1) Consider task of optimally clustering the following 12 vectors in $\mathbb{R}^2$

X												
	0.8	0.9	0.1	0.9	0.6	0.1	0.3	0.5	1.0	1.0	0.2	1.0
	1.0	0.5	0.8	0.1	0.4	0.9	0.8	1.0	0.7	0.0	0.8	0.9

into four distinct clusters under the L_1 distance. Write a `Matlab` code that heuristically solves the clustering problem using the *k-median* algorithm and succinctly describe it (the fundamental steps, leaving aside the unnecessary details). Then run it on the data above; with the 12 vectors numbered 1, 2, ..., 12 from left to right, use the clusters $\{1, 2, 3\}$, $\{4, 5, 6\}$, $\{7, 8, 9\}$, $\{10, 11, 12\}$ as the initial solution. Comment on the behaviour of the algorithm and report the obtained solution (the centroids and the clusters) and its cost. Optionally, discuss how you could determine if the obtained solution is optimal, and possibly do so.

SOLUTION

The clustering problem in the L_1 norm requires finding four vectors $c_p \in \mathbb{R}^2$, $p = 1, \ldots, 4$ that solve

$$\min\{ f(c) = \textstyle\sum_{1,\ldots,12} \min_{1,\ldots,4} \| c_p - X_i \|_1 \, : \, c \in \mathbb{R}^{4\times 2} \}$$

or equivalently, with the introduction of binary variables,

$$\begin{array}{lll} \min & \textstyle\sum_{i=1,\ldots,12} \sum_{p=1,\ldots,4} z_{ip} \| c_p - X_i \|_1 & \\ & \textstyle\sum_{p=1,\ldots,4} z_{ip} = 1 & i = 1, \ldots, 12 \\ & z_{ip} \{0, 1\} & p = 1, \ldots, 4 \, , \; i = 1, \ldots, 12 \end{array}$$

A way to implement it is to define, together with $X \in \mathbb{R}^{12\times 2}$ and $c \in \mathbb{R}^{4\times 2}$, a vector $k \in \{1, \ldots, 4\}^{12}$ indicating the cluster number, i.e., $k(i) = p$ meaning that X_i belongs to the cluster of centroid c_p. The required starting point corresponds to

$$k = [1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4]$$

The algorithm then just iterates between forming the optimal centroids corresponding to the given clusters in k by just computing their median, e.g.,

```
for p = 1 : 4
    c( p , : ) = median( X( clusters == p , : ) , 1 );
end
```

(note the importance of the ", 1" parameter, without which when the cluster is a singleton the result is a single number: the median of the entries), and then recomputing the clusters as the points having the minumum distance from the given centroids, e.g.,

```
for i = 1 : 12
    dist = vecnorm( c - X( i , : ) , 1 , 2 );
    [ ~ , ci ] = min( dist );
    clusters( i ) = ci;
end
```

The process ends when the objective value, computed as in

```
v = sum( vecnorm( X - c( clusters , : ) , 1 , 2 ) );
```

stops strictly decreasing from one iteration to the next.

Applied to the given data the algorithm should perform 4 iterations, with the objective value starting at 5.6000 and terminating at 2.0000, with the final centroids

c				
	0.75	0.95	0.20	1.00
	0.45	0.05	0.80	0.90

and the corresponding final clusters

$$k = [4, 1, 3, 2, 1, 3, 3, 3, 4, 2, 3, 4]$$

To verify whether the solution is optimal one could write an exact MILP formulation of the problem, such as

$$
\begin{array}{ll}
\min \sum_{i=1}^{12} \sum_{p=1}^{4} \| v_{ip} \|_1 & \\
(\overline{x} - X_i) z_{ip} \geq v_{ip} \geq (\underline{x} - X_i) z_{ip} & p = 1, \ldots, 4 \ , \ i = 1, \ldots, 12 \\
c_p - X_i z_{ip} - \underline{x}(1 - z_{ip}) \geq v_{ip} \geq c_p - X_i z_{ip} - \overline{x}(1 - z_{ip}) & p = 1, \ldots, 4 \ , \ i = 1, \ldots, 12 \\
\overline{x} \geq c_p \geq \underline{x} & p = 1, \ldots, 4 \\
\sum_{p=1}^{4} z_{ip} = 1 & i = 1, \ldots, 12 \\
z_{ip} \in \{ 0 , 1 \} & p = 1, \ldots, 4 \ , \ i = 1, \ldots, 12
\end{array}
$$

for properly defined worst-case bounds $\overline{x}$ and $\underline{x}$ (the maximum and minimum of X over the columns, respectively) and then solving it with an exact MILP solver. Since MILP solvers do not generally support the L_1 norm as a primitive, one should typically reformulate it via auxiliary variables and linear constraints, using the modelling trick whereby

$$\min\{ \| x \|_1 \ : \ x \in \mathbb{R}^n \}$$

is equivalent to

$$
\begin{array}{ll}
\min \sum_{i=1}^{n} v_i & \\
v_i \geq x_i \ , \ v_i \geq -x_i & i = 1, \ldots, n
\end{array}
$$

Doing so would reveal that the optimal solution has value 1.9000, corresponding to centroids

c	0.8	0.9	0.1	1.0
	1.0	0.5	0.8	0.1

and the corresponding clusters

$$k = [1 , 2 , 3 , 4 , 2 , 3 , 3 , 1 , 2 , 4 , 3 , 1]$$

Thus, in this case k-median does not provide an optimal solution, which is not surprising since it's only a heuristic approach. On the other hand, the solution via the MILP solver will typically be orders of magnitude slower than that using k-median, and therefore only practical for these toy instances.

2) Solve the box-constrained quadratic optimization problem

$$(P)\ \ \min\left\{ x^TQx/2+qx\,:\,0\le x\le u\right\}\quad\text{with data}\quad Q=\begin{bmatrix}4&2&0\\2&4&2\\0&2&4\end{bmatrix}\quad,\quad q=\begin{bmatrix}-20\\-24\\-20\end{bmatrix}\quad,\quad u=\begin{bmatrix}3\\3\\3\end{bmatrix}$$

using the Lagrangian Dual method where all the "box" constraints are relaxed. The optimization of the Lagrangian Dual have to be carried using a projected gradient (projection over sign constraints being trivial) starting from all-0 Lagrangian multipliers and using *fixed stepsize* $\alpha=1$. Briefly describe the fundamental mathematical aspects of method, specifically commenting to its applicability to the instance at hand, then write a `Matlab` code that implements the algorithm, succinctly describe it (the fundamental steps, leaving aside the unnecessary details) and run it on the instance above and report the obtained (approximate) solution (detailing the stopping condition and what tolerances have been used in it). Optionally, augment the approach with a Lagrangian heuristic (describing it) and comment on the effect it has on the efficiency and effectiveness of the approach.

SOLUTION

The Lagrangian Dual approach to (P) is based on forming the *Lagrangian relaxation*

$$\psi(\lambda^+,\lambda^-)=\min\left\{ x^TQx/2+qx+\lambda^-(-x)+\lambda^+(x-u)\right\}$$

w.r.t. the "box" constraints. It is well-known that, for $\lambda^+\ge 0$ and $\lambda^-\ge 0$, $\psi(\lambda^+,\lambda^-)\le\nu(P)$, which immediately leads to the definition of the Lagrangian Dual

$$(D)\ \ \max\left\{\psi(\lambda^+,\lambda^-)\,:\,\lambda^+\ge 0\,,\ \lambda^-\ge 0\right\}$$

for which we know that $\nu(D)=\nu(P)$, and therefore solving (D) can be considered equivalent to solving (P). Crucially, $Q\succ 0$, as it can be verified with

```
>> eig(Q)'
ans = 1.1716    4.0000    6.8284
```

This means that $\psi(\lambda^+,\lambda^-)<\infty$ everywhere, as the optimal solution to the Lagrangian relaxation is obtained simply as

$$x^*(\lambda^+,\lambda^-)=-Q^{-1}(q-\lambda^-+\lambda^+)$$

and that $\psi\in C^1$, where

$$\nabla\psi(\lambda^+,\lambda^-)=[\,x^*(\lambda^+,\lambda^-)-u\,,\,-x^*(\lambda^+,\lambda^-)\,]$$

As a consequence, $x^*(\lambda^+_*,\lambda^-_*)$ for the optimal solution $(\lambda^+_*,\lambda^-_*)$ of (D) is optimal for (P), and therefore solving (D) is indeed equivalent (minus tolerances issues, to be discussed later) to solving (P).

For efficiency, one should invert Q only once; in fact it's better to rather use a Cholesky factorisation, such as in

```
R = chol( Q );
```

so that $x^*(\lambda^+,\lambda^-)$ can be obtained in $O(n^2)$ via two backsolves, as in

```
opts.LT = true;
z = linsolve( R' , - q - lambdaplus + lambdaminus , opts );
opts.LT = false;
opts.UT = true;
x = linsolve( R , z , opts );
```

Differentiability of ψ makes it possible to use a projected gradient approach, since projecting the anti-gradient on the sign constraints is trivial. What is not necessarily trivial is that a line search would be needed, but a fixed stepsize of $\alpha=1$ works surprisingly well in this instance and therefore that complexity can be avoided. The only issue is that the step need be chosen in such a way that the next iterate does not violate the sign constraints, and therefore some logic akin to

```
maxt = 1;
ind = dplus < 0;
maxt = min( maxt , min( - lambdaplus( ind ) ./ dplus( ind ) ) );
ind = dminus < 0;
maxt = min( maxt , min( - lambdaminus( ind ) ./ dminus( ind ) ) );
```

is necessary in the stepsize selection. After that, the update of the iterate is just

```
lambdaplus = lambdaplus + maxt * dplus;
lambdaminus = lambdaminus + maxt * dminus;
```

and the algorithm can stop when

```
norm( dplus ) + norm( dminus ) <= eps
```

where we used `eps = 1e-6`. Ran with the standard starting point $\lambda^+=\lambda^-=0$ the algorithms stops in around 20 iterations reporting optimal value `-102.0000`, which is indeed the optimal value of (P).

For the optional part, a Lagrangian heuristic can be very easily obtained by projecting $x^*(\lambda^+, \lambda^-)$ on the box, which just amounts at

```
x( x < 0 ) = 0;
ind = x > u;
x( ind ) = u( ind );
```

Then, the objective value of such a solution, computed just as

```
0.5 * x' * Q * x + q' * x
```

is a valid upper bound on $\nu(P)$. Keeping the x^* with the best (lowest) one—as the upper bound values are not guaranteed to be monotone—and comparing it with the current lower bound $\psi(\lambda^+, \lambda^-)$—which instead is monotone increasing—provides an estimate of the quality of x^* as an approximation to the optimal solution of the problem, thereby providing an alternative stopping criterion. Doing so allows to get a feasible solution $x = [\,3.0000\,,\,3.0000\,,\,3.0000\,]$ with an error less than `1e-6` in less than 10 iterations, and the optimal solution $x^* = [\,3.0000\,,\,3.0000\,,\,3.0000\,]$ (with the corresponding $\lambda^+_* = [\,2.0000\,,\,0.0000\,,\,2.0000\,]$ and $\lambda^- = 0$) with a guaranteed error of less than `1e-12` in less than 20 iterations.

3) Consider the following constrained multiobjective optimization problem:

$$\begin{cases} \min\ (x_1 + x_2 - x_3,\ x_1 - x_3) \\ x_1 - x_2 - x_3 \leq 0 \\ -x_1 \leq 0 \\ x_3 \leq 1 \\ (x_1, x_2, x_3) \in \mathbb{R}^3 \end{cases}$$

(a) Prove that the problem admits a Pareto minimum point.

(b) Find the set of all weak Pareto minima.

(c) Find a suitable subset of Pareto minima.

(d) Does the problem admit any ideal minimum?

SOLUTION

We preliminarly observe that the problem is linear, since the objective and the constraint functions are linear. Therefore the set of weak minima coincides with the set of solutions of the scalarized problems (P_{α_1}), i.e.

$$\begin{cases} \min\ \psi(\alpha_1) := \alpha_1(x_1 + x_2 - x_3) + (1 - \alpha_1)(x_1 - x_3) = x_1 + \alpha_1 x_2 - x_3 \\ x_1 - x_2 - x_3 \leq 0 \\ -x_1 \leq 0 \\ x_3 \leq 1 \\ (x_1, x_2, x_3) \in \mathbb{R}^3 \end{cases}$$

where $0 \leq \alpha_1 \leq 1$, while the set of minima coincides with the set of solutions of the scalarized problems (P_{α_1}), where $0 < \alpha_1 < 1$.

(a) Let $\alpha_1 = \frac{1}{2}$, then, for every feasible point, we have:

$$\psi(\alpha_1) = \frac{1}{2}(x_1 + x_2 - x_3) + \frac{1}{2}(x_1 - x_3) = x_1 - x_3 + \frac{1}{2}x_2 \geq -1 + \frac{1}{2}(x_1 - x_3) \geq -1 - \frac{1}{2}.$$

Therefore $P_{\frac{1}{2}}$ admits finite optimum and the related optimal solutions are minima for the given problem.

(b) - (c) By solving P_{α_1} with Matlab, we have:

```
C = [1 1 -1;  1 0  -1] ;

A =[ 1  -1 -1;   -1 0 0; 0 0 1 ];

 b = [0 0 1]';

 % solve the scalarized problem with 0 =< alfa1 =< 1

MINIMA=[Inf,Inf,Inf,Inf]; %first column: values of \alfa1, (in the same row) columns 2,3,4: optimal solution

lambda=[Inf,Inf,Inf,Inf]; %first column: values of \alfa1, (in the same row) columns 2,3,4: Lagrange multiplier

 for alfa1 = 0 : 0.01 : 1
 [x,fval,exitflag,output,Lambda] = linprog(alfa1*C(1,:)+(1-alfa1)*C(2,:),A,b) ;

      MINIMA=[MINIMA; alfa1 x'];
     lambda=[lambda;alfa1,Lambda.ineqlin'];

 end
```

We obtain

$$x(\alpha_1) = (0, -1, 1) \quad for\ 0 \leq \alpha_1 \leq 1, \quad \lambda(\alpha_1) = \begin{cases} (0, 1, 1) \quad for\ \alpha_1 = 0 \\ (\lambda_1(\alpha_1), \lambda_2(\alpha_1), \lambda_3(\alpha_1)) > (0, 0, 0) \quad for\ 0 < \alpha_1 \leq 1 \end{cases}$$

Since the problem is linear then the KKT conditions provide a necessary and sufficient condition for an optimal solution of

(P_{α_1}):

$$\begin{cases} 1+\lambda_1-\lambda_2=0 \\ \alpha_1-\lambda_1=0 \\ -1-\lambda_1+\lambda_3=0 \\ \lambda_1(x_1-x_2-x_3)=0 \\ \lambda_2(-x_1)=0 \\ \lambda_3(x_3-1)=0 \\ x_1-x_2-x_3\leq 0 \\ -x_1\leq 0 \\ x_3\leq 1 \\ \lambda\geq 0 \\ 0\leq\alpha_1\leq 1, \end{cases}$$

It is known that if, for fixed α_1, there exists a solution (λ, x) of the KKT system with $\lambda > 0$ then x is the unique solution of P_{α_1}. It follows that:

(i) For $0 < \alpha_1 \leq 1$, $x(\alpha_1) = (0, -1, 1)$ is the unique optimal solution for P_{α_1}, being $\lambda(\alpha_1) > 0$, so that $\bar{x} = (0, -1, 1)$ is the unique minimum point of the given problem.

(ii) For $\alpha_1 = 0$ the set of optimal solutions of P_0 is given by the following system

$$\begin{cases} x_1-x_2-x_3\leq 0 \\ -x_1=0 \\ x_3=1 \\ (x_1,x_2,x_3)\in\mathbb{R}^3 \end{cases}$$

Then, $Weak\ Min(P) = \{(x_1, x_2, x_3) : x_1 = 0,\ x_2 \geq -1, x_3 = 1\}$,

$$Min(P) = \{\bar{x}\}.$$

(d) Since the optimal solutions of P_0 and P_1 coincide then $\bar{x} = (0, -1, 1)$ is an ideal minimum, indeed it minimizes both the objective functions.

4) Consider the following bimatrix game:

$$C_1 = \begin{pmatrix} 5 & 4 & 3 \\ 7 & 5 & 2 \end{pmatrix} \qquad C_2 = \begin{pmatrix} 4 & 3 & 2 \\ 2 & 1 & 5 \end{pmatrix}$$

(a) Find the set of pure strategies Nash equilibria, if any. Alternatively, show that no pure strategies Nash equilibrium exists.

(b) Find a mixed strategies Nash equilibrium.

SOLUTION

(a) Strategy 1 of Player 2 is dominated by Strategy 2, so that $y_1 = 0$ and column 1 in the two matrices can be deleted. The reduced game is given by the matrices

$$C_1^R = \begin{pmatrix} 4 & 3 \\ 5 & 2 \end{pmatrix} \qquad C_2^R = \begin{pmatrix} 3 & 2 \\ 1 & 5 \end{pmatrix}$$

The minima on the columns of C_1^R are the elements of the principal diagonal, but no elements on the principal diagonal of C_2^R are minima on the rows of C_2^R, which implies that no pure strategies Nash equilibria exist.
This will also be shown in part (b) in the wider context of mixed strategies Nash equilibria.

(b) Consider the reduced game obtained in (a). The optimization problem associated with Player 1 is

$$\begin{cases} \min\ x^T C_1^R y = (4x_1 + 5x_2)y_2 + (3x_1 + 2x_2)y_3 \\ x_1 + x_2 = 1 \\ x_1, x_2 \geq 0 \end{cases} \equiv \begin{cases} \min\ (1 - 2y_3)x_2 - y_3 + 4 \\ 0 \leq x_2 \leq 1 \end{cases} \qquad (P_1(y_3))$$

where, we have eliminated the variables x_1 and y_2, since $x_1 = 1 - x_2$ and $y_2 = 1 - y_3$,taking into account that $y_1 = 0$. Then, the best response mapping associated with $P_1(y_3)$ is:

$$B_1(y_3) = \begin{cases} 0 & \text{if } y_3 \in [0, 1/2) \\ [0,1] & \text{if } y_3 = 1/2 \\ 1 & \text{if } y_3 \in (1/2, 1) \end{cases}$$

Similarly, the optimization problem associated with Player 2 is

$$\begin{cases} \min\ x^T C_2^R y = x_1(3y_2 + 2y_3) + x_2(y_2 + 5y_3) \\ y_2 + y_3 = 1 \\ y_2, y_3 \geq 0 \end{cases} \equiv \begin{cases} \min\ (5x_2 - 1)y_3 - 2x_2 + 3 \\ 0 \leq y_3 \leq 1 \end{cases} \qquad (P_2(x_2))$$

Then, the best response mapping associated with $P_2(x_2)$ is:

$$B_2(x_2) = \begin{cases} 0 & \text{if } x_2 \in (1/5, 1] \\ [0,1] & \text{if } x_2 = 1/5 \\ 1 & \text{if } x_2 \in [0, 1/5) \end{cases}$$

The couples (x_2, y_3) such that $x_2 \in B_1(y_3)$ and $y_3 \in B_2(x_2)$ are

1. $x_2 = \frac{1}{5}, y_3 = \frac{1}{2}$,

so that, recalling that $y_1 = 0$,

- $(x_1, x_2) = (\frac{4}{5}, \frac{1}{5})$, $\quad (y_1, y_2, y_3) = (0, \frac{1}{2}, \frac{1}{2})$, is a mixed strategies Nash equilibrium and no pure Nash equilibrium strategy exists.