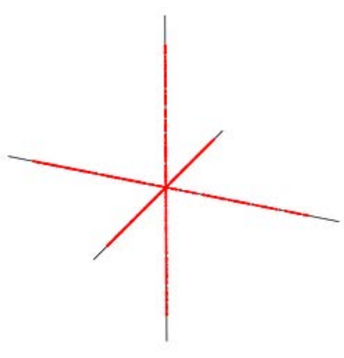
$$\boxed{\begin{aligned}&\min_{c} \|x - Ac\|^2, \\ &\text{s.t. } \|c\|_0 \le k.\end{aligned}} \qquad \textbf{P2}$$

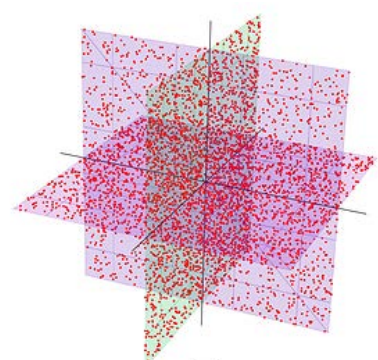
That is, minimize the Euclidean distance from x to a hyperplane spanned by at most k bases in A. k is the sparse degree satisfying that $k \le m$.

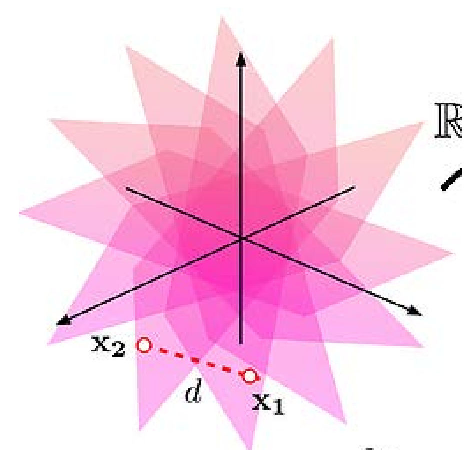
Union of subspace

- Sparse representation is a union-of-subspace model.
- In fact, sparse representation is not necessarily to be used in the case of $m > n$ only. It can also be used when $m \le n$. However, $k \le m$ should always be satisfied (k is the sparse degree).
- It assumes that a k-sparse signal lies on one of the k-dimensional subspaces with the subspace spanned by some of the k bases in A.

For the illustration purpose, we use $n = 3, m = 3,$ and $k = 1, 2$ in the left two figures.







How to solve it?

$$\min_{c} \|c\|_0, \text{ s.t. } Ac = x. \quad \textbf{P1}$$

- The problem is NP-hard and cannot be solved in polynomial time.
- One of the solution is to relax the problem P1 to an $L1$-norm minimization problem.

$$\boxed{\min_{c} \|c\|_1, \text{ s.t. } Ac = x., \quad \textbf{PL1}}$$

where $\|c\|_1$ is the one-norm of c, i.e., sum of the absolute values, $\|c\|_1 = |c_1| + |c_2| + \cdots + |c_m|$.

- Problem PL1 is a **linear programming problem** (because both the objective function and the constraints are linear equations); thus it can be solved in polynomial time.

L-1 norm relaxation

- It can be shown that, under a broad range of conditions, solving the problem PL1 can also find the sparse solution required for the problem P1.
- In signal processing, algorithms solving Problem PL1 is called the **basis pursuit** method. In Statistics, it is called LASSO.
- Besides, there have been many algorithms solving PL1 more efficiently than linear-programming. Eg., iterated reweighted least square (IRLS), and LARS.
- **Focus of the slides**: greedy approaches solving P1 or P2 directly.

Greedy solution for the L0-norm sparse problem

- Considering Problem P1 that is a combinatorial-explosion problem, we seek to solve it via **greedy search** (and find an approximate solution at most time or exact solution under some conditions).
- Greedy search principle: in pattern recognition and data mining, there are three main greedy search strategies,
  - **Sequential forward search**
  - **Sequential backward elimination**

Sequential forward search

- The problem is to select k bases (or codewords) that minimizes a loss function.
- **Sequential forward search (SFS)**
  - First, select a single basis in the dictionary, which minimizes the loss function. Assume that this basis is $a_{s_1},\ s_1 \in \{1 \dots m\}$
  - Then, fix the basis a_{s_1} and keep finding the second basis among the others so that $\{a_{s_1}, a_{s_2}\}$, $s_2 \in \{1 \dots m\} \backslash \{s_1\}$ minimizes the loss with two bases.
  - Repeat the procedure until k bases are selected.

Sequential backward elimination

- **Sequential backward elimination (SBE)**: start from the entire set of bases, $\{a_1, a_2, \ldots, a_m\}$, removing one of the worst basis, i.e., the basis causes the smallest loss increment when it is removed.
  - Then, remove the second basis that is the worst, and repeat the process until $m - k$ bases have been removed.

Matching pursuit (MP) algorithm

- MP algorithm finds the sparse bases by using the SFS principle.
- To simplify the algorithm presentation, assume that the bases $\{a_1, a_2, \dots, a_m\}$ are **pre-normalized** to unit-length vectors $\{b_1, b_2, \dots, b_m\}$ with

$$b_i = \frac{a_i}{\|a_i\|_2},$$

  respectively. Then $\|b_i\|^2 = 1$.
- Because the bases are given, the pre-normalization can always be done in advance.
- Let $B = [b_1, b_2, \dots, b_m]$ be the matrix consisting of these unit-length bases.

Matching pursuit (MP) algorithm

- **The first step** of MP is to solve the following single-basis problem,

$$\min_{c} \|x - Bc\|^2, \ \text{s.t.} \ \|c\|_0 = 1.$$

- That is, find a single basis b_{s_1} in $\{b_1, b_2, \ldots, b_m\}$, such that $\min_{c_{s_1}} \left\|x - c_{s_1} b_{s_1}\right\|^2 \le \min_{c_i} \left\|x - c_i b_i \right\|^2, \forall i \neq s_1$.
- The answer of the single-basis problem is easily solved as

$$\boxed{s_1 = \operatorname*{argmax}_{i \in \{1 \ldots m\}} \left|b_i^T x\right|}$$

(why?)

Single-basis problem

- That is, the solution is obtained by computing the m inner products between the input signal x and the length-normalized basis b_i and taking the absolute values,

$$|b_i^T x_i|, i = 1 \dots m.$$

- Then choose the largest one in $i = 1 \dots m$.

Single-basis problem

- **Proof**: for any b_i, the c_i that minimizes

$$e = \left\| x - c_i \ b_i \right\|^2 = \|x\|^2 - 2c_i b_i^T x + c_i^2 \|b_i\|^2$$
$$= \|x\|^2 - 2c_i b_i^T x + c_i^2,$$

the solution is given by $\frac{\partial e}{\partial c_i} = -2b_i^T x + 2c_i = 0$.

The optimum coefficient occurred when $\hat{c}_i = b_i^T x$.

The minimal $\hat{\boldsymbol{e}} = \|x\|^2 - \left(b_i^T x\right)^2$.

Because $\|x\|^2$ is a constant in bases finding, minimizing e is equivalent to maximizing $\left(b_i^T x\right)^2$; or equivalently, maximizing $|b_i^T x|$ for $i = 1 \ldots\ldots m$.

- In the above, we have selected the first basis, b_{s_1}.

Entire MP algorithm

- Initially, let the **residue signal** be $r_1 = x - c_{s_1} b_{s_1}$, where $c_{s_1} = b_{s_1}^T x$.
- **Iteration**: assuming that l bases $b_{s_1}, b_{s_2}, \dots, b_{s_l}$ have been chosen, $1 \le l \le m$, we seek to select the $(l+1)$-th basis.
- In MP, the $(l+1)$-th basis is chosen as

$$\boxed{s_{l+1} = \operatorname*{argmax}_{i \in \{1,2,\dots m\} \setminus \{s_1, s_2, \dots, s_l\}} |b_i^T r_l|}$$

and then the **residue signal** is modified as

$$\boxed{r_{l+1} = r_l - c_{s_{l+1}} b_{s_{l+1}},}$$

with $\boxed{c_{s_{l+1}} = b_{s_{l+1}}^T r_l.}$

(continue)

- **Stopping criterion**:
  - (for Problem P2) If $l + 1 = k$, i.e., the maximum-allowed sparsity is achieved, then stop.
  - (for Problem P1) If $r_l = 0$, then stop.
  - in practice, we can also stop if the reconstruction error r_l is smaller than a pre-defined threshold.
- Output approximation signal (if k bases are chosen):

$$\boxed{\hat{x} = c_{s_1} b_{s_1} + c_{s_2} b_{s_2} + \cdots + c_{s_k} b_{s_k}.}$$

- Matching pursuit algorithm has been used in many problems to find the matched bases. Eg., in machine learning, image coding, and so on.
- It is easy to implement and the computation is fast.

R. Neff and A. Zakhor. M. Vetterly, ”Very low bit-rate video coding based on matching pursuits,” IEEE Transactions on Circuits and Systems for Video Technology, 7(1):158–171, 1997.

Z. Hussain and J. Shawe-Taylor, “Theory of matching pursuit,” NIPS 2008.

MP and SFS

- MP follows the SFS principle for searching the bases.
- In the iterations of MP, once the bases $b_{s_1}, b_{s_2}, \ldots, b_{s_l}$ have been chosen, the corresponding coefficients are set as $\hat{c}_{s_i}$ that are fixed and remain unchanged.
- Hence, MP applies the SFS principle to select both the bases and coefficients in each iteration, and once they are selected, they cannot be changed.