In The Name of GOD

A Three-dimension Localization in Wireless Sensor Network Nodes Based on SVM

Electrical Engineering Department of
Shahid Beheshti University

Course name: Pattern Classification

Advisor: Dr.Gharavian
Lecturer: A.Hajihoseini

June 2015

Localization

Methods for acquiring the position of a target node :

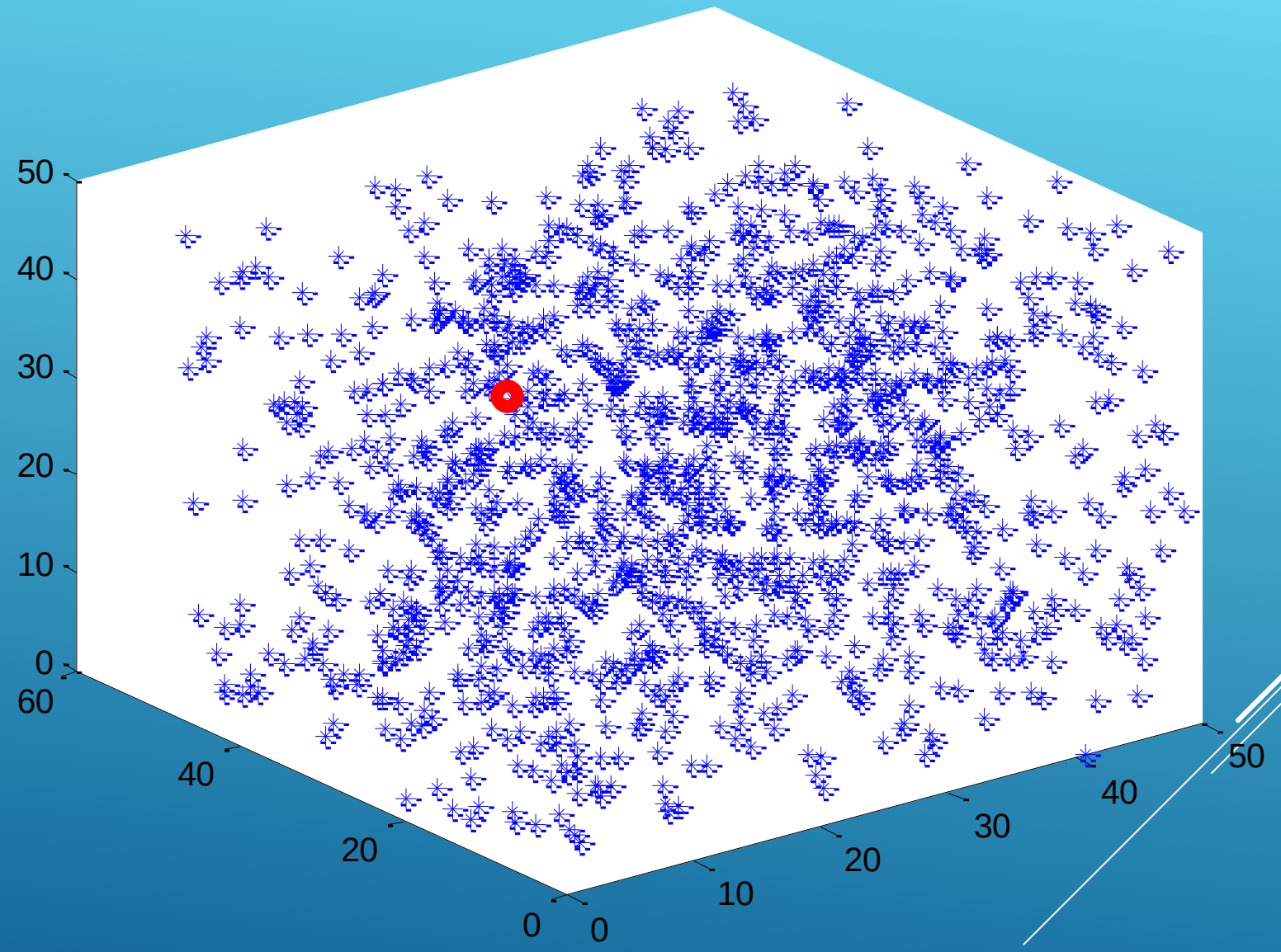
- Time-of-Arrival (TOA)
- Time-difference of Arrival (TDOA)
- Angle of Arrival (AOA)
- Received Signal Strength Indicator (RSSI)

$$P_R[\mathrm{dB}] = P_0 - 10n_p \log_{10}\left(\frac{d}{d_0}\right) + X,$$

$$\hat{d}_n = d_0\ 10^{\left(\frac{P_0 - P_{R,n}}{10n_p}\right)}$$

Localization

- Anchor nodes
- Unknown nodes



Support Vector Machine

$$f(x) = \mathrm{sgn}\{\sum_{i=1}^{l} \alpha_i^* y_i K(x_i, x) + b^*\}$$

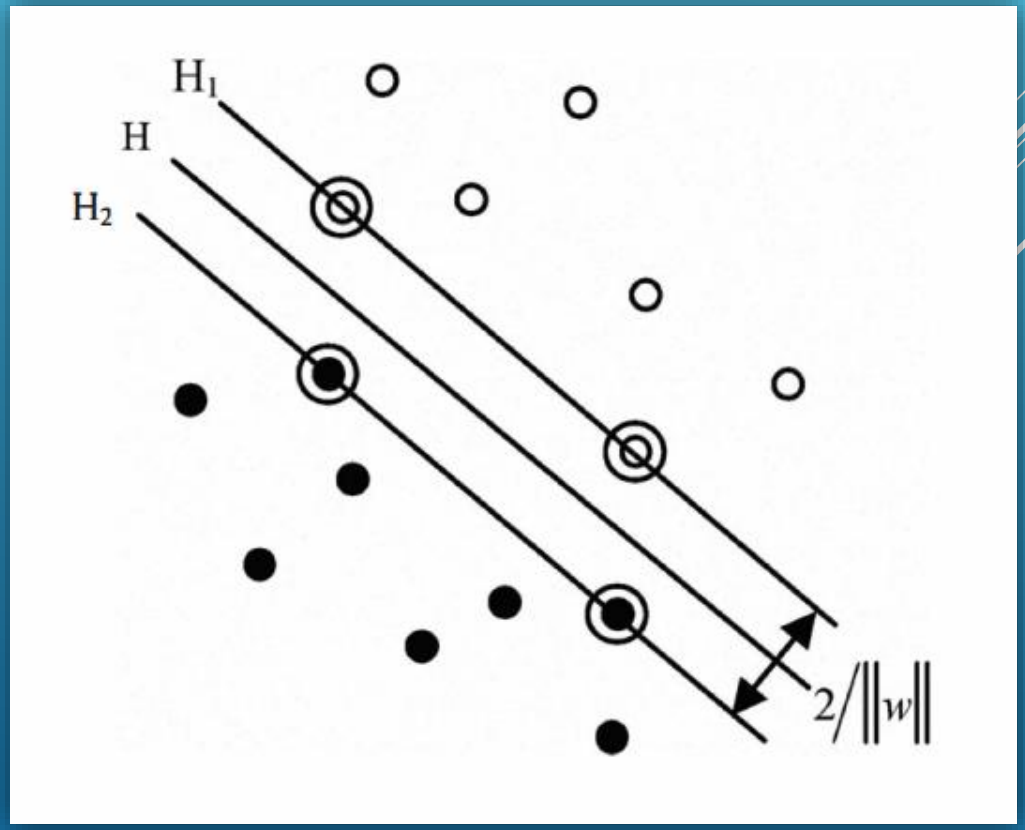
x_i : support vector

y_i : class lable {-1,1}

K(xi,x) : kernel function

α_i^*: Lagrange multiplier

b^*: classification threshold



Modelling for SVM

Anchor nodes $\left[h(S_i,S_1),h(S_i,S_2),\cdots,h(S_i,S_k)\right]$

In each dimension we have M classes:

$\{cx_1,cx_2,\ldots,cx_M\}$

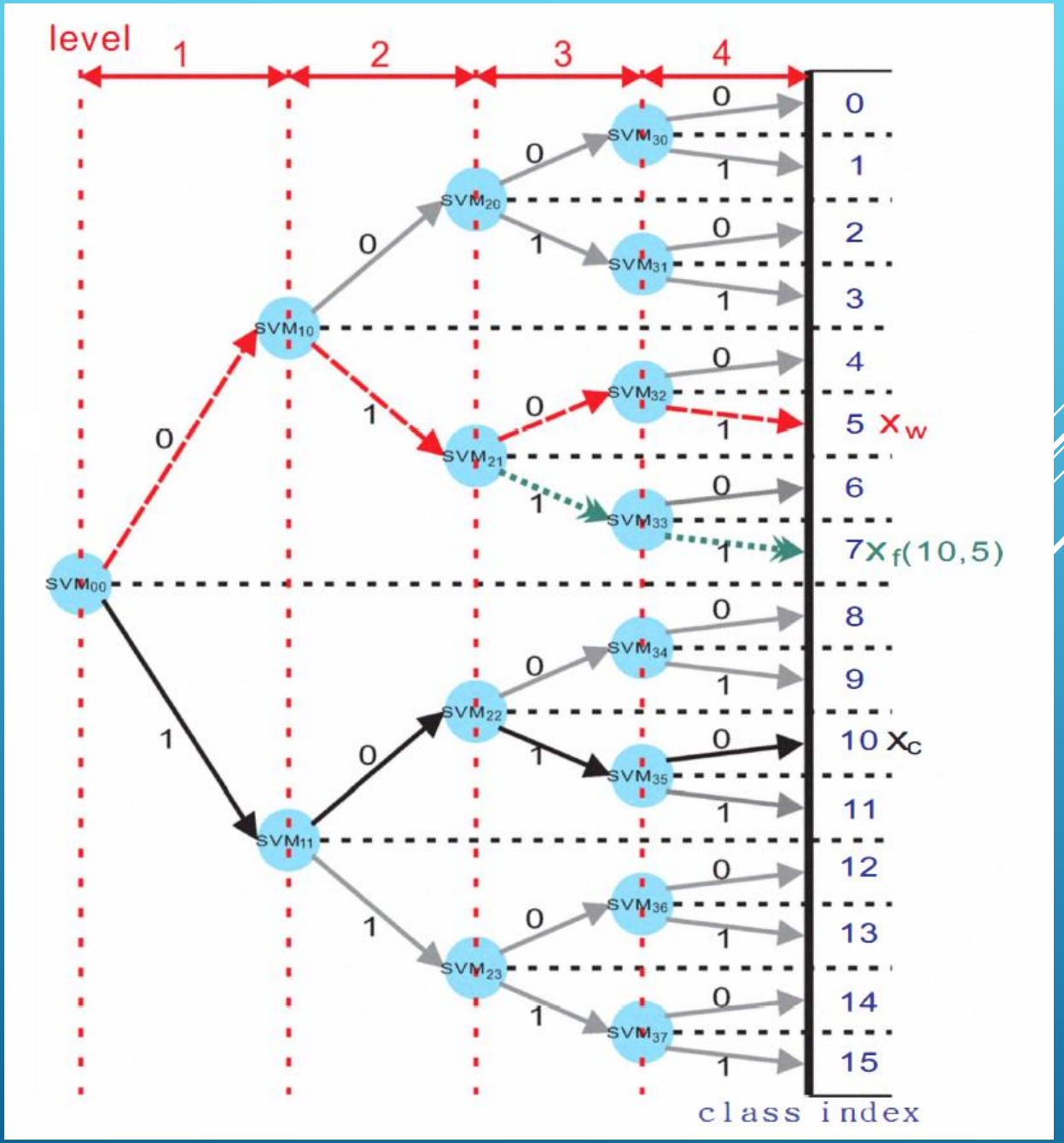
$\{cy_1,cy_2,\ldots,cy_M\}$

$\{cz_1,cz_2,\ldots,cz_M\}$

Training Data $\left\{\left[h(S_i,S_1),h(S_i,S_2),\cdots,h(S_i,S_k)\right],cx_i\right\}$

SVM Model in each dimension: $\left\{x_{ix},\alpha^*_{ix},b^*_x\right\}$

Multiclass SVM



MatLab Code

```
%% parameters
N           = 1251;                 % number of nodes
area        = 50;                   % length of environment
M           = 15;                   % number of classes
R           = 20;                   % communication radious
k           = 1250;                 % number of anchors
nodes_loc   = area*rand(3,N);       % location of nodes
P0          = 0;                    % in dBm
d0          = 1;                    % in m
np          = 2;
sigma_db    = 2;                    % in dBm
m           = log2(M+1);            % classification level
```

MatLab Code

```
%% distance calculations
main_distance = zeros(N,k);
for i=1:N
    for j=1:k
        main_distance(i,j) =
                sqrt(sum((nodes_loc(:,i) - nodes_loc(:,j)).^2));
    end
end
X     = sigma_db * randn(N,k);
d     = main_distance;
P_R   = P0 -10*np*log10(d/d0) + X;
dHat = d0*10.^((P0-P_R)/(10*np));
```

MatLab Code

```
%% classification of x
D = area; a_x = 0; b_x = D; threshold = area/2;
for j = 1:m
    group_x                 = floor(nodes_loc(1,:)/threshold).';
    group_x(group_x>=1)     = 1;
    struct_x                = svmtrain(dHat(1:k,:),group_x(1:k));
    label_x                 =
svmclassify(struct_x,dHat(k+1:N,:));
    if label_x == 1
        a_x         = threshold;
        threshold = (b_x + a_x)/2;
    else
        b_x         = threshold;
        threshold = (b_x + a_x)/2;
    end
end
```

MatLab Code

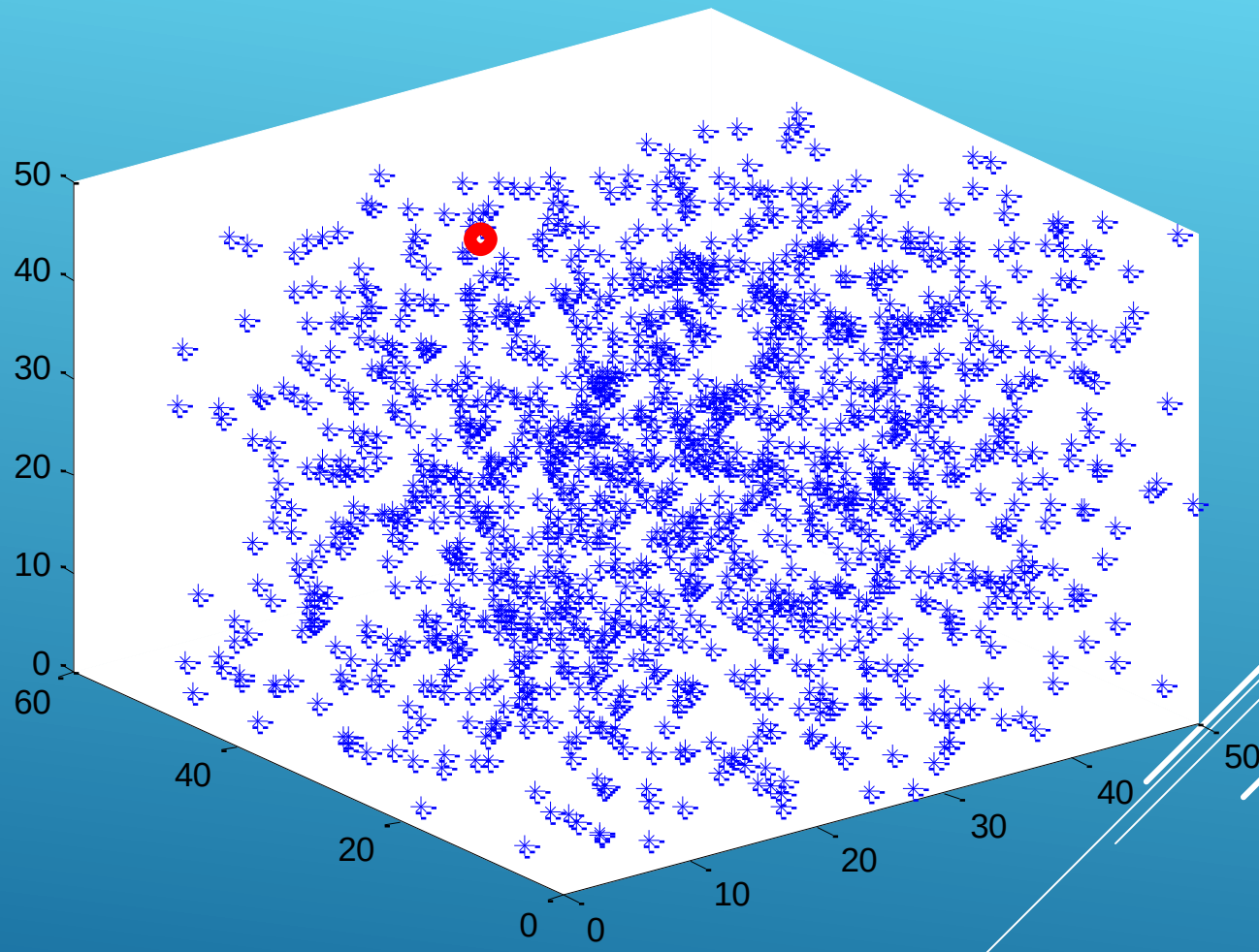
```
%% results
x = (b_x + a_x)/2;
y = (b_y + a_y)/2;
z = (b_z + a_z)/2;
error = sqrt(sum(([x;y;z]-nodes_loc(:,N)).^2));
 clc
disp([num2str(x),'     ',num2str(nodes_loc(1,k+1:N))])
disp([num2str(y),'     ',num2str(nodes_loc(2,k+1:N))])
disp([num2str(z),'     ',num2str(nodes_loc(3,k+1:N))])
disp(['error:    ',num2str(error)])

plot3(nodes_loc(1,1:k),nodes_loc(2,1:k),nodes_loc(3,1:k),'b*')
hold on
plot3(nodes_loc(1,k+1:N),nodes_loc(2,k+1:N),nodes_loc(3,k+1:N)
,'ro','linewidth',4)
```

Results

۱۷.۶۵۱۳	۱۸.۱۵۶۳
۳۷.۷۱۶۶	۳۷.۶۲۵
۴۶.۵۰۶۵	۴۶.۱۸۷۵

ERROR: ۰.۶۰۴۲۶



Thanks For Your Attention