Robot Learning
5. RANSAC

Jeong-Yean Yang

2020/10/22

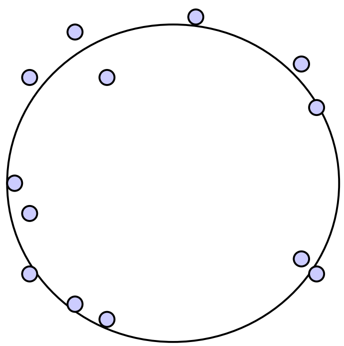
5

Nonlinear Regression with Optimization Method

Modeling with Least Square Eq.

- Find the proper Circle for the given points.
- Cost Function

$$Model: x^2 + y^2 = r^2$$



$$J = J(x_0, y_0, r)$$
$$= \sum_i^N \left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right]^2$$

 $X_i = (x_i, y_i) \in R^2$

Minimizing Error with Gradient Descent method

- Gradient of J

$$J = J(x_0, y_0, r)$$
$$= \sum_i^N \left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right]^2$$

$$\nabla J = \nabla J(x_0, y_0, r)$$
$$= \frac{\partial J}{\partial x_0}\hat{x}_0 + \frac{\partial J}{\partial y_0}\hat{y}_0 + \frac{\partial J}{\partial r}\hat{r}$$

- Gradient Descent Method

$$P = (x_0, y_0, r)$$
$$P_{k+1} = P_k - \alpha \nabla J$$

$$\nabla J = \left(2\sum_{i}^{N} \left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right] \left[-2(x_i - x_0) \right] \right) \hat{x}_0 +$$

$$\left(2\sum_{i}^{N} \left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right] \left[-2(y_i - y_0) \right] \right) \hat{y}_0 +$$

$$\left(2\sum_{i}^{N} \left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right] \left[-2r \right] \right) \hat{r}$$

$$P = (x_0, y_0, r)$$

$$P_{k+1} = P_k - \alpha \nabla J$$

$$x_0 |_{k+1} = x_0 |_k - \alpha \left(2\sum_{i}^{N} \left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right] \left[-2(x_i - x_0) \right] \right)$$

$$y_0 |_{k+1} = y_0 |_k - \alpha \left(2\sum_{i}^{N} \left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right] \left[-2(y_i - y_0) \right] \right)$$

$$r |_{k+1} = r |_k - \alpha \left(2\sum_{i}^{N} \left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2 \right] \left[-2r \right] \right)$$

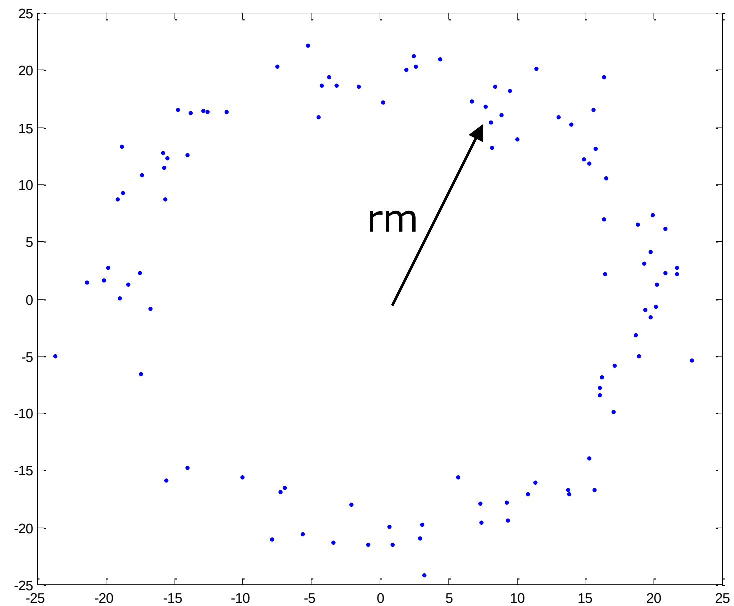
Example of GDM-based Regression

- randn : Normal distribution= x~N(0,1)
- Our distribution with $x' \sim N(\mu, \sigma)$
  - In Matlab code: circle1.m

$$x' \sim N(\mu, \sigma) = \mu + \sigma \cdot randn$$

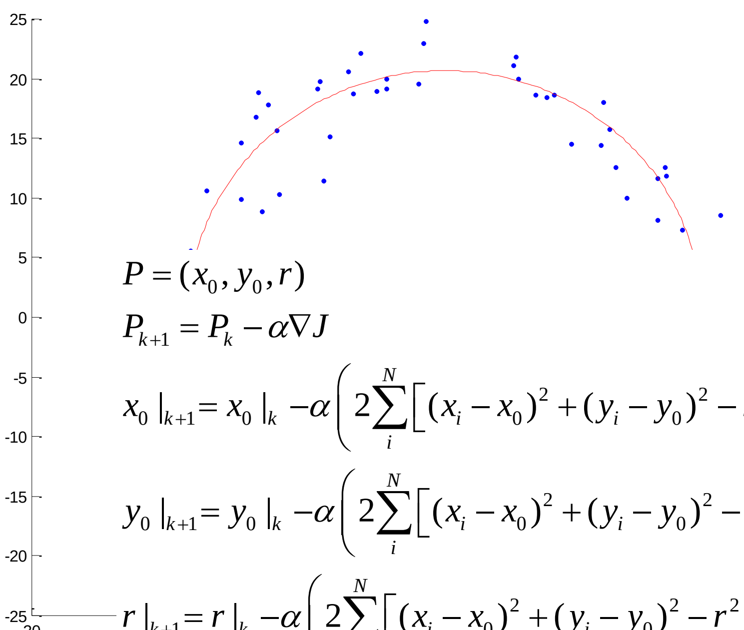
Ex) Radius=20
Sigma =2

```
% generate data
N=100;

rm=20;
rs=2;

qm=0
qs=360;

r= rm+ randn(N,1)*rs;
q= qm+ randn(N,1)*qs;

x=[r,q];

X=[ r.*cos(q.*pi/180), r.*sin(q.*pi/180)]

figure(1)
clf
plot(X(:,1),X(:,2),'.')
```



Example of GDM-based Regression

- GDM for finding x0,y0,r
- (circle2.m)



$$P = (x_0, y_0, r)$$

$$P_{k+1} = P_k - \alpha \nabla J$$

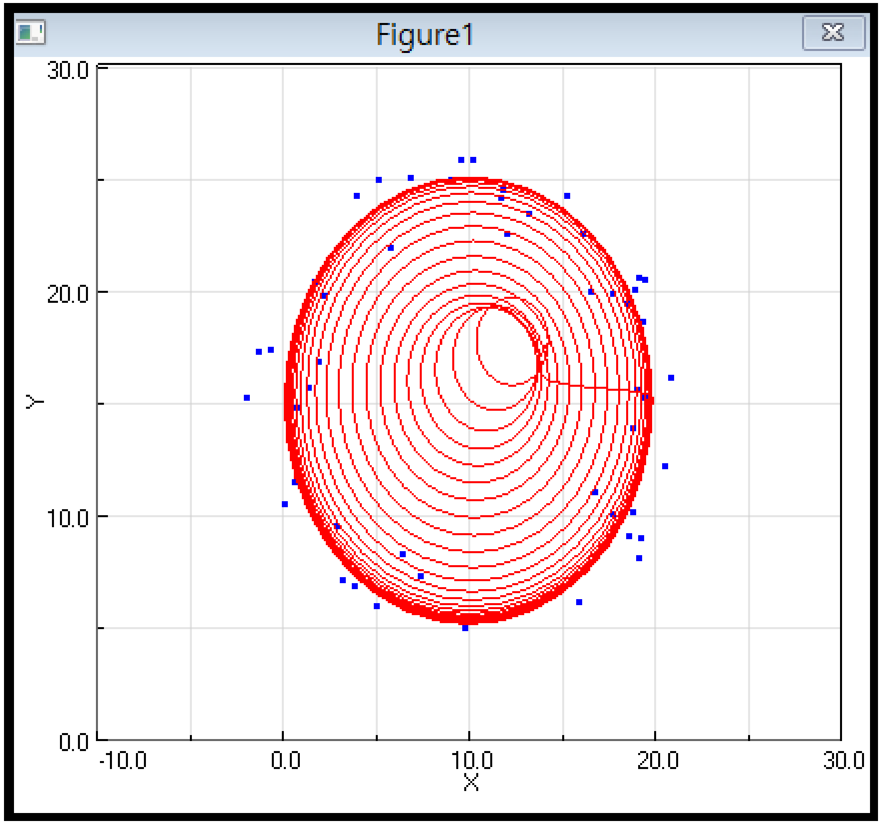
$$x_0|_{k+1} = x_0|_k - \alpha\left(2\sum_i^N\left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2\right]\left[-2(x_i - x_0)\right]\right)$$

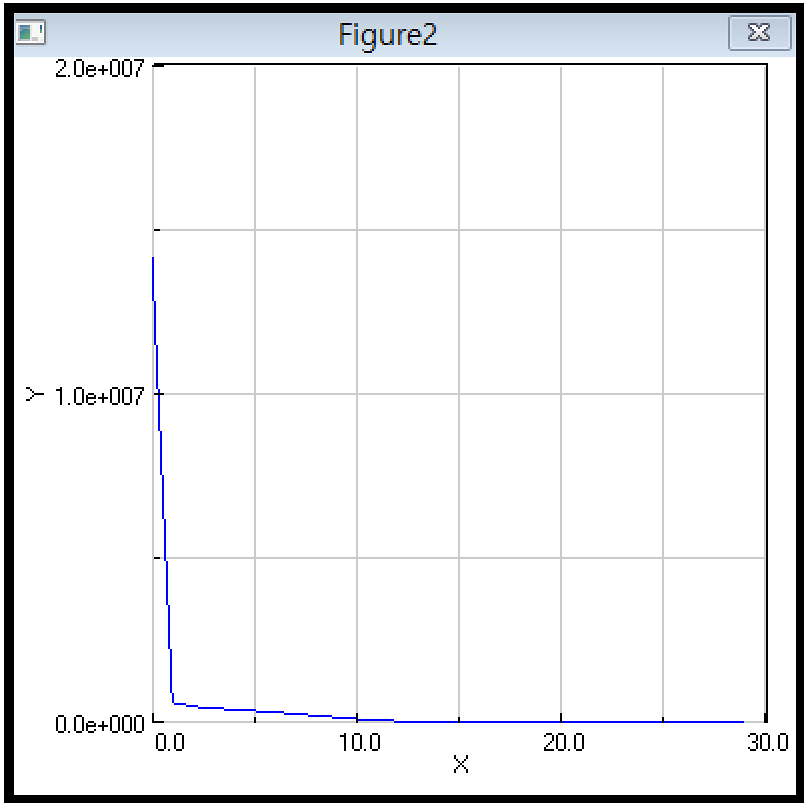
$$y_0|_{k+1} = y_0|_k - \alpha\left(2\sum_i^N\left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2\right]\left[-2(y_i - y_0)\right]\right)$$

$$r|_{k+1} = r|_k - \alpha\left(2\sum_i^N\left[(x_i - x_0)^2 + (y_i - y_0)^2 - r^2\right]\left[-2r\right]\right)$$

```
clear

%generate data
test1

% X:data
% for test, X=X+(3,0)
X(:,1)=X(:,1)+3;

% Exact x=3, y=0, r=20

x=0;
y=0;
r=0.1;
alpha=1e-6;

for i=0:100
    dx = X(:,1)-x;    %(xi-x)
    dy = X(:,2)-y;    %(yi-y)

    sumx=0;
    sumy=0;
    sumr=0;
    for j=1:N
        sumx = sumx+2*(dx(j)^2+dy(j)^2-r*r)*(-2)*dx(j);
        sumy = sumy+2*(dx(j)^2+dy(j)^2-r*r)*(-2)*dy(j);
        sumr = sumr+2*(dx(j)^2+dy(j)^2-r*r)*(-2)*r;
    end

    dJ_dx = sumx;
    dJ_dy = sumy;
    dJ_dr = sumr;

    x=x-alpha*dJ_dx;
    y=y-alpha*dJ_dy;
    r=r-alpha*dJ_dr;

    % calc error
    J = (dx'*dx+dy'*dy-N*r*r)^2;

    if (J<1e-4)
        break;
    end

    [i,J,x,y,r]
end
```

l5regcircle



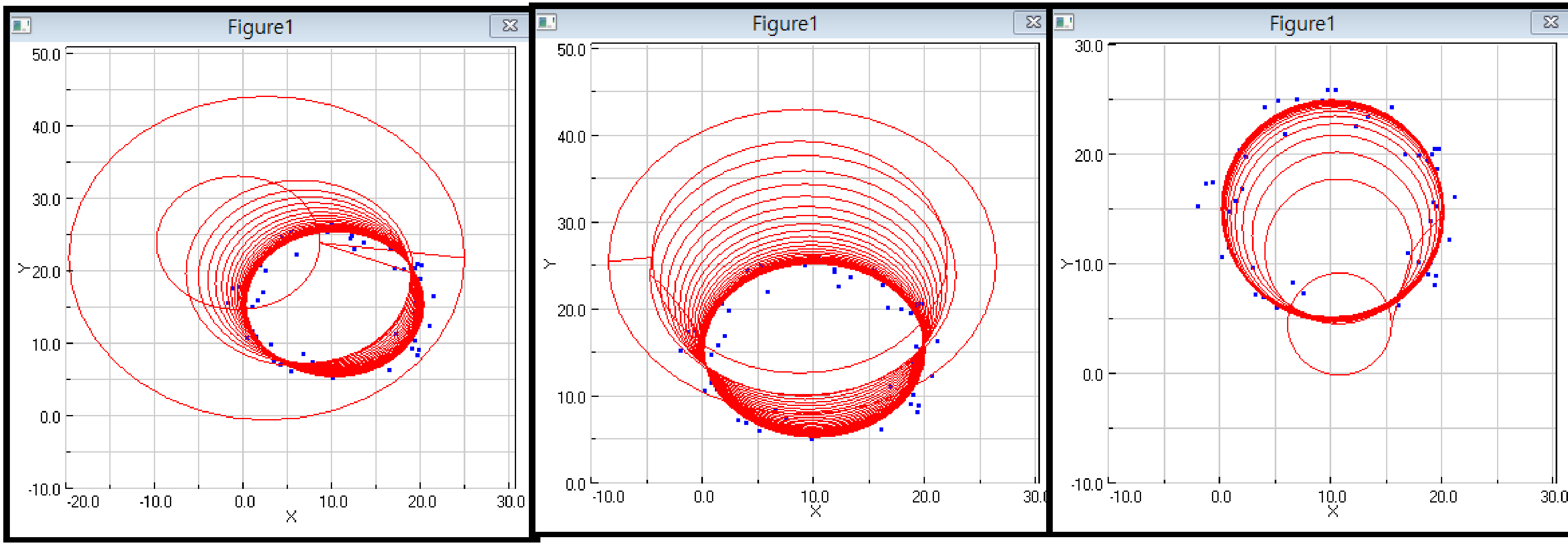


ex/ml/l5regcircle
l5regcircle.test(0,0,1)

Experiment:
Let's try many initial guesses.

Test(0,0,1) test(20,20,50) , etc.

Various Initial Guess

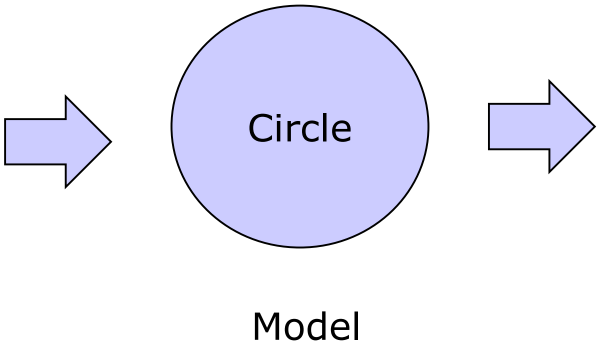


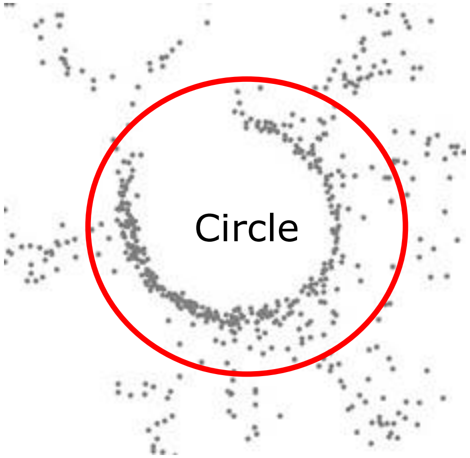
Large error of initial guess makes bad effects?

Try test(0,0,100) or test(100,0,1)

6 Stochastic Regression (RANSAC)

Regression has the Problem like this



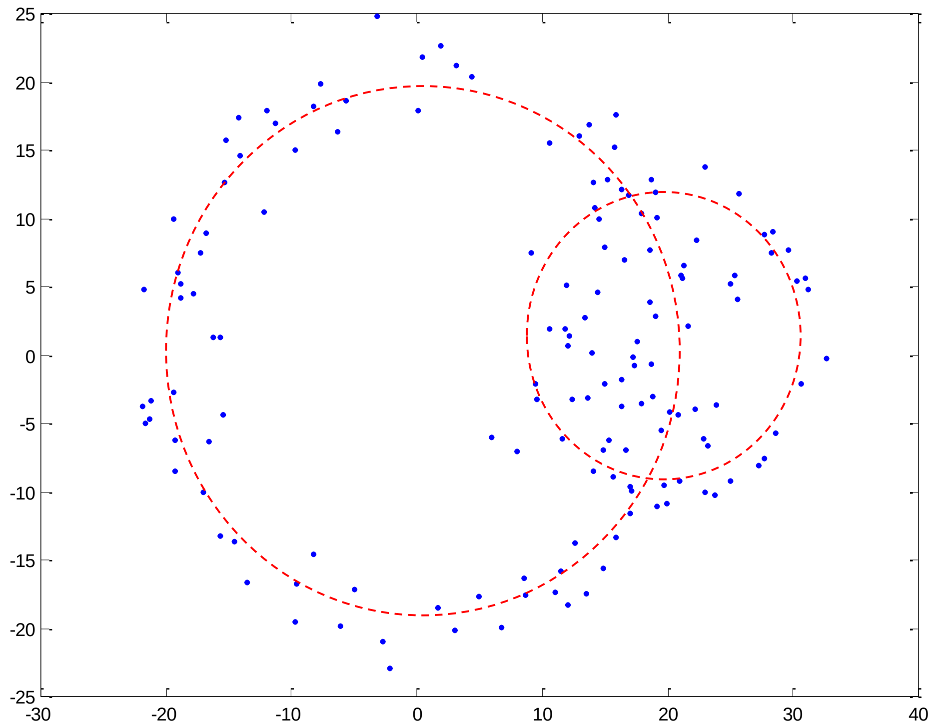


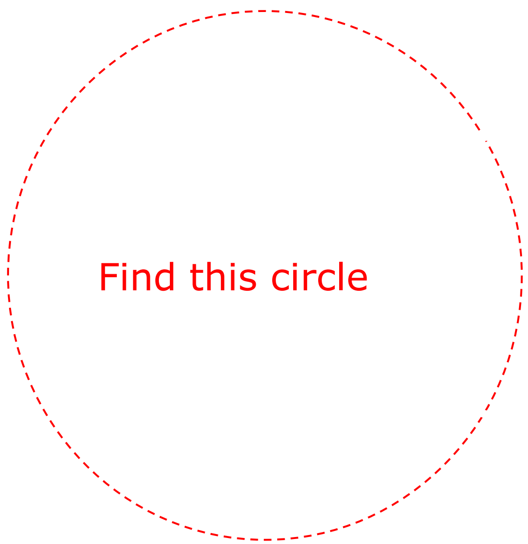
$$J = \sum_{i}^{N} \| (x_i - a)^2 + (y_i - b)^2 - r^2 \|^2$$

- Regression is Model-based method
- **The given DATA is NOT a circle**
- Thus, our model is wrong for data
  - **However, the data is a kind of Circles…**

Generate Non Circle Data

- Circle3.m generate two overlapped circles





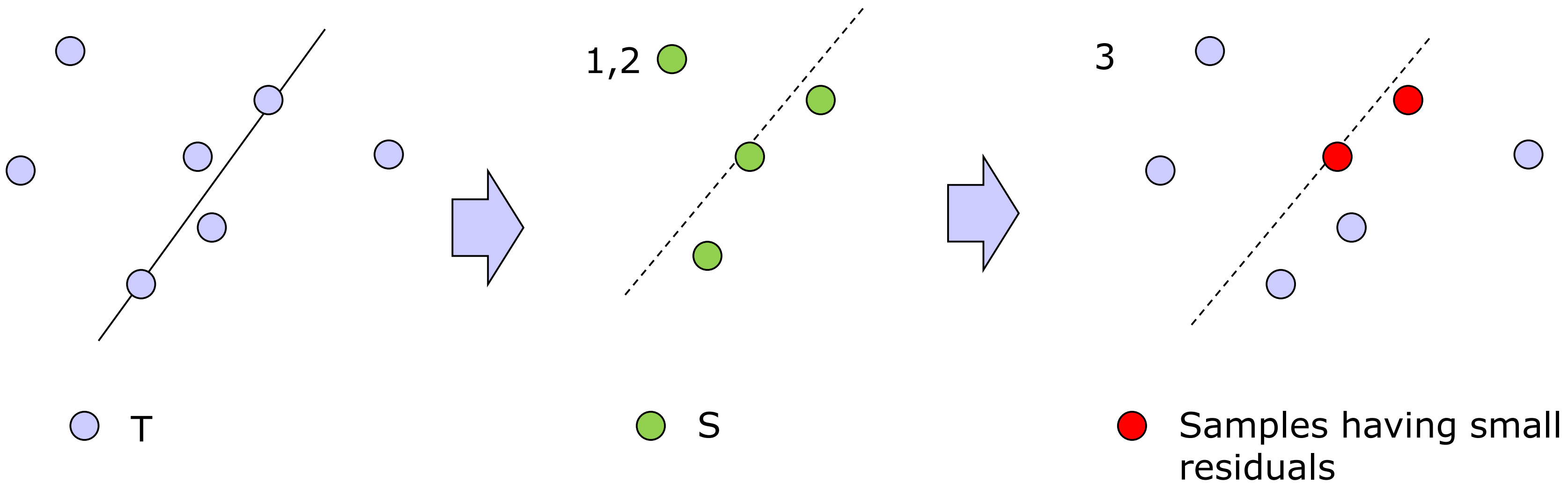
Remove small circle

RANSAC

- RANSAC
  - Random Sample Consensus
  - Pick Good sample from data and Throw away bad sample.

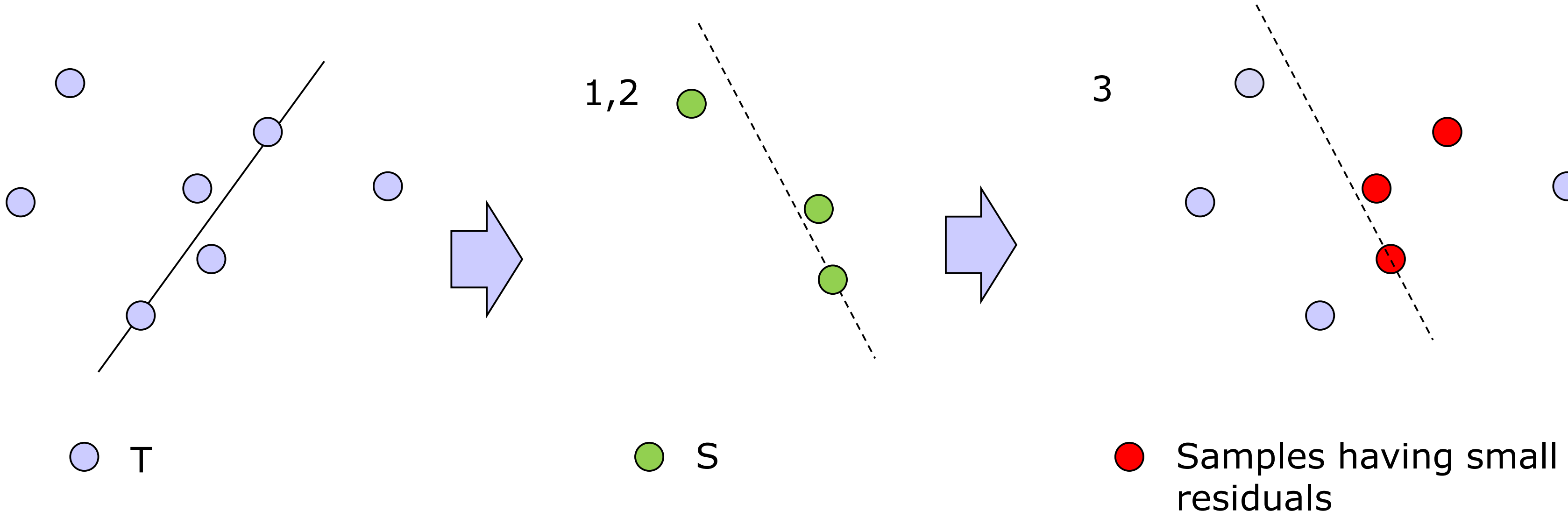
- Training set, T is given
- Pseudo code of RANSAC
  - 1. Take a random Sample, S of size m from T.
  - 2. Build a model, J with S
  - 3. Compute error, e of J with all data T
  - 4. if |e|< threshold, add S or small residuals into consensus set, CS
  - 5. repeat to 1 until CS is larger than some value.

Pseudo Code of RANSAC 1



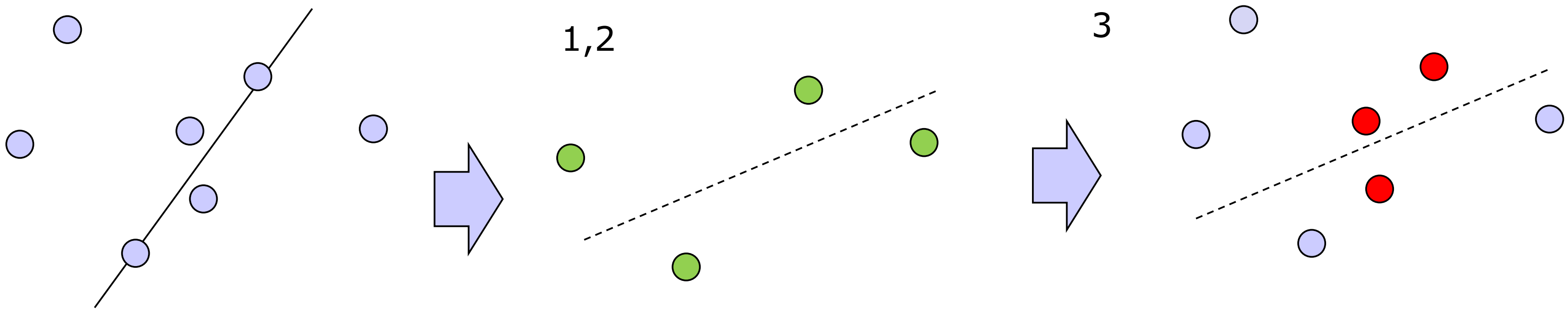
Pseudo code of RANSAC
1. Take a random Sample, S of size m from T.
2. Build a model, J with S
3. Compute error, e of J with all data T
4. if |e|< threshold, add S or small residuals into consensus set, CS
5. repeat to 1 until CS is larger than some value.

Pseudo Code of RANSAC 2



Pseudo code of RANSAC
1. Take a random Sample, S of size m from T.
2. Build a model, J with S
3. Compute error, e of J with all data T
4. if |e|< threshold, add S or small residuals into consensus set, CS
5. repeat to 1 until CS is larger than some value.

Pseudo Code of RANSAC 3



T

S

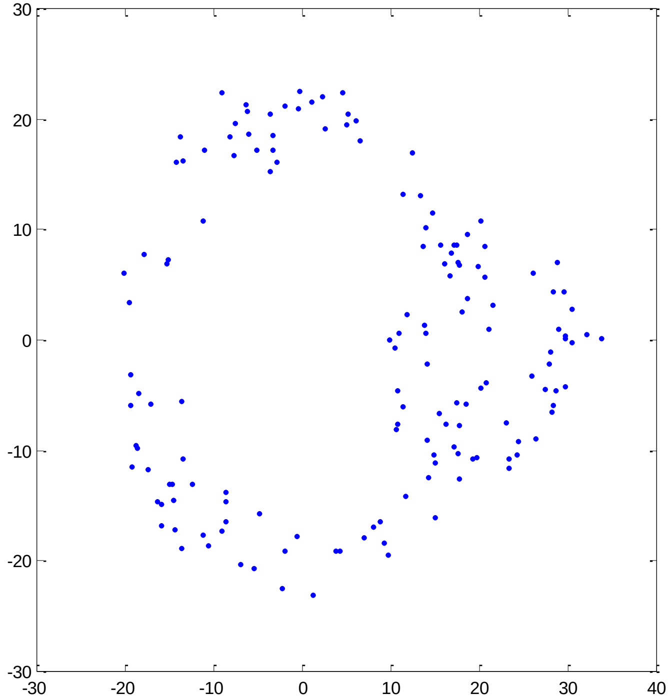
Samples having small residuals

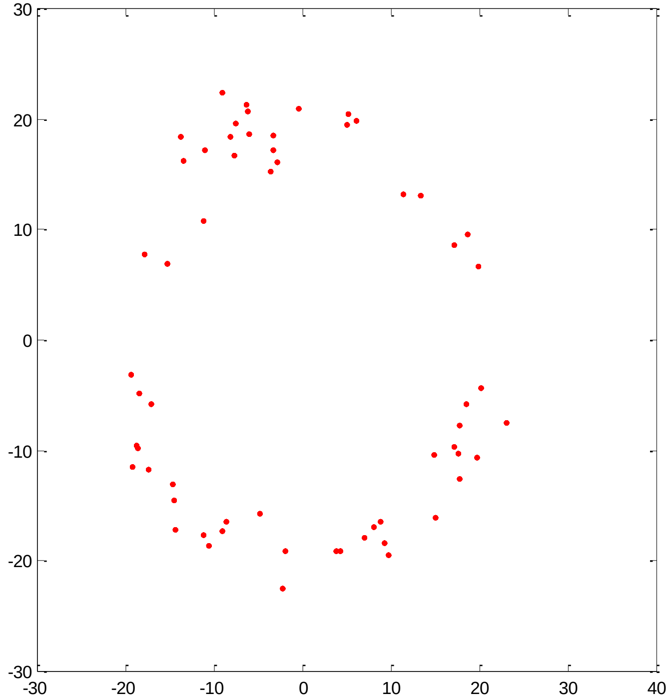
It does NOT satisfy |e|< threshold, then no residual is NOT added into CS

Pseudo code of RANSAC
1. Take a random Sample, S of size m from T.
2. Build a model, J with S
3. Compute error, e of J with all data T
4. if |e|< threshold, add S or small residuals into consensus set, CS
5. repeat to 1 until CS is larger than some value.

Circle3 and Circle4.m

- Circle3: generate Overlapped circle
- Circle4: Simple RANSAC



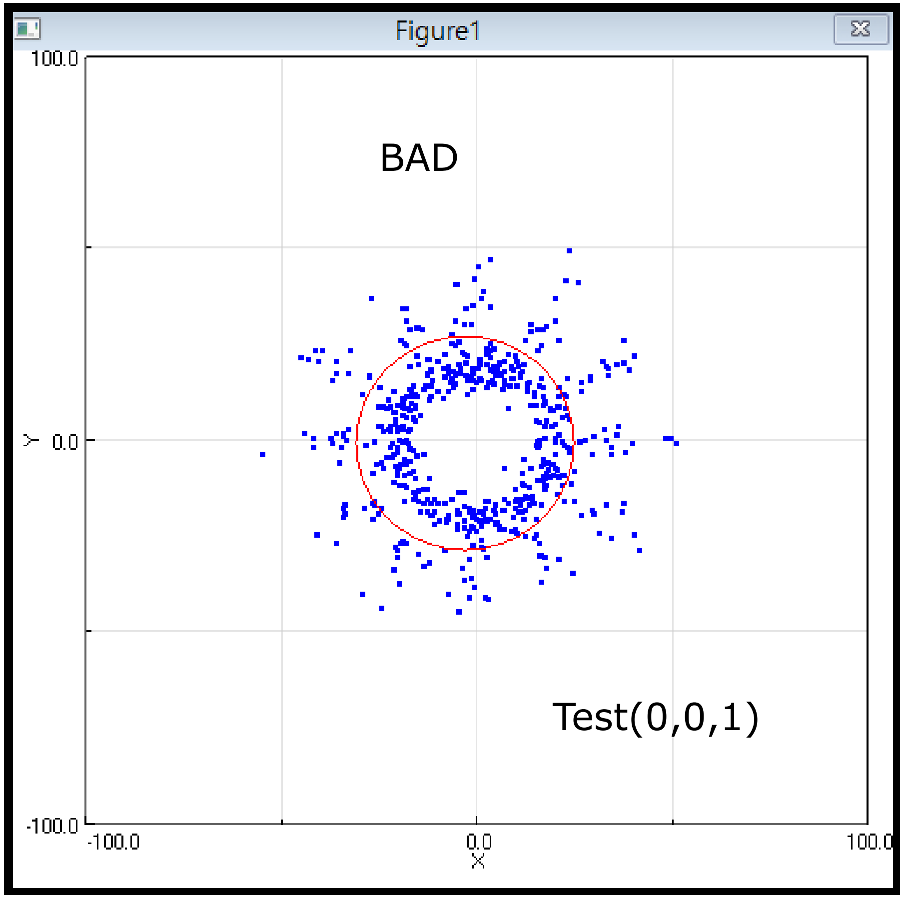


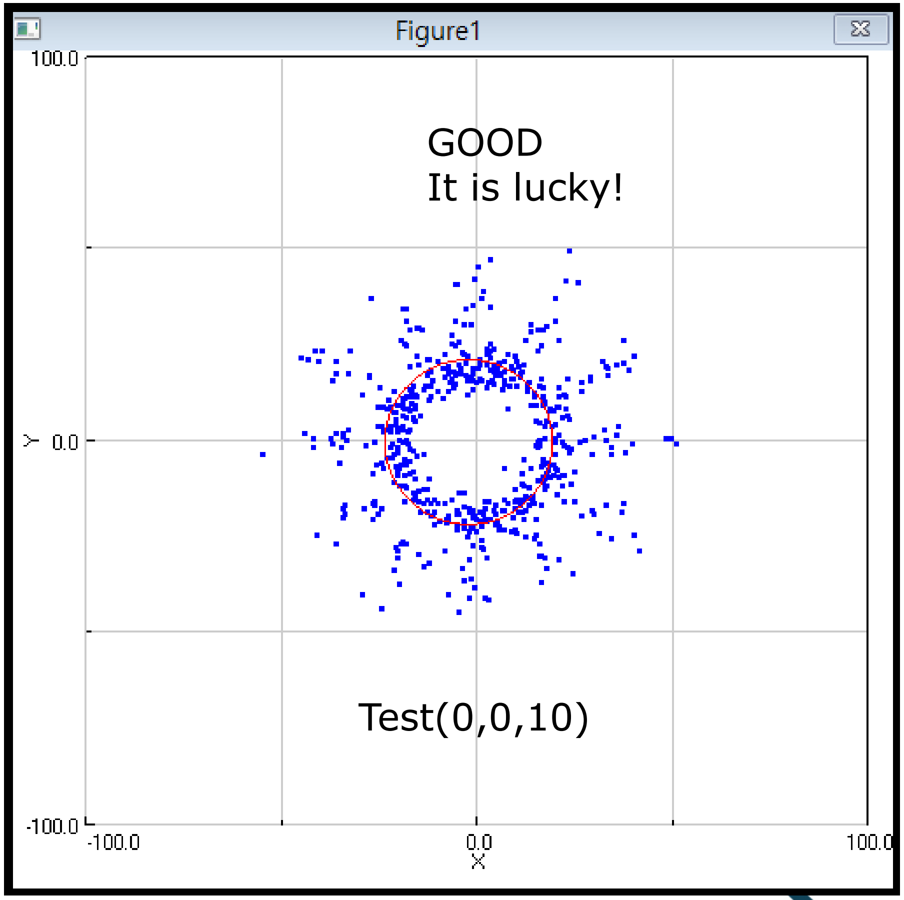
Test4.m

```
clear

%generate data
test3

Ns = 60;   %SAMPLE
Xi = zeros(Ns,1);
Xin = [];

for i=0:30
   % get sample,Xs
   for j=1:Ns
      Xi(j) = ceil(rand*Ns);
   end
   Xs = X(Xi,:);

   %find circle from sample
   [J,x,y,r] = fcircle(Xs);

   if (J<1e-2)
      Xin = [Xin;Xs];
   end

end

figure(2);
clf;
subplot(121)
plot(X(:,1),X(:,2),'.');
axis([-30 40 -30 30]);
subplot(122)
plot(Xin(:,1),Xin(:,2),'r.');
axis([-30 40 -30 30]);
```

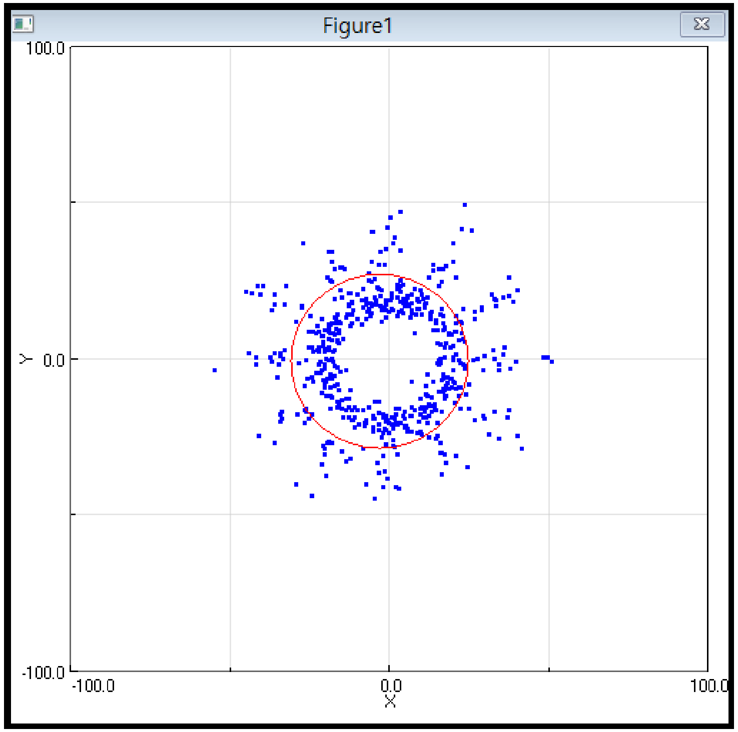
RANSAC:
Why it is so useful?

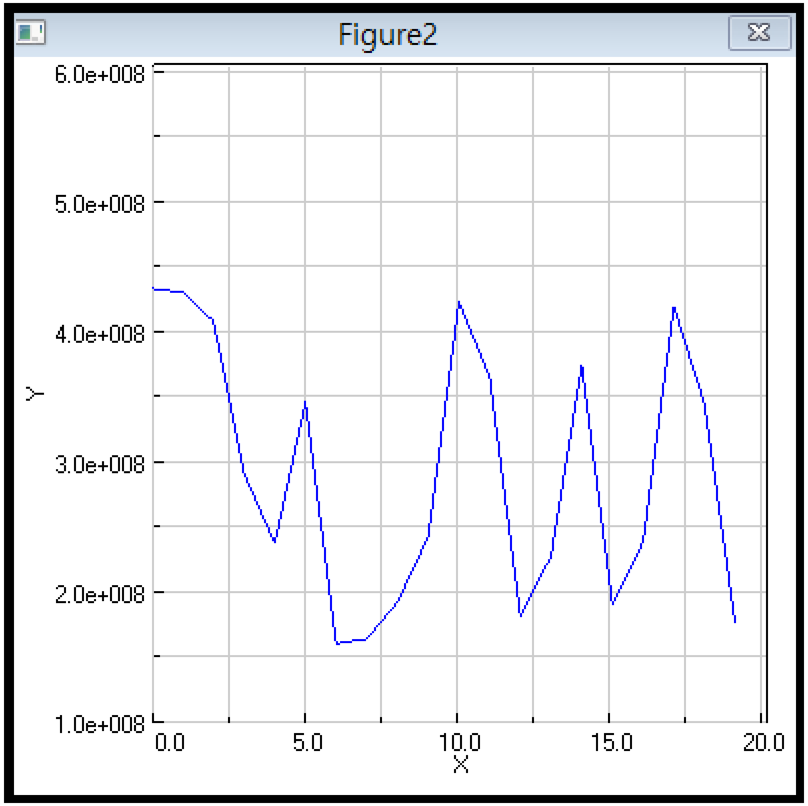
- l5regransac.py DOES NOT use RANSAC





See Iteration Error Graph



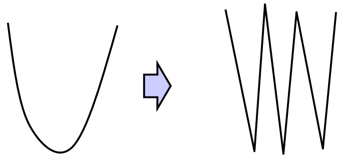


- Is it GOOD? Remind Convex Hulls.

Test with Various Initial guess What kinds of Problems occur?

- 1. Too sensitive to Initial guess.
  - Regression method cannot satisfy circle + noise simultaneously.
  - Thus, It becomes UNSTABLE

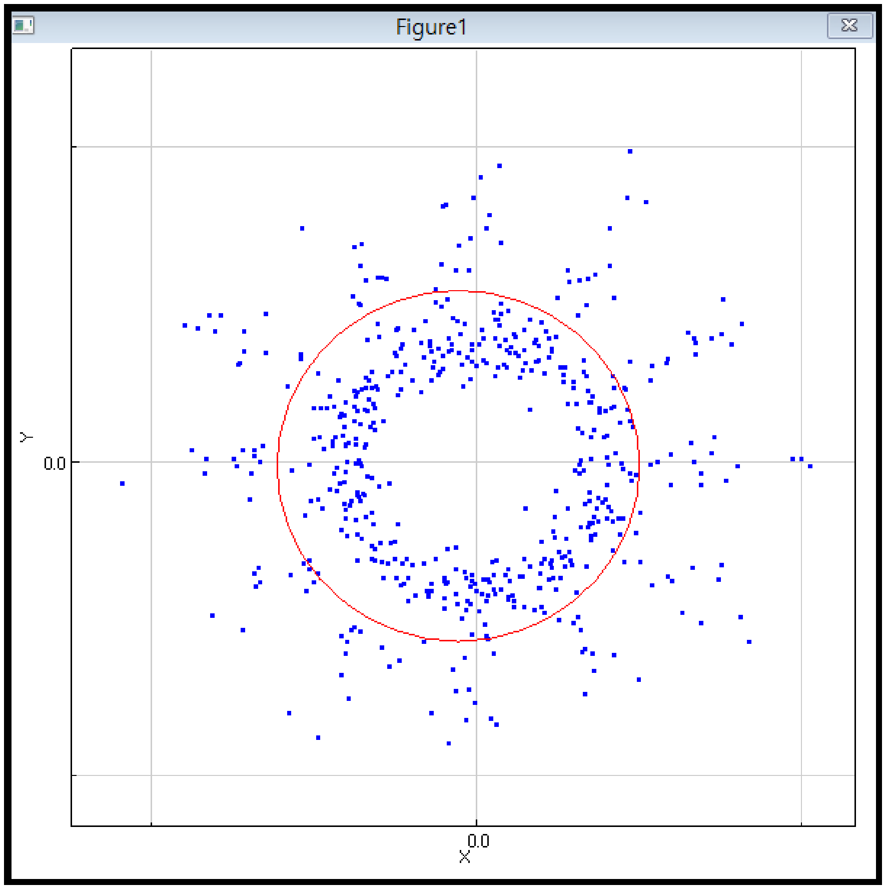
- 2. In other words, Convex hulls are not so strong
  - l2regcircle shows good stability
  - However, l2regransac shows poor performance
  - Convex hull in this example was so sharp
  - Think differentiation leads to the optimal value

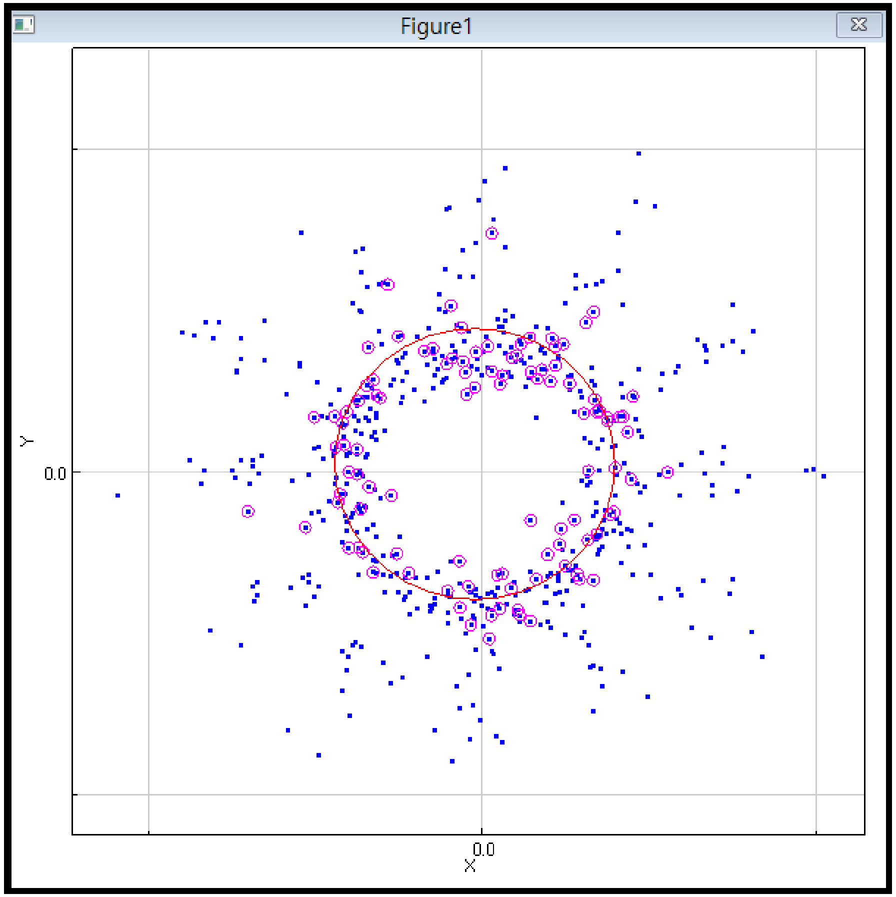


Solvable

Probably Not Solvable. Diff becomes infinity

Comparison with RANSAC





Without RANSAC
l5regransac.test(0,0,1)

With RANSAC
l5regransac2.test()

l5regransac2.py

- edit ex/ml/l5regransac2
- l5regransac2.test()

```
def test():
    load()
    cs = findcs()

    #draw cs
    lg.figure(1)
    lg.graph(2)
    for i in range(0,len(cs)):
        s = cs[i]
        x   = data[s][0]
        y   = data[s][1]
        lg.plot(x,y,'mo')

    # run regression with cs
    if (len(cs)==0):
        print("No CS")
        return;
    xc  = 0;
    yc  = 0;
    r   = 1;
```

Load 2d point data

l5regransac2.data →
[-9.76,16.2], [-6.88,-19.6],….

Get CS by RANSAC

cs =findcs()

cs is the INDEX of data
cs=[51,28,2,20,53,…]

Therefore,
Good sample is
data[cs[0]]
data[cs[1]]
data[cs[2]]
…

findcs : find good sample set= CS

```
def findcs():
    s=[]
    CS=[]
    ndata   = len(data)
    for k in range(0,1000):
        # generate random sample
        sample=[]
        m= 20;
        for i in range(0,m):
            s   = lm.randint(ndata)
            sample.append(s)

        # add sample into CS
        if (reg(sample)==True):
            s   = len(sample)
            n   = len(CS)

            for i in range(0,s):
                bExist = False
                for j in range(0,n):
                    if (CS[j]==sample[i]):
                        bExist = True

                if (bExist==False):
                    CS.append(sample[i])

        # if CS>taget number
        if (len(CS)>100):
            break
    return CS
```

Do sampling 1000 times

Number of sampling,
m=20

Larger m
or
Smaller m
?

1. Data number is 544
2. Sample number is m
3. Chose randint(544)
 0, 1, 100, 200, 45, or 543
4. sample=[120, 3, …]

5. reg() is a regression
Function with sample.

Reg() tries to find a good
solution

HW. RANSAC Performance

- findcs() runs N=1000 times sampling with m=20
- HW. 1 :
  - If we increase or decrease N, what happens?
  - Explain why it occurs

- HW. 2:
  - If we increase m=20, what happens?
  - Explain why it occurs

- HW. 3: reg() function has stop condition,

```
err = J(sample,xc,yc,r)
if (err<4e6):
    return True;
```

if we increase error threshold, 4e6 , what happens?