EE16A - Lecture 23 Notes

Name: Felix Su SID: 25794773

Spring 2016 GSI: Ena Hariyoshi

Matching Pursuit Setup

> 1. Received signal $y = \sum_k \alpha_k S^{N_k} \vec{z}_k$ is a linear combination of delayed versions of the signals sent z_k where S^{N_k} is the circular shift matrix
> 2. Sparse Representation: The linear combination (received signal) of the delayed messages contains very few terms relative to the max number of broadcasted signals.

- Scenario: A lot of players are trying to communicate with you, but you can only receive a few at the time.
- Say you have$L = 2000$ players (trying to communicate), each sends a message $\vec{z}_k \in \mathbb{R}^{N=400}$
- Usually $L > N$ and typically $L >> N$
- Received vector $y = \sum_k \alpha_k S^{N_k} \vec{z}_k$, which is a linear combination of delayed versions of the signals sent z_k where S^{N_k} is the circular shift matrix.
- Sparse Representation: The linear combination (received signal) of the delayed messages contains very few terms relative to the 2000 possible terms.

Sparsity (Using Dictionary)

> 1. Use a larger, more specific dictionary to make sparser messages

- The larger the dictionary is, the more specific descriptions can be and the less words you need to use to convey a message.
- Dictionary contains all the circularly shifted forms of the messages, so it will be of size $L \times N$ (no. of msgs $\times$ no. of possible shifts = msg. vec length)
  - Called a **Redundant Dictionary**
  - $D = \{\vec{z}_1 S_{z_1} \dots \vec{z}_1 S_{z_1}^{N-1} | \vec{z}_2 S_{z_2} \dots \vec{z}_2 S_{z_2}^{N-1} | \dots \vec{z}_n S_{z_n}^{N-1}\}$
  - Assume: D is complete and includes N linearly independent vectors
  - Assume: Every vector is unit length, $\|\vec{z}_1\| = 1$
  - Cannot Assume: Orthogonality
- Now let $D = \{\phi_1, \dots, \phi_T\}, T =$ Dictionary size
- Need to figure out which subset of ϕ_k's is represented in the received signal y.
- Simplification: Assume no delay, so $y = \sum_k \alpha_k \vec{z}_k$ and $D = \{\phi_1 = \vec{z}_1, \dots, \phi_L = \vec{z}_L\}$
- Residuals will approach 0.

Matching Pursuit Algorithm

> 1. Take observation vector y as the initial residual vector $r^{[0]}$
> 2. Project each residual $r^{[m-1]}$ onto the space of each of the vectors in the dictionary D and pick the maximum projection. (Choose a $\phi_k \in D$ s.t. $\vec{v}_m = \text{ argmax}_k |\langle \vec{r}^{[m-1]}, \phi_k \rangle|$)
> 3. Write out equation for residual: $r^{[m-1]} = \langle \vec{r}^{[m-1]}, \vec{v}_m \rangle \vec{v}_m + \vec{r}^{[m]}$ where $r^{[m]} \perp \vec{v}_m$
> 4. Decompose the next residual ($r^{[m]}$)
> 5. Resulting signal at iteration M is $\vec{y} = \sum_{m=0}^{M-1} \langle r^{[m]}, \vec{v}_{m+1} \rangle \vec{v}_{m+1} + \vec{r}^{[M]}$

- A greedy algorithm: Step by step. At each step, search for a local optimum to reach a global optimum.
- Goal: Estimate $y \in \mathbb{R}^N$ by finding the main signal contributors
  - Let $M =$ no. of terms in the estimation (signal contributors)
  - Let $D = \{\phi_l\}, L = 1, \dots, L$ where $L \geq M$ and $L > N(\phi_L \in \mathbb{R}^N)$
  - D is complete and $\|\phi_l\| = 1$
- At each step you have **Residue of** y=$\vec{r}^{[0]}$ (unaccounted portion)
- Step 0: $\vec{r}^{[0]} = y$
- Step 1: Choose a $\phi_k \in D$ s.t. $\vec{v}_1 = \text{ argmax}_k |\langle y, \phi_k \rangle|$
  - Choose ϕ_k that gives the largest projection of y onto ϕ_k
  - Write $y = \langle \vec{y}_1, \vec{v}_1, \vec{\rangle} v_1 + \vec{r}^{[1]}$ where $r^{[1]} \perp \vec{v}_1$
  - $\|\vec{y}\|^2 = \|\langle \vec{y}_1, \vec{v}_1, \vec{\rangle} v_1\|^2 + \|\vec{r}^{[1]}\|^2 = |\langle \vec{y}_1, \vec{v}_1, \rangle|^2 \|\vec{v}_1\|^2 + \|\vec{r}^{[1]}\|^2$
  - $\|\vec{y}\|^2 = |\langle \vec{y}_1, \vec{v}_1, \rangle|^2 + \|\vec{r}^{[1]}\|^2$ (becuase $\vec{v}_1$ has unit length)
- Step 2: Decompose $\vec{r}^{[1]}$: Find the vector in D that best represents $\vec{r}^{[1]}$
  - Choose a $\phi_k \in D$ s.t. $\vec{v}_2 = \text{ argmax}_k |\langle \vec{r}^{[1]}, \phi_k \rangle|$
  - $\vec{r}^{[1]} = \langle \vec{r}^{[1]}, \vec{v}_2 \rangle \vec{v}_2 + \vec{r}^{[2]}$ where $r^{[2]} \perp \vec{v}_2$
  - $r^{[2]}$ may not be (probably not) $\perp \vec{v}_1$
- Step m: Decompose $\vec{r}^{[m-1]}$
  - $\vec{v}_y^{[m-1]} = \langle \vec{r}^{[m-1]}, \vec{v}_m \rangle \vec{v}_m + \vec{r}^{[m]}$
  - To stop at iteration M, express $\vec{y} = \sum_{m=0}^{M-1} \langle r^{[m]}, \vec{v}_{m+1} \rangle \vec{v}_{m+1} + \vec{r}^{[M]}$
- Residuals approach 0
  - $\|\vec{y}\|^2 = \sum_{m=0}^{M-1} |\langle r_y^{[M]}, \vec{v}_{m+1} \rangle|^2 + \|\vec{r}_y^{[M]}\|^2$
  - Fixed term on left
  - Sum of positive terms on right plus risidual
  - As sum increases last residual must decrease for each step

Another View of the Matching Pursuit Algorithm

> 1. If A_m has Orthonormal Columns: $A_m^T A_m = I$

- Start with $y = r^{[m]}$ (observed measurement)
- Create a matrix A_m at each step that has the columns of the vectors in the dictionary you have obtained up to that point. (Matrix of principal directions up to this step)
- Start from $m = 1$: while $r^{[m]} \neq 0$, look for principal direction: $\vec{v}_m = \text{argmax}_k |\langle r^{[m]}, \vec{z}_k \rangle|$
  - argmax: Scan all indices k, on k for which the inner product is the largest, use that inner product and assign $\vec{v}_m = \vec{z}_k$ as the mth principal direction
- Augment matrix of principal directions: $A_m = \left[A_{m-1} | \vec{v}_m\right]$
- Project y onto sol. space of A_m: $\vec{y} = A_m \alpha_m + \vec{\varepsilon}$
- Approximate y in Least Squares sense
  - $\vec{y}_m = A_m \hat{\alpha}_m$ where $\hat{\alpha}_m$ is the LS soln to $A_m \alpha_m = \vec{y}$
  - $\hat{\alpha}_m = (A_m^T A_m)^{-1} A_m^T \vec{y}$
  - $\vec{y}_m = A_m (A_m^T A_m)^{-1} A_m^T \vec{y}$
  - $\vec{r}_m = \vec{y} - \vec{y}_m$
  - Problem: When columns of A_m, the principal directions $\vec{v}_k$ are not orthogonal, the LS soln, is very costly.
- Keep iterating forward: $m = m + 1$ until end.