# CVX101 Homework 6 solutions

A7.5 *Three-way linear classification.* We are given data

$$x^{(1)}, \ldots, x^{(N)}, \quad y^{(1)}, \ldots, y^{(M)}, \quad z^{(1)}, \ldots, z^{(P)},$$

three nonempty sets of vectors in $\mathbf{R}^n$. We wish to find three affine functions on $\mathbf{R}^n$,

$$f_i(z) = a_i^T z - b_i, \quad i = 1, 2, 3,$$

that satisfy the following properties:

$$\begin{array}{ll} f_1(x^{(j)}) > \max\{f_2(x^{(j)}), f_3(x^{(j)})\}, & j = 1, \ldots, N, \\ f_2(y^{(j)}) > \max\{f_1(y^{(j)}), f_3(y^{(j)})\}, & j = 1, \ldots, M, \\ f_3(z^{(j)}) > \max\{f_1(z^{(j)}), f_2(z^{(j)})\}, & j = 1, \ldots, P. \end{array}$$

In words: $f_1$ is the largest of the three functions on the $x$ data points, $f_2$ is the largest of the three functions on the $y$ data points, $f_3$ is the largest of the three functions on the $z$ data points. We can give a simple geometric interpretation: The functions $f_1$, $f_2$, and $f_3$ partition $\mathbf{R}^n$ into three regions,

$$\begin{array}{rcl} R_1 & = & \{z \mid f_1(z) > \max\{f_2(z), f_3(z)\}\}, \\ R_2 & = & \{z \mid f_2(z) > \max\{f_1(z), f_3(z)\}\}, \\ R_3 & = & \{z \mid f_3(z) > \max\{f_1(z), f_2(z)\}\}, \end{array}$$

defined by where each function is the largest of the three. Our goal is to find functions with $x^{(j)} \in R_1$, $y^{(j)} \in R_2$, and $z^{(j)} \in R_3$.

Pose this as a convex optimization problem. You may not use strict inequalities in your formulation.

Solve the specific instance of the 3-way separation problem given in `sep3way_data.m`, with the columns of the matrices `X`, `Y` and `Z` giving the $x^{(j)}$, $j = 1, \ldots, N$, $y^{(j)}$, $j = 1, \ldots, M$ and $z^{(j)}$, $j = 1, \ldots, P$. To save you the trouble of plotting data points and separation boundaries, we have included the plotting code in `sep3way_data.m`. (Note that `a1`, `a2`, `a3`, `b1` and `b2` contain arbitrary numbers; you should compute the correct values using CVX.)

**Solution.** The inequalities

$$\begin{array}{ll} f_1(x^{(j)}) > \max\{f_2(x^{(j)}), f_3(x^{(j)})\}, & j = 1, \ldots, N, \\ f_2(y^{(j)}) > \max\{f_1(y^{(j)}), f_3(y^{(j)})\}, & j = 1, \ldots, M, \\ f_3(z^{(j)}) > \max\{f_1(z^{(j)}), f_2(z^{(j)})\}, & j = 1, \ldots, P. \end{array}$$

are homogeneous in $a_i$ and $b_i$ so we can express them as

$$\begin{array}{ll} f_1(x^{(j)}) \geq \max\{f_2(x^{(j)}), f_3(x^{(j)})\} + 1, & j = 1, \ldots, N, \\ f_2(y^{(j)}) \geq \max\{f_1(y^{(j)}), f_3(y^{(j)})\} + 1, & j = 1, \ldots, M, \\ f_3(z^{(j)}) \geq \max\{f_1(z^{(j)}), f_2(z^{(j)})\} + 1, & j = 1, \ldots, P. \end{array}$$

Note that we can add any vector $\alpha$ to each of the $a_i$, without affecting these inequalities (which only refer to difference between $a_i$'s), and we can add any number $\beta$ to each of the $b_i$'s for the same reason. We can use this observation to normalize or simplify the $a_i$ and $b_i$. For example, we can assume without loss of generality that $a_1 + a_2 + a_3 = 0$ and $b_1 + b_2 + b_3 = 0$.

The following script implements this method for 3-way classification and tests it on a small separable data set

```
clear all; close all;
% data for problem instance
M = 20;
N = 20;
P = 20;

X = [

    3.5674    4.1253    2.8535    5.1892    4.3273    3.8133    3.4117 ...
    3.8636    5.0668    3.9044    4.2944    4.7143    3.3082    5.2540 ...
    2.5590    3.6001    4.8156    5.2902    5.1908    3.9802 ;...
   -2.9981    0.5178    2.1436   -0.0677    0.3144    1.3064    3.9297 ...
    0.2051    0.1067   -1.4982   -2.4051    2.9224    1.5444   -2.8687 ...
    1.0281    1.2420    1.2814    1.2035   -2.1644   -0.2821];

Y = [
   -4.5665   -3.6904   -3.2881   -1.6491   -5.4731   -3.6170   -1.1876 ...
   -1.0539   -1.3915   -2.0312   -1.9999   -0.2480   -1.3149   -0.8305 ...
   -1.9355   -1.0898   -2.6040   -4.3602   -1.8105    0.3096; ...
    2.4117    4.2642    2.8460    0.5250    1.9053    2.9831    4.7079 ...
    0.9702    0.3854    1.9228    1.4914   -0.9984    3.4330    2.9246 ...
    3.0833    1.5910    1.5266    1.6256    2.5037    1.4384];

Z = [
    1.7451    2.6345    0.5937   -2.8217    3.0304    1.0917   -1.7793 ...
    1.2422    2.1873   -2.3008   -3.3258    2.7617    0.9166    0.0601 ...
   -2.6520   -3.3205    4.1229   -3.4085   -3.1594   -0.7311; ...
   -3.2010   -4.9921   -3.7621   -4.7420   -4.1315   -3.9120   -4.5596 ...
   -4.9499   -3.4310   -4.2656   -6.2023   -4.5186   -3.7659   -5.0039 ...
```

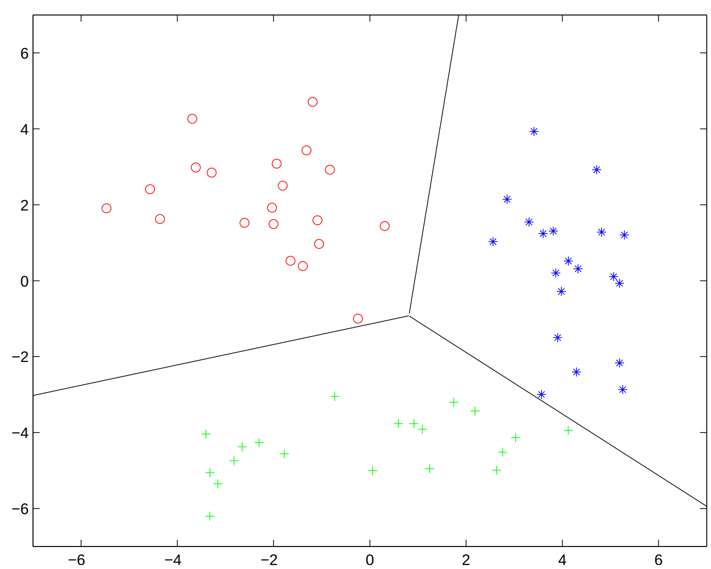
```
   -4.3744   -5.0559   -3.9443   -4.0412   -5.3493   -3.0465];

cvx_begin
variables a1(2) a2(2) a3(2) b1 b2 b3
    a1'*X-b1 >= max(a2'*X-b2,a3'*X-b3)+1;
    a2'*Y-b2 >= max(a1'*Y-b1,a3'*Y-b3)+1;
    a3'*Z-b3 >= max(a1'*Z-b1,a2'*Z-b2)+1;
    a1 + a2 + a3 == 0
    b1 + b2 + b3 == 0
cvx_end

% now let's plot the three-way separation induced by
% a1,a2,a3,b1,b2,b3
% find maximally confusing point
p = [(a1-a2)';(a1-a3)']\[(b1-b2);(b1-b3)];

% plot
t = [-7:0.01:7];
u1 = a1-a2; u2 = a2-a3; u3 = a3-a1;
v1 = b1-b2; v2 = b2-b3; v3 = b3-b1;
line1 = (-t*u1(1)+v1)/u1(2); idx1 = find(u2'*[t;line1]-v2>0);
line2 = (-t*u2(1)+v2)/u2(2); idx2 = find(u3'*[t;line2]-v3>0);
line3 = (-t*u3(1)+v3)/u3(2); idx3 = find(u1'*[t;line3]-v1>0);
plot(X(1,:),X(2,:),'*',Y(1,:),Y(2,:),'ro',Z(1,:),Z(2,:),'g+',...
     t(idx1),line1(idx1),'k',t(idx2),line2(idx2),'k',t(idx3),line3(idx3),'k');
axis([-7 7 -7 7]);
```

The following figure is generated.



A7.16 *Fitting a sphere to data.* Consider the problem of fitting a sphere $\{x \in \mathbf{R}^n \mid \|x - x_c\|_2 = r\}$ to $m$ points $u_1, \ldots, u_m \in \mathbf{R}^n$, by minimizing the error function

$$\sum_{i=1}^{m} \left(\|u_i - x_c\|_2^2 - r^2\right)^2$$

over the variables $x_c \in \mathbf{R}^n$, $r \in \mathbf{R}$.

(a) Explain how to solve this problem using convex or quasiconvex optimization. The simpler your formulation, the better. (For example: a convex formulation is simpler than a quasiconvex formulation; an LP is simpler than an SOCP, which is simpler than an SDP.) Be sure to explain what your variables are, and how your formulation minimizes the error function above.

(b) Use your method to solve the problem instance with data given in the file `sphere_fit_data.m`, with $n = 2$. Plot the fitted circle and the data points.

**Solution.**

i. The problem can be formulated as a simple *least-squares problem*, the simplest nontrivial convex optimization problem!
We will formulate the problem as

$$\text{minimize} \quad \|Ax - b\|_2^2.$$

Choose as variables $x = (x_c, t)$ with $t$ defined as $t = r^2 - \|x_c\|_2^2$. Use the optimality conditions $A^T(Ax - b) = 0$ of the least-squares problem to show that $t + \|x_c\|_2^2 \geq 0$ at the optimum. This ensures that $r$ can be computed from the optimal $x_c$, $t$ using the formula $r = (t + \|x_c\|_2^2)^{1/2}$.
Take

$$A = \begin{bmatrix} 2u_1^T & 1 \\ 2u_2^T & 1 \\ \vdots & \vdots \\ 2u_m^T & 1 \end{bmatrix}, \qquad x = \begin{bmatrix} x_c \\ t \end{bmatrix}, \qquad b = \begin{bmatrix} \|u_1\|_2^2 \\ \|u_2\|_2^2 \\ \vdots \\ \|u_m\|_2^2 \end{bmatrix}.$$

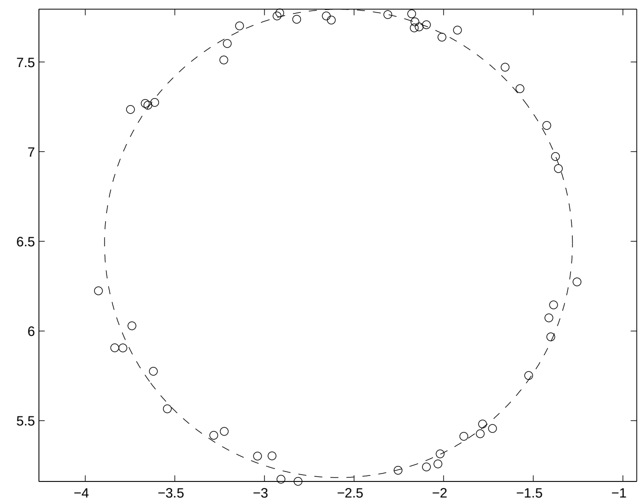
The last equation in $A^T(Ax - b) = 0$ gives

$$\sum_{i=1}^{m} \left(2u_i^T x_c + t - \|u_i\|_2^2\right) = 0,$$

from which we obtain

$$t + \|x_c\|_2^2 = \frac{1}{m} \sum_{i=1}^{m} \|u_i - x_c\|_2^2.$$

ii. $x_c = (-2.5869, 6.4883)$, $R = 1.3052$.




A5.15 *Learning a quadratic pseudo-metric from distance measurements.* We are given a set of $N$ pairs of points in $\mathbf{R}^n$, $x_1, \dots, x_N$, and $y_1, \dots, y_N$, together with a set of distances $d_1, \dots, d_N > 0$.

The goal is to find (or estimate or learn) a quadratic pseudo-metric $d$,

$$d(x,y) = \left((x-y)^T P (x-y)\right)^{1/2},$$

with $P \in \mathbf{S}^n_+$, which approximates the given distances, *i.e.*, $d(x_i, y_i) \approx d_i$. (The pseudo-metric $d$ is a metric only when $P \succ 0$; when $P \succeq 0$ is singular, it is a pseudo-metric.)

To do this, we will choose $P \in \mathbf{S}^n_+$ that minimizes the mean squared error objective

$$\frac{1}{N}\sum_{i=1}^{N}(d_i - d(x_i, y_i))^2.$$

(a) Explain how to find $P$ using convex or quasiconvex optimization. If you cannot find an exact formulation (*i.e.*, one that is guaranteed to minimize the total squared error objective), give a formulation that approximately minimizes the given objective, subject to the constraints.

(b) Carry out the method of part (a) with the data given in `quad_metric_data.m`. The columns of the matrices `X` and `Y` are the points $x_i$ and $y_i$; the row vector `d` gives the distances $d_i$. Give the optimal mean squared distance error.

We also provide a test set, with data `X_test`, `Y_test`, and `d_test`. Report the mean squared distance error on the test set (using the metric found using the data set above).

**Solution.**

(a) The problem is

$$\text{minimize} \quad \tfrac{1}{N}\textstyle\sum_{i=1}^{N}(d_i - d(x_i, y_i))^2$$

with variable $P \in \mathbf{S}^n_+$. This problem can be rewritten as

$$\text{minimize} \quad \frac{1}{N}\sum_{i=1}^N (d_i^2 - 2d_i d(x_i, y_i) + d(x_i, y_i)^2),$$

with variable $P$ (which enters through $d(x_i, y_i)$). The objective is convex because each term of the objective can be written as (ignoring the $1/N$ factor)

$$d_i^2 - 2d_i \left((x_i - y_i)^T P (x_i - y_i)\right)^{1/2} + (x_i - y_i)^T P (x_i - y_i),$$

which is convex in $P$. To see this, note that the first term is constant and the third term is linear in $P$. The middle term is convex because it is the negation of the composition of a concave function (square root) with a linear function of $P$.

(b) The following code solves the problem for the given instance. We find that the optimal mean squared error on the training set is 0.887; on the test set, it is 0.827. This tells us that we probably haven't overfit. In fact, the optimal $P$ is singular; it has one zero eigenvalue. This is correct; the positive semidefinite constraint is active.

```
%% learning a quadratic metric

quad_metric_data;

Z = X-Y;
cvx_begin
    variable P(n,n) symmetric
    % objective
    f = 0;
    for i = 1:N
        f = f + d(i)^2 -2*d(i)*sqrt(Z(:,i)'*P*Z(:,i)) + Z(:,i)'*P*Z(:,i);
    end
    minimize (f/N)
    subject to
        P == semidefinite(n);
cvx_end

Z_test = X_test-Y_test;
d_hat   = norms(sqrtm(P)*Z_test);
obj_test = sum_square(d_test - d_hat)/N_test
```

8.16 *Maximum volume rectangle inside a polyhedron.* Formulate the following problem as a convex optimization problem. Find the rectangle

$$\mathcal{R} = \{x \in \mathbf{R}^n \mid l \preceq x \preceq u\}$$

of maximum volume, enclosed in a polyhedron $\mathcal{P} = \{x \mid Ax \preceq b\}$. The variables are $l, u \in \mathbf{R}^n$. Your formulation should not involve an exponential number of constraints.

**Solution.** A straightforward, but very inefficient, way to express the constraint $\mathcal{R} \subseteq \mathcal{P}$ is to use the set of $m2^n$ inequalities $Av^i \preceq b$, where $v^i$ are the ($2^n$) corners of $\mathcal{R}$. (If the corners of a box lie inside a polyhedron, then the box does.) Fortunately it is possible to express the constraint in a far more efficient way. Define

$$a_{ij}^+ = \max\{a_{ij}, 0\}, \qquad a_{ij}^- = \max\{-a_{ij}, 0\}.$$

Then we have $\mathcal{R} \subseteq \mathcal{P}$ if and only if

$$\sum_{i=1}^n (a_{ij}^+ u_j - a_{ij}^- l_j) \leq b_i, \quad i = 1, \ldots, m,$$

The maximum volume rectangle is the solution of

$$\begin{array}{ll} \text{maximize} & \left(\prod_{i=1}^n (u_i - l_i)\right)^{1/n} \\ \text{subject to} & \sum_{i=1}^n (a_{ij}^+ u_j - a_{ij}^- l_j) \leq b_i, \quad i = 1, \ldots, m, \end{array}$$

with implicit constraint $u \succeq l$. Another formulation can be found by taking the log of the objective, which yields

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^n \log(u_i - l_i) \\ \text{subject to} & \sum_{i=1}^n (a_{ij}^+ u_j - a_{ij}^- l_j) \leq b_i, \quad i = 1, \ldots, m. \end{array}$$