EE 771 : Recent Topics in Analytical Signal Processing
Paper Review 4

Arka Sadhu - 140070011

April 26, 2018

1 Q1

OMP has the problem that if an incorrect index is added to the set S_n then it stays for all the subsequent iterations. Hence if an incorrect index is added in the OMP algorithm, then s iterations are not enough to recover a vector with sparsity s if following OMP algorithm.

This is solved by CoSaMP algorithm. It takes union of s sparse indices given by support of the previous x and the 2s highest indices in the residuals and recomputes the solution. In this case, even if an incorrect index was added, it can be removed easily in the subsequent steps. Unfortunately, we loose computation time for this.

2 Q2

We want to see the effect of scaling on the algorithm CoSaMP. Since both support of x^n and the highest 2s values of the residual are unaffected by scaling U^{n+1} remains the same. For the next step, it is evident that the solution u^{n+1} will scale by a. If the previous solution was u then the new solution has to be au. Moreover it is noted that since $a > 0$ the order of values in au and u remains the same. This implies $L_s(u) = L_s(au)$ and therefore the new x^{n+1} will also be scaled by the same factor a.

3 Q3

Complexity of OMP vs CoSaMP.

- OMP1 vs CoSaMP1:
  - We note OMP1 requires only the argmax, hence it can be done in O(n) time.
  - For CoSaMP1 we need support of x^n which requires O(n). $L_{2s}(r)$ will require O(nlogn) since it requires sorting.
- OMP2 vs CoSaMP2:
  - OMP2 needs the pseudo inverse of $A_{S^{n+1}}$ which is the most expensive step. Let $s_1 = |S^{n+1}|$. This would take $O(s_1^w)$ time where w is constant for matrix multiplication.
  - CoSaMP2 needs pseudo inverse of $A_{U^{n+1}}$ which is again the most expensive step. Let $u_1 = |U^{n+1}|$. This would take $O(u_1^w)$ where w is as defined above.
  - It is easy to note that s_1 increases linearly over time and is always less than sparsity s of the x-vector. On the other hand, u_1 has atleast more than 2s and can go upto 3s. This is where the main difference of computaiton complexity arises.
- CoSaMP3: This is again hard thresholding and will need sorting which can be done in O(nlogn).

4 Q4

Assuming x_0 is indeed the true solution. We go through each of the steps in the algorithm:

- Support of x_0 will be the set S such that $s = |S|$. The residual would be zero so $L_{2s}(r)$ will simply choose the first 2s components.
- For the argmin step, we will recover the original x_0 as we know it is the correct solution. It will be s-sparse solution.
- Third step, would do nothing and simply choose the s non-zero values. Finally we would essentially get $x_1 = x_0$.