Speech Dereverberation based on Convex Optimization Algorithms for Group Sparse Linear Prediction

JAYANT RANGI(MA17BTECH11006)
RITESH YADAV(MA17BTECH11009)

March 6, 2019

Introduction :

Speech Dereverberation has become an integral component of front end processing techniques for **automatic speech recognition (ASR)**. In particular, the recent advent of smart loudspeakers like the Amazon Echo, Google Home, and Sonos One, has pushed the robustness required in far-field ASR, as the user expects the same level of performance in multiple condition, including being at different distances in different acoustic environments.**This makes Dereverberation one of the most prominent algorithm for enabling far-field human-computer interaction.**

- **Dereverberation** is the process by which the effects of reverberation(**i.e.** echo, resonance) are removed from sound.

Motivation :

- Speech dereverberation fundamental for enabling far-field **human computer** interaction, particularly with the recent advent of smart loudspeaker devices(**e.g.** Amazon echo, Apple siri).

Figure: Amazon Echo

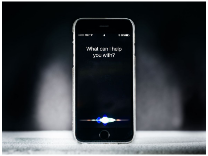

Figure: Apple Siri

Motivation :

- Blind methods based on multi-channel linear prediction (MCLP) applied in the **STFT(Short Fourier Transform)**-domain particularly effective for the task:
  - no prior knowledge of the room acoustics,
  - relatively easy and cheap to implement.

- Popular MCLP-based methods look for a sparse desired speech signal, assuming reverberation as a convolutive process (approximated by the predicted speech) on a **STFT bin-by-bin basis**. This is done by applying **nonconvex** algorithms.

- We propose alternative formulations for sparse approximation based on convex optimization

Fundamentals :

We consider an acoustic system composed of one speech point source and M microphones. The signal at the *m*-th microphone at time n is :

$$x_m(n) = \sum_{m=1}^{M} r_m(n) * s(n) + e_m(n)$$

where *s(n)* is the clean speech signal, $r_m(n)$ is the **RIR(Room Impulse Response)** between the speech source and the *m*-th microphone, and $*$ is the convolution operator.

Fundamentals :

We focus our attention on so-called **utterance-based batch processing** techniques where a full reverberant speech file is processed all at once. Denoting *s(k;n)* as the STFT of the clean speech, with frame index n $\in$ (1,.....N) and frequency bin, index k $\in$ (1,....K) the reverberant speech signal at the *m*-th microphone becomes :

$$x_m(k,n) = \sum_{l=0}^{L_h-1} h_m(k,l)\mathrm{s(k,n\text{-}l)} + \mathrm{e}_m(k,n)$$

where $h_m(k;l)$ models the acoustic transfer function between the speech source and the *m*-th microphone in the *k*-th frequency bin with length L_h.

MCLP-based Dereverberation:

The model divides the time-domain convolution into a set of convolution in the time-frequency domain and has been widely adopted in the dereverberation literature. Given the general assumption of ignoring the noise term, we can rewrite the equation as:

MCLP-based Dereverberation

$$x_m(k,n) = \sum_{l=1}^{\tau-1} h_m(k,l)s(k,n-l) + \sum_{l=\tau}^{L_g-1} h_m(k,l)s(k,n-l)$$

- $n \in 1, ..., N$ frame index, $k \in 1, ..., K$ frequency bin index
- s(k,n): clean speech
- $d_m(k,n)$: desired speech
- $r_m(k,n)$: reverberation term
- $h_m(k,l)$ Acoustic Transfer Function between the speech source and m-th microphone
- τ: Delay to model direct speech and early reflections
- L_g: prediction order

MCLP-based Dereverberation

Desired speech signal using M predictors (order($L_g - 1$)) :

$$d_m(k,n) = x_m(k,n) - \sum_{i=1}^{M} \sum_{l=0}^{L_g-1} x_i(k, n-\tau-l) g_{m,i}(k,l)$$

$g_{m,i}$: l-th prediction coefficient between the i-th and the m-th channel

MCLP-based Dereverberation

The equivalent model in matrix notation is:
$\mathbf{D}(k) = \mathbf{X}(k) - \mathbf{X}_\tau(k)\mathbf{G}(k)$ with

- $\mathbf{D}(k) = [d_1(k), ..., d_m(k)]$
- $d_m(k) = [d_m(k, 1), ...d_m(k, N)]^T$
- $X(k) = [x_1(k), ...x_M(k)]$
- $x_m(k) = [x_m(k, 1), ...x_m(k, N)]^T$
- $X_\tau(k) = [X_{\tau,1}(k), ..., X_{\tau,M}(k)]$
- $G(k) = [g_1(k), ...g_M(k)]$
- $g_m(k) = [g_{m,1}(k, 0), ..., g_{m,1}(k, L_g - 1), ...g_{m,M}(k, 0), ..., g_{m,M}(k, L_g - 1)]^T$
- $X_{\tau,m}(k)$ is the convolution matrix of $x_m(k, n - \tau$
- The prediction matrix is $G(k) = [g_1(k), ..., g_M(k)]$

$\mathbf{G}$(k) is then found by solving the optimization problem:

$$\hat{G} = \underset{\mathbf{G}}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{X}_\tau \mathbf{G}||_{p,1}^1 + \alpha ||\mathbf{G}||_{1,1}^1$$

- For p=1 , equation is a element-wise regularized least-sum-of-absolute
- For p=2 , equation is a group LASSO problem

Norm is a convex function so, this problem is now converted into convex optimization problem.

Proving that norm is a convex function:

Triangle Inequality: $||\lambda$x+(1-$\lambda)y|| \leq \lambda||x|| + (1-\lambda)||y||$

So, $f(\lambda$x+(1-$\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$ where f(x)=$||x||$

,$\lambda \in [0,1]$ which is a property of a convex function

Alternating Direction Method of Multipliers :

- A method :
  - with good robustness of method of multipliers
  - which can support decomposition
- **Decomposable method of multipliers**

ADMM problem form (with f, g convex) :

minimize $f(x) + g(x)$
subject to $\mathrm{A}x + \mathrm{B}z = c$

– two sets of variables, with separable objective
$L_p(x, y, z) = f(x)+g(z)+y^T(Ax + Bz - c) + (\rho/2)||Ax + Bz - c||_2^2$

ADMM :
$x^{k+1} := argmin_x L_p(x, z^k, y^k)$
$z^{k+1} := argmin_z L_p(x^{k+1}, z, y^k)$
$y^{k+1} := y^k + \rho(Ax^{k+1} + Bz^{k+1} - c)$

Alternating Direction Method of Multipliers :

- if we minimized over x and z jointly, reduces to method of multipliers
- instead, we do one pass of a **Gauss-Seidel method**
- we get splitting since we minimize over x with z fixed, and vice versa

ADMM and Optimality conditions

optimality conditions (for differentiable case):
primal feasibility: Ax + Bz - c = 0
dual feasibility: ∇f(x) + A$^T y = 0$,∇g(z) + B$^T y = 0$
Since z^{k+1} minimizes $L_p(x^{k+1}, z, y^k)$
$0 = \nabla \mathrm{g}(\mathrm{z}^{k+1}) + B^T y^k + \rho \mathrm{B}^T (Ax^{k+1} + Bz^{k+1} - c)$
$0 = \nabla \mathrm{g}(\mathrm{z}^{k+1}) + B^T y^{k+1}$

So, with ADMM dual variable update, $(x^{k+1}, z^{k+1}, y^{k+1})$ satisfies second dual feasibility condition.
Primal and first dual feasibility are achieved as k$\rightarrow \infty$

ADMM with scaled dual variables

combine linear and quadratic terms in augmented Lagrangian

$L_p(x,y,z) = f(x)+g(z)+y^T(Ax+Bz-c)+(\rho/2)||Ax+Bz-c||_2^2$

$L_p(x,y,z) = f(x)+g(z)+(\rho/2)||Ax+Bz-c||_2^2 +$ const.

with $u^k = (1/\rho)y^k$

ADMM (scaled dual form):

$$x^{k+1} := \underset{x}{\operatorname{argmin}}(f(x)+(\rho/2)||Ax+Bz^k-c+u^k||_2^2)$$

$$z^{k+1} := \underset{z}{\operatorname{argmin}}(g(z)+(\rho/2)||Ax^{k+1}+Bz-c+u^k||_2^2)$$

$$u^{k+1} := u^k + (Ax^{k+1}+Bz^{k+1}-c)$$

Sources :

- **SPEECH DEREVERBERATION BASED ON CONVEX OPTIMIZATION ALGORITHMS FOR GROUP SPARSE LINEAR PREDICTION**
  Daniele *Giacobello*[1] and Tobias Lindstrom *Jensen*[2]

- **Alternating Direction Method of Multipliers**
  Prof S. Boyd EE364b, Stanford University