MAT 3007 – Optimization

Solutions 6

Assignment A6.1 (An Unconstrained Optimization Problem): (approx. *25* points)
Consider the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^2} \; f(x) := x_1^4 + \frac{2}{3}x_1^3 + \frac{1}{2}x_1^2 - 2x_1^2x_2 + \frac{4}{3}x_2^2.$$

a) Calculate all stationary points of the mapping f and investigate whether the stationary points are local maximizer, local minimizer, or saddle points.

b) Create a 3D or contour plot of the function using `MATLAB` or `Python` and decide whether the problem possesses a global solution or not.

Solution :

a) We first calculate the gradient and Hessian of f:

$$\nabla f(x) = \begin{pmatrix} 4x_1^3 + 2x_1^2 + x_1 - 4x_1x_2 \\ -2x_1^2 + \frac{8}{3}x_2 \end{pmatrix}, \quad \nabla^2 f(x) = \begin{pmatrix} 12x_1^2 + 4x_1 + 1 - 4x_2 & -4x_1 \\ -4x_1 & \frac{8}{3} \end{pmatrix}.$$

We have $\nabla f(x) = 0$ if and only if $4x_2 = 3x_1^2$ and $x_1(x_1^2 + 2x_1 + 1) = 0$. This yields the two stationary points $x^* = (0,0)$ and $y^* = (-1, \frac{3}{4})$. It holds that

$$\nabla^2 f(x^*) = \begin{pmatrix} 1 & 0 \\ 0 & \frac{8}{3} \end{pmatrix} \quad \text{and} \quad \nabla^2 f(y^*) = \begin{pmatrix} 6 & 4 \\ 4 & \frac{8}{3} \end{pmatrix}.$$

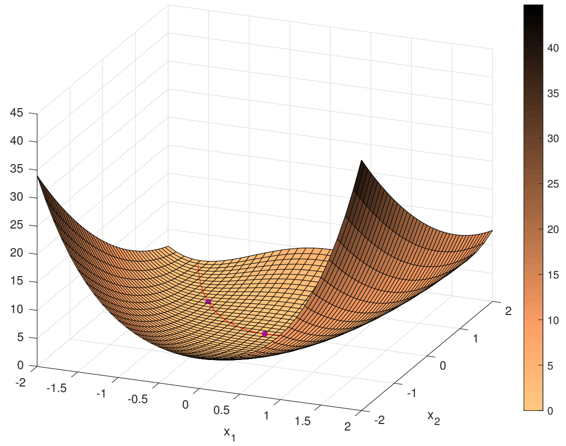
The Hessian $\nabla^2 f(x^*)$ is a diagonal matrix with eigenvalues 1 and $\frac{8}{3}$. Hence, $\nabla^2 f(x^*)$ is positive definite and x^* is a strict local minimizer of f. We further have $\det(\nabla^2 f(y^*)) = 2 \cdot 8 - 16 = 0$ and $\mathrm{tr}(\nabla^2 f(y^*)) = 6 + \frac{8}{3} > 0$. Hence, $\nabla^2 f(y^*)$ is positive semidefinite. In order to decide whether y^* is a saddle point or minimizer, we discuss the objective function around y^* in more detail. It holds that

$$\begin{aligned} f(x) &= x_1^4 + \frac{2}{3}x_1^3 + \frac{1}{2}x_1^2 - 2x_1^2x_2 + \frac{4}{3}x_2^2 = x_1^4 + \frac{2}{3}x_1^3 + \frac{1}{2}x_1^2 - \frac{3}{4}x_1^4 + \left(\frac{\sqrt{3}}{2}x_1^2 - \frac{2}{\sqrt{3}}x_2\right)^2 \\ &= \frac{x_1^2}{2}\left(\frac{1}{2}x_1^2 + \frac{4}{3}x_1 + 1\right) + \left(\frac{\sqrt{3}}{2}x_1^2 - \frac{2}{\sqrt{3}}x_2\right)^2 \end{aligned}$$

and $f(y^*) = \frac{1}{2}(\frac{1}{2} - \frac{4}{3} + 1) + 0 = \frac{1}{12}$. Our idea is to discuss the behavior of $\frac{x_1^2}{2}\left(\frac{1}{2}x_1^2 + \frac{4}{3}x_1 + 1\right)$ in a neighborhood of -1. Let us consider the path $t \mapsto \phi(t) := (t-1, \frac{3}{4}(t-1)^2)$. Then, we have

$$f(\phi(t)) = \frac{(t-1)^2}{2}\left(\frac{1}{2}t^2 - t + \frac{1}{2} + \frac{4}{3}t - \frac{4}{3} + 1\right) = \cdots = \frac{t^4}{4} - \frac{t^3}{3} + \frac{1}{12} = f(y^*) + t^3\left(\frac{t}{4} - \frac{1}{3}\right).$$

Thus, for all $t \in (0, \frac{4}{3})$, we obtain $f(\phi(t)) < f(y^*)$ and for all $t < 0$, it holds that $f(\phi(t)) > f(y^*)$. We conclude that y^* is a saddle point of f.


(a) Surface Plot of f

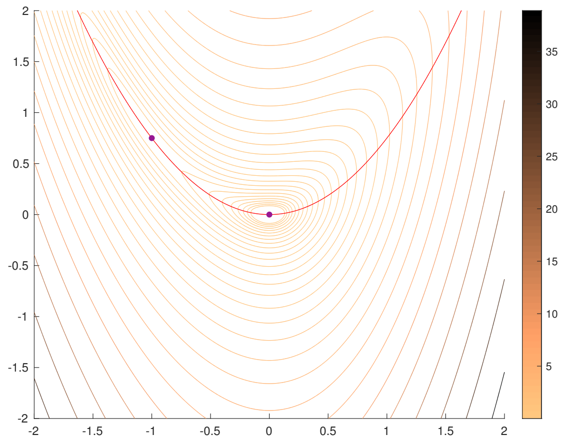

(b) Contour Plot of f

Figure 1: Illustration of f in A6.1

b) A surface and contour plot of f is shown in Figure 1. The red line in the plots corresponds to the function $t \mapsto \phi(t)$. The code generating these plots can be found in Listing 1 & 2.

The plots suggest that f has a global solution and that it is given by x^*.

In fact, we can also prove this mathematically. (This is not required as part of your reply). The function $g(x_1) = \frac{1}{2}x_1^2 + \frac{4}{3}x_1 + 1$ attains its global minimum at $x_1 = -\frac{4}{3}$, i.e., we have $g(x_1) \geq g(-\frac{4}{3}) = \frac{1}{9}$ for all x_1. Reusing the calculations from part a), this shows:

$$f(x) = \frac{x_1^2}{2} g(x_1) + \left(\frac{\sqrt{3}}{2}x_1^2 - \frac{2}{\sqrt{3}}x_2\right)^2 \geq \frac{x_1^2}{18} \geq 0$$

for all $x \in \mathbb{R}^2$. Since $x^* = (0,0)$ is a stationary point with $f(x^*) = 0$, this implies that x^* is a global minimum of f.

Assignment A6.2 (Circle Fitting): (approx. *25* points)

Suppose that the m points $a_1, a_2, \ldots, a_m \in \mathbb{R}^n$ are given. In this exercise, we want to find a circle with center $x \in \mathbb{R}^n$ and radius r that best fits the m points, i.e., we want to determine x and r such that

$$\|x - a_i\| \approx r \quad \forall\, i = 1, \ldots, m.$$

Since these approximate equations can be inconsistent, x and r are recovered as global solutions of the following nonlinear least-squares problem:

$$\min_{x,r}\ f(x,r) := \sum_{i=1}^m (\|x - a_i\|^2 - r^2)^2. \tag{1}$$

a) Consider the related optimization problem

$$\min_{y \in \mathbb{R}^{n+1}}\ g(y) := \sum_{i=1}^m (\|a_i\|^2 - b_i^\top y)^2, \quad b_j := \begin{pmatrix} 2a_j \\ -1 \end{pmatrix}, \quad j = 1, \ldots, m \tag{2}$$

and show/verify the following statements:

– For all $(x, r) \in \mathbb{R}^n \times \mathbb{R}$ it holds that $g((x^\top, \|x\|^2 - r^2)^\top) = f(x, r)$.

- Let $y^* \in \mathbb{R}^{n+1}$ be a global solution of (2) and set $\bar{y} = (y_1^*, \dots, y_n^*)^\top$. Show that we have $y_{n+1}^* \le \|\bar{y}\|^2$.

  Hint: Assume that the result is wrong, i.e., we have $y_{n+1}^* > \|\bar{y}\|^2$. Can you then find a point $z \in \mathbb{R}^{n+1}$ with $g(z) < g(y^*)$?

- Given the global minimizer y^* of (2), can you construct a global solution (x^*, r^*) of the initial problem (1)?

b) Assume that the matrix $B^\top = (b_1, b_2, \dots, b_m) \in \mathbb{R}^{n+1 \times m}$ has full row rank. Show that problem (2) has a unique strict local minimizer and compute it.

c) Write a `MATLAB` or `Python` code to calculate a solution (x^*, r^*) of problem (1) for a given set of points $A = (a_1, a_2, \dots, a_m) \in \mathbb{R}^{n \times m}$. Test your code on the following dataset:

$$a_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 0.25 \\ 0 \end{pmatrix}, \quad a_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad a_4 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad a_5 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad a_6 = \begin{pmatrix} -0.5 \\ -0.5 \end{pmatrix}.$$

Visualize your solution and the points a_1–a_6 using an appropriate plot.

Solution :

a) Using the matrix B from part b), the function g can be written as follows:

$$g(y) = \|c - By\|^2, \quad \text{where} \quad B^\top = (b_1, \dots, b_m) \in \mathbb{R}^{n+1 \times m}, \quad c = (\|a_1\|^2, \dots, \|a_m\|^2)^\top \in \mathbb{R}^m.$$

It holds that

$$g((x^\top, \|x\|^2 - r^2)^\top) = \sum_{i=1}^{m} (\|a_i\|^2 - 2a_i^\top x + \|x\|^2 - r^2)^2 = \sum_{i=1}^{m} (\|x - a_i\|^2 - r^2)^2 = f(x, r).$$

for all $(x, r) \in \mathbb{R}^n \times \mathbb{R}$.

As we have already shown, we have $g(y^*) = \sum_{i=1}^m (\|\bar{y} - a_i\|^2 + y_{n+1}^* - \|\bar{y}\|^2)^2$. If $y_{n+1}^* - \|\bar{y}\|^2 > 0$, we can set $z = (\bar{y}^\top, \|\bar{y}\|^2)^\top$. Then, it follows $y_{n+1}^* - \|\bar{y}\|^2 > 0 = z_{n+1} - \|\bar{y}\|^2$ and $g(y^*) > g(z)$. This is a contradiction to the global optimality of y^*.

We just have shown that the two problem (1) and (2) are equivalent. (Our discussion guarantees that the optimal radius $(r^*)^2 = \|\bar{y}\|^2 - y_{n+1}^* \ge 0$ is nonnegative). Based on this derivation, we can recover a global solution (x^*, r^*) from y^* as follows:

$$x^* = \bar{y} = (y_1^*, ..., y_n^*)^\top, \quad r^* = \sqrt{\|\bar{y}\|^2 - y_{n+1}^*}.$$

Remark: In this exercise, we have shown that the circle fitting problem (1) can be reduced to a (simpler) least squares problem.

b) Using the notation from part b), we have

$$g(y) = (c - By)^\top (c - By) = \|c\|^2 - 2c^\top By + y^\top B^\top By.$$

Hence, g is a quadratic function. Since B has full column rank, the matrix $B^\top B$ is invertible and positive definite. We have $\nabla g(y) = 2B^\top B - 2B^\top c$ and $\nabla^2 g(y) = B^\top B$. Hence, $y^* = (B^\top B)^{-1} B^\top c$ is the unique stationary point of problem (2). In addition, due to the positive definiteness of $\nabla^2 g(y)$, the second order sufficient conditions are satisfied and we can infer that y^* is a unique strict local minimizer.

c) An exemplary `MATLAB` and `Python` code is given in Listing 3 & 4. Figure 2 shows the result for the exemplary dataset. In particular, the solutions are given by $x^* \approx (0.2543, 0.3190)^\top$ and $r^* \approx 0.7868$.

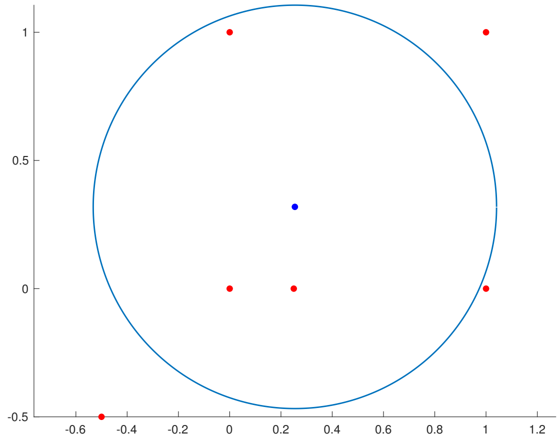


Figure 2: Illustration of the Circle Fitting Problem A6.2c)

Assignment A6.3 (KKT Conditions – I): (approx. *10* points)
Let us consider the optimization problem

$$\begin{aligned} &\min_{x\in\mathbb{R}^3} \quad f(x) := 2x_1^2 + x_1x_2 + x_2^2 + x_2x_3 + x_3^2 - 6x_1 - 7x_2 - 8x_3 - 9 \\ &\text{subject to} \quad x_1 + x_2 + x_3 \le 1, \quad x_1 - x_2^2 = 0. \end{aligned}$$

Write down the KKT conditions for this problem.

Solution : We define $g(x) := x_1 + x_2 + x_3 - 1$ and $h(x) = x_1 - x_2^2$. Then, it holds that

$$\nabla f(x) = \begin{pmatrix} 4x_1 + x_2 - 6 \\ x_1 + 2x_2 + x_3 - 7 \\ x_2 + 2x_3 - 8 \end{pmatrix}, \quad \nabla g(x) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \nabla h(x) = \begin{pmatrix} 1 \\ -2x_2 \\ 0 \end{pmatrix}.$$

Hence, the KKT conditions are given by: find $x \in \mathbb{R}^3$, $\lambda, \mu \in \mathbb{R}$ such that

$$\begin{aligned} 4x_1 + x_2 - 6 + \lambda + \mu &= 0, \\ x_1 + 2x_2 + x_3 - 7 + \lambda - 2x_2\mu &= 0, \\ x_2 + 2x_3 - 8 + \lambda &= 0, \\ \lambda \ge 0, \quad \lambda(x_1 + x_2 + x_3 - 1) &= 0, \quad g(x) \le 0, \quad h(x) = 0. \end{aligned}$$

Assignment A6.4 (KKT Conditions – II): (approx. *20* points)
Consider the problem

$$\min_{x\in\mathbb{R}^3} \; 2x_1x_2 + \frac{1}{2}x_3^2 \quad \text{subject to} \quad 2x_1x_3 + \frac{1}{2}x_2^2 \le 0, \quad 2x_2x_3 + \frac{1}{2}x_1^2 \le 0.$$

a) Write down the KKT conditions for this problem.

b) Investigate whether the point $x^* = (0,0,0)^\top$ is a KKT point satisfying the KKT conditions.

Solution : We define $f(x) := 2x_1x_2 + \frac{1}{2}x_3^2$, $g_1(x) := 2x_1x_3 + \frac{1}{2}x_2^2$ and $g_2(x) = 2x_2x_3 + \frac{1}{2}x_1^2$. Then, it holds that

$$\nabla f(x) = \begin{pmatrix} 2x_2 \\ 2x_1 \\ x_3 \end{pmatrix}, \quad \nabla g_1(x) = \begin{pmatrix} 2x_3 \\ x_2 \\ 2x_1 \end{pmatrix}, \quad \nabla g_2(x) = \begin{pmatrix} x_1 \\ 2x_3 \\ 2x_2 \end{pmatrix}.$$

a) The KKT conditions are given by: find $x \in \mathbb{R}^3$, $\lambda \in \mathbb{R}^2$ such that

$$\begin{aligned}
2x_2 + 2x_3\lambda_1 + x_1\lambda_2 &= 0, \\
2x_1 + x_2\lambda_1 + 2x_3\lambda_2 &= 0, \\
x_3 + 2x_1\lambda_1 + 2x_2\lambda_2 &= 0, \\
\lambda_1(2x_1x_3 + \tfrac{1}{2}x_2^2) &= 0, \\
\lambda_2(2x_2x_3 + \tfrac{1}{2}x_1^2) &= 0, \\
\lambda_1, \lambda_2 \geq 0, \quad g_1(x) \leq 0,& \quad g_2(x) \leq 0.
\end{aligned}$$

b) The point $x^* = (0, 0, 0)^\top$ obviously satisfies the KKT conditions for all $\lambda_1, \lambda_2 \geq 0$.

Assignment A6.5 (Projection Onto a Ball): (approx. *20* points)
Let $m \in \mathbb{R}^n$ and $r > 0$ be given and define the ball $C := \{x \in \mathbb{R}^n : \|x - m\| \leq r\}$. In this exercise, we want to compute the projection $\mathcal{P}_C(x)$ for $x \in \mathbb{R}^n$, i.e., we want solve the optimization problem

$$\min_{y \in \mathbb{R}^n} \frac{1}{2}\|y - x\|^2 \quad \text{subject to} \quad \|y - m\|^2 \leq r^2. \tag{3}$$

a) Write down the KKT conditions for problem (3).

b) Show that the KKT conditions have a unique solution and calculate the corresponding KKT pair explicitly.

Solution :

a) The Lagrangian of problem (3) is given by $L(y, \lambda) = \frac{1}{2}\|y - x\|^2 + \lambda(\|y - m\|^2 - r^2)$. Hence, the KKT conditions are given by:

$$\nabla_y L(y, \lambda) = y - x + 2\lambda(y - m) = 0, \quad \lambda \geq 0, \quad \lambda(\|y - m\|^2 - r^2) = 0, \quad \|y - m\| \leq r.$$

b) We first assume that $\|y - m\| < r$. In this case, the complementarity conditions imply $\lambda = 0$ and using the main condition, we can infer $y = x$. (Thus, this case requires $\|x - m\| < r$). Next, suppose that $\|y - m\| = r$. Rearranging the main condition, it follows

$$y = m + \frac{x - m}{1 + 2\lambda} \quad \text{and} \quad r = \|y - m\| = \frac{\|x - m\|}{1 + 2\lambda} \quad \Longrightarrow \quad 2\lambda = \frac{\|x - m\| - r}{r}.$$

Consequently, we obtain $y = m + r \cdot \frac{x-m}{\|x-m\|}$. In addition, our calculations show $\|x - m\| = (1 + 2\lambda)r \geq r$. Overall, we can summarize: In the case $\|x - m\| < r$, the unique KKT pair (y^*, λ^*) of (3) is given by $(y^*, \lambda^*) = (x, 0)$. In the case $\|x - m\| \geq r$, the unique KKT pair (y^*, λ^*) of (3) is given by $(y^*, \lambda^*) = (m + r\frac{x-m}{\|x-m\|}, \frac{1}{2}(\frac{\|x-m\|}{r} - 1))$.

Listing 1: A6.1: `MATLAB` code: Plotting f

```
% contour plot
x       = -2:0.01:2;
[X,Y]   = meshgrid(x);
Z       = X.^4+(2/3)*X.^3+0.5*X.^2 -2*X.^2.*Y+ (4/3)*(Y).^2;
t       = -2:0.01:3;

figure;
hold on

contour(X,Y,Z,logspace(-2,3,40))

plot3(t-1,0.75*(t-1).^2,1/12+t.^3.*(t/4-1/3),'Color','r');

plot3(0,0,0.01,'.','Color',[152,24,147]/255,'MarkerSize',16);
plot3(-1,3/4,1/12+0.01,'.','Color',[152,24,147]/255,'MarkerSize',16);

colormap(flipud(copper))

axis([-2 2 -2 2])
colorbar;
hold off

set(gcf,'Renderer', 'painters');
saveas(gcf,'contour_a61','epsc');

% surface plot
x       = -2:0.1:2;
[X,Y]   = meshgrid(x);
Z       = X.^4+(2/3)*X.^3+0.5*X.^2 -2*X.^2.*Y+ (4/3)*(Y).^2;

figure;
hold on

surf(X,Y,Z)

plot3(t-1,0.75*(t-1).^2,1/12+t.^3.*(t/4-1/3),'Color','r');

plot3(0,0,0.01,'.','Color',[152,24,147]/255,'MarkerSize',16);
plot3(-1,3/4,1/12+0.01,'.','Color',[152,24,147]/255,'MarkerSize',16);

colormap(flipud(copper))

axis([-2 2 -2 2])
colorbar; xlabel('x_1'); ylabel('x_2');

hold off

view(22,25);
grid on

set(gcf,'Renderer', 'painters');
saveas(gcf,'surf_a61','epsc');
```

Listing 2: A6.1: `Python` code: Plotting f

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d

# contour plot
x = np.arange(-2,2,0.01)
X, Y=np.meshgrid(x,x)
Z = X**4 + (2/3) *(X**3) +0.5*(X**2) - 2 * (X**2) * Y + (4/3) * (Y**2)
t= np.arange(-2,3,0.01)

fig1=plt.figure()
cset=plt.contour(X,Y,Z,np.logspace(-2,3,40))

plt.plot(t-1, 0.75*(t-1)**2,c='black')
plt.scatter(0,0,0.01,c='red',marker='*',linewidths=12)
plt.scatter(-1,3/4,1/12+0.01,c='red',marker='*',linewidths=12)

plt.axis([-2,2,-2,2])
plt.colorbar(cset)
# plt.title('contour plot')
plt.show()


# surface plot
x = np.arange(-2,2,0.1)
X, Y = np.meshgrid(x, x)
Z = X**4 + (2/3) *(X**3) +0.5*(X**2) - 2 * (X**2) * Y + (4/3) * (Y**2)

fig2=plt.figure()
ax= plt.axes(projection='3d')
c=ax.plot_surface(X,Y,Z,cmap='BuPu')
ax.contour(X,Y,Z,zdir = 'z', offset=-2,cmap='BuPu')

ax.plot3D(t-1, 0.75*(t-1)**2, 1/12+t**3.*(t/4-1/3),'black')
ax.scatter3D(0,0,0.01,c='red', marker='*', linewidths=8, cmap='rainbow')
ax.scatter3D(-1,3/4,1/12+0.01, c='red', marker='*', linewidths=8, cmap='rainbow')

plt.xlabel('x_1')
plt.ylabel('x_2')
plt.axis([-2,2,-2,2])
plt.colorbar(c)
# ax.set_title('surface plot')
plt.show()
```

Listing 3: A6.2: MATLAB code: Circle Fitting

```matlab
function circle_fitting

  % inner function
  function [x,r] = cfit_general(A)

  % === INPUT ==========
  % A    a n x m matrix representing the m poins: a_1, a_2, ... a_m

  % We first construct the m x n+1 matrix B with B' = (b_1,...,b_m)
  % and b_i = [2a_i ; -1]
```

```
  [n,m] = size(A);
  B    = [2*A',-ones(m,1)];

  % Construct the vector c = (||a_1||^2, ..., ||a_m||^2)' and the solution y
  c    = sum(A.^2)';
  y    = (B'*B) \ (B'*c);

  % Reconstruct the center x and radius r
  x    = y(1:n);
  r    = sqrt(norm(x)^2 - y(n+1));
  end

A    = [0, 0.25, 1, 1, 0, -0.5; 0, 0, 0, 1, 1, -0.5];
[x,r] = cfit_general(A);

alpha = 0:0.01:2*pi;

hold on
plot(x(1)+r*cos(alpha),x(2)+r*sin(alpha),'LineWidth',1.2);
plot(A(1,:),A(2,:),'r.','MarkerSize',12);
hold off

end
```

Listing 4: A6.2: `Python` code: Circle Fitting

```
import numpy as np
import matplotlib.pyplot as plt

# === INPUT ==========
# A    a n x m matrix representing the m poins: a_1, a_2, ... a_m
A = np.array([[0, 0.25, 1, 1, 0, -0.5], [0, 0, 0, 1, 1, -0.5]])

# We first construct the m x n+1 matrix B with B' = (b_1,...,b_m)
# and b_i = [2a_i ; -1]
n, m = np.shape(A)
B = np.append(2 * A.T, -np.ones([m, 1]), axis=1)

# Construct the vector c = (||a_1||^2, ..., ||a_m||^2)' and the solution y
c = np.sum(A**2, axis=0).T
y = np.dot(np.linalg.inv(np.dot(B.T, B)), np.dot(B.T, c))

# Reconstruct the center x and radius r
x = y[0:n]
r = np.sqrt(np.dot(x, x.T) - y[n])

# Visualization
alpha = np.arange(0,2*np.pi, 0.02)
plt.plot(x[0]+r*np.cos(alpha), x[1]+r*np.sin(alpha))
plt.scatter(A[0, :], A[1, :], c='red')
plt.show()
```