Test for a algorithm about AP selection

李文新
王浩

2016/06/05

1 Introduction

The characteristic of AP_i can be described with a complex parameter $Z_i = p_i e^{2i\phi_j}(0 \leq \phi_i \leq 2\pi)$. p_j represent the distinctiveness of signal strength influenced by signal gradient and noise. The direction is $2\phi_j$, double of signal gradient direction. After substituting Z_i, we should maximize:

$$\mathcal{F}(\mathcal{V}_n) = \{(\sum_{i\in\mathcal{V}_n} |Z_i|)^2 - |\sum_{i\in\mathcal{V}_n} Z_i|^2\}$$

The optimal strategy is denoted as $\mathcal{V}_n^*$,$\mathcal{V}_n^* \in \mathrm{U^n}$,where

$$\mathcal{V}_n^* = \underset{\mathcal{V}_n\in\mathrm{U^n}}{\mathrm{argmax}}\mathcal{F}(\mathcal{V}_n)$$

2 Algorithm

The Algorithm to be tested is:

Algorithm 1: Near Optimal Strategy for Fingerprints Reporting

Require: complex number vector $(z_1, z_2 \cdots z_m)$ where $z_j = p_j e^{i\phi_j}(0 \leq \phi_j \leq 2\pi)$ represent the ith AP.
Ensure: A measurement sequence $(n_1, n_2 \dots n_l)$, where $1 \leq n_i \leq l$ represent choosing the n_ith AP.
- $S \leftarrow \emptyset$
- $w_i = 0(1 \leq i \leq lm)$
- $z^* = 1$
- **for** $i = 1$ to lm **do**
  - $\sigma(i) \leftarrow \lfloor \frac{i-1}{l} + 1 \rfloor$
- **end for**
- **for** $i = 1$ to l **do**
  - **for** $i \in [lm]\backslash S$ **do**
    - $w_j \leftarrow w_j + p_{\sigma(j)} |z^*| \sin^2(\phi_{\sigma(j)} - \arg(z^*))$
  - **end for**
  - $i^* \leftarrow \underset{i\in[lm]\backslash S}{\arg\max}\, w_i$
  - $z^* \leftarrow s_{\sigma(i^*)}$;
  - $S \leftarrow S \cup \{i^*\}$;
- **end for**
- **return** $(\sigma(i))_{i\in S}$

3 Test

We showcase the efficiency of the optimal AP selection strategy by comparing it with random selection.*select* show our algorithm, when *select_rand* show the random selection.
We know $z_j = p_j e^{i\phi_j}$. So, when p_j and ϕ_j are in Random Uniform Distribution in interval $[0, 1]$ and $[0, 2\pi]$, the data z_j is also in.
In matlab code:

```
m = 6; % m = 6,12,18
p = rand(m,1);
```

```
phi = 2*pi*rand(m,1);
z = p.*exp(sqrt(-1).*phi);
```

Our algorithm in matlab:
We input the AP vector z and sequence number l:

```
function [res,vector] = select(z,l)
m = length(z);
S = [];
w = zeros(l*m,1);
sigma = zeros(l*m,1);
L = 1:1:l*m;
z_ = 1 + 0i;

for i = 1:l*m
    sigma(i) = floor((i-1)/l+1);
end

for i = 1:l
    for j = setdiff(L,S)
         w(j) = w(j) + abs(z(sigma(j))) *abs(z_) *
                         ( sin(angle(z(sigma(j))) - angle(z_)))^2;
    end
    [value,i_] = max(w);
    z_ = z(sigma(i_));
    S = union(S,[i_]);
    w(i_) = 0;
end

vector = [];
for i = S
    vector = [vector;[sigma(i)]];
end
res = sum(abs(z(vector)))^2 - abs(sum(z(vector)))^2;
```

The output `res` is the value of $\mathcal{F}(\mathcal{V}_n)$, `vector` is the measurement sequence.
When selecting randomly, we calculate 300 times in order to get more accuracy.

```
for i=1:300
    vector = randi([1 m],l,1);
    res = res + sum(abs(z(vector)))^2 - abs(sum(z(vector)))^2;
end
res = res/300;
```

Sequence number l is integer for $m/2$ to $3m/2$ ($m = 6, 12, 18$).

```
m = length(z);
v = [];
v_ = [];
for l=m/2:3*m/2
    [res,vector] = select(z,l);
    v = [v;[res]];
    v_ = [v_;[select_rand(z,l)]];
end
```

The valuable `v` represent the value of $\mathcal{F}(\mathcal{V}_n)$ selected by our algorithm, and `v_` represent the value selected randomly. We should save that value to plot them.
Just using **gunplot**, we get 3 graphs when $m = 6, 12, 18$.

Obviously, $\mathcal{F}(\mathcal{V}_n)$ get greater value by our algorithm.

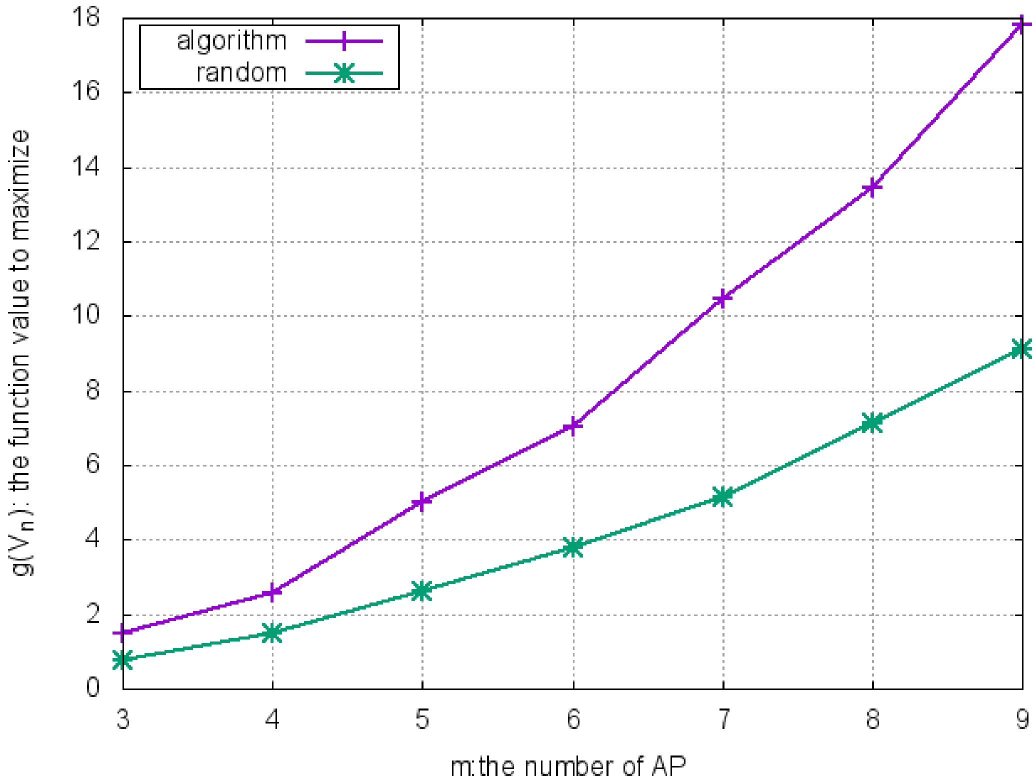

Figure 1: m = 6

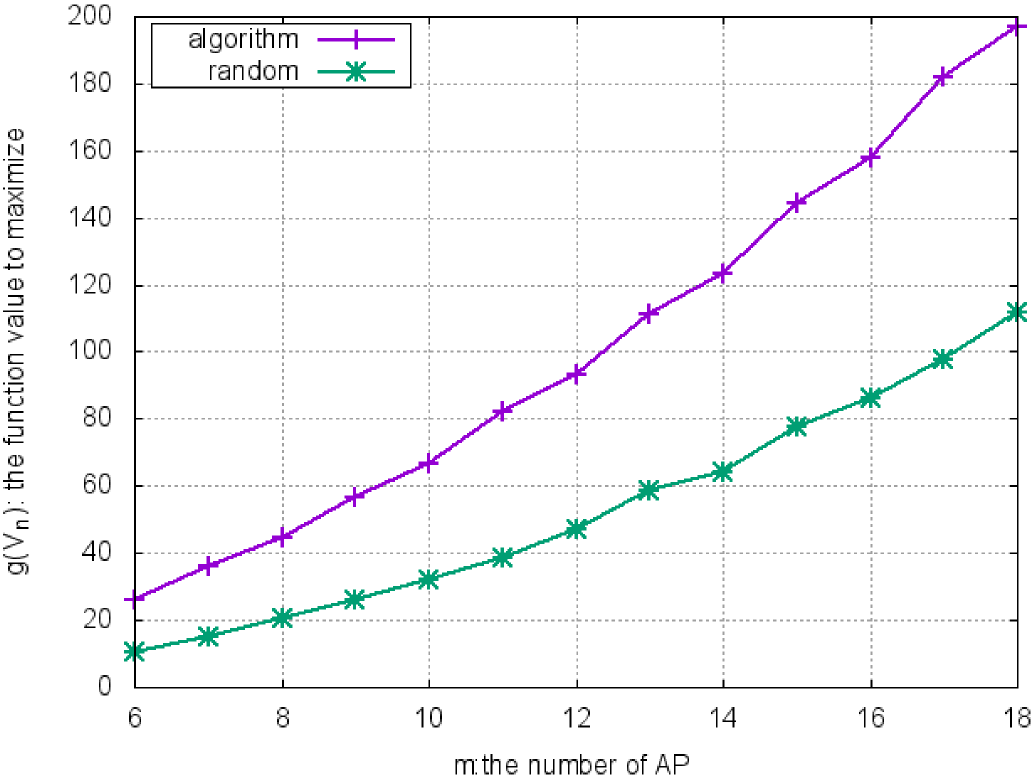

Figure 2: m = 12

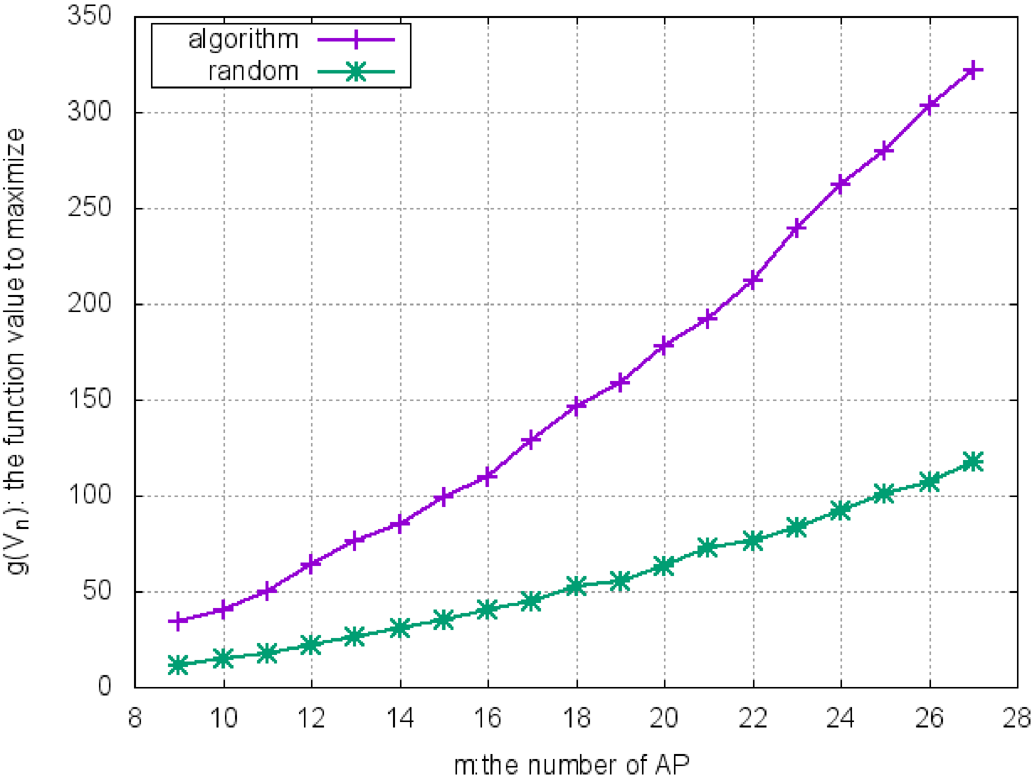

Figure 3: m = 18