1) Consider the (univariate) regression task defined by the data

x	-5	-4	-3	-2	-1	0	1	2	3	4	5
y	-284	-168	-88	-38	-12	-4	-8	-18	-28	-32	-24

Write a `Matlab` code that solves the polynomial interpolation problem for degree $k = 3$ and succinctly describe it (the fundamental steps, leaving aside the unnecessary details). Then run it on the data above and report the obtained solution (the coefficients of the polynomial) and the Mean Square Error (MSE) between the interpolated polynomial and the original data, commenting on the quality of the obtained interpolation.

SOLUTION

We want to construct the third degree polynomial

$$c_3p^3 + c_2p^2 + c_1p + c_0$$

that best approximates the given y values on the given x ones in the Mean Square Error sense, i.e., minimizing

$$\sum_{i=1}^{11}(y_i - c_3x_i^3 + c_2x_i^2 + c_1x_i + c_0)^2$$

In order to solve this polynomial interpolation problem we first construct, given the vector $x \in \mathbb{R}^11$, the matrix $X \in \mathbb{R}^{4\times 11}$ whose i-th row is

$$[\, 1 \,,\, x_i \,,\, x_i^2 \,,\, x_i^3 \,]$$

This can be done, e.g., with the `Matlab` code

```
X = ones( 11 , 4 );
X( : , 2 ) = x;
for i = 3 : 4
    X( : , i ) = X( : , i - 1  ) .* x;
end
```

assuming that $x \in \mathbb{R}^{11}$ is a column vector (i.e., `size( x ) = { 11 , 1 }`). Then we solve the Linear Least Square problem

$$\min \left\{ \, \| \, Xc - y \, \|_2^2 \; : \; c \in \mathbb{R}^4 \, \right\}$$

This requires computing the pseudo-inverse of X, and it can be performed with just the single command

```
c = y' / X';
```

that produces a row vector $c \in \mathbb{R}^4$ (i.e., `size( c ) = { 1 , 4 }`) containing the coefficients of the powers of x in increasing order, i.e., $c[1] = c_0$ is the constant coefficient, $c[2] = c_1$ the first-order coefficient, and so on. So doing produces the answer

```
c = -4.0000    1.0000   -6.0000    1.0000
```

corresponding to the polynomial

$$p^3 - 6p^2 + p - 4$$

To compute the MSE one can just compute the values of the polynomial as

```
z = x.^3 - 6 * x.^2 + x - 4
```

and then the error as `y - z`: this results in the all-0 vector, and hence in a MSE of 0. In other words, the obtained polynomial exactly fits the provided data.

2) Solve the box-constrained quadratic optimization problem

$$\min\{\, x^T Qx/2 + qx \,:\, 0 \leq x \leq u \,\} \quad \text{with data} \quad Q = \begin{bmatrix} 3 & 1 \\ 1 & 3 \end{bmatrix} \quad , \quad q = \begin{bmatrix} -3 \\ -9 \end{bmatrix} \quad , \quad u = \begin{bmatrix} 2.0 \\ 3.1 \end{bmatrix}$$

using the Frank-Wolfe (conditional gradient) method starting from the point "in the middle of the box" $x^0 = u/2$. Write a `Matlab` code that implements the algorithm and succinctly describe it (the fundamental steps, leaving aside the unnecessary details), then 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 comment on the number of iterations and the convergence rate of the approach.

SOLUTION

Starting from the initial iterate `x = u / 2`, the Frank-Wolfe algorithm computes the current function value and gradient as

```
v = 0.5 * x’ * Q * x + q’ * x;
g = Q * x + q;
```

One then needs to solve the problem

$$y = \operatorname{argmin}\{\, gx \,:\, 0 \leq x \leq u \,\}$$

which trivially decomposes for each component: $y_i = 0$ if $g_i \geq 0$, and $y_i = u_i$ otherwise. This can be obtained, e.g., by the simple code

```
y = zeros( n , 1 );
ind = g < 0;
y( ind ) = u( ind );
```

It is then useful to compute the lower bound corresponding to the linearization, i.e., $l = f(x) + g(y - x)$, or simply

```
l = v + g’ * ( y - x );
```

We know that $v \geq f_* \geq l$, hence one can stop when the upper estimate on the relative error of x given by

$$(\, v - l \,) \,/\, |\, l \,|$$

is smaller than some threshold, say `1e-6` (some adjustment is clearly needed for the case where $l = 0$ can happen, but this is not an issue in our instance). Since l is not necessarily nondecreasing over the iterations one may also want to keep the best (largest) value of l and use it for the comparison. Conversely, we know that v is necessarily decreasing: indeed, if the algorithms does not stop we know that `d = y - x` is a descent direction. One can then compute the unconstrained minimum along it with the usual formula

```
alpha = ( - g’ * d ) / d’ * Q * d;
```

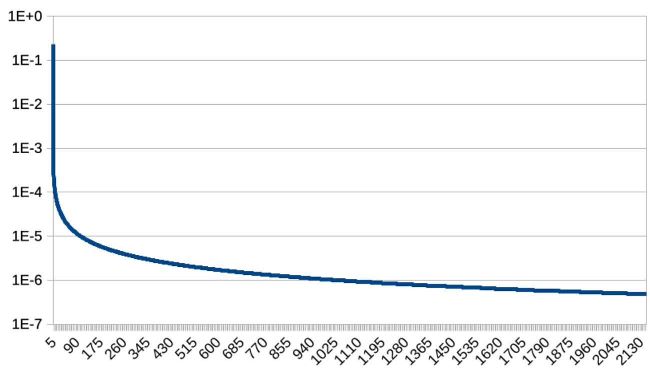
(where in general one should account for the possibility of the denumerator is zero, but again this is not happening in our instance). However, any step greater than 1 would lead to an unfeasible solution, so one actually has to take the minimum between `alpha` as computed above and 1. The corresponding next iterate

```
x = x + alpha * d;
```

is guaranteed to be feasible and the process can be repeated, eventually leading to convergence. With a tolerance of `1e-6` the process should take around 2000 iterations and return a solution close to

```
x = [ 0.0022 ; 2.9993 ] , v = -1.34999935e+01 , l = -1.35000070e+01
```

For the optional part one just has to have all the values of v at every iteration written down and compute the gap $(v - f_*)/f_*$ (where clearly $f_* = -13.5$). Then, plotting the resulting values in logarithmic scale one should obtain something like



The fact that the curve is not "straight" but it "flattens" (the derivative is negative but it clearly goes to 0) means that the convergence of the Frank-Wolfe approach in this instance is sublinear, as the theory predicts.

3) Consider the following multiobjective optimization problem:

$$\begin{cases} \min\ (x_1 + 2x_2^2,\ x_1^2 + 4x_2^2) \\ x_1^2 + 2x_2^2 - 4 \leq 0 \end{cases}$$

(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.

SOLUTION

(a) Since the feasible set X is compact and the objective function is continuous then the problem admits a (Pareto) minimum point.

(b) - (c) We preliminarly observe that the problem is convex, since the objective and the constraint functions are convex. Therefore the set of weak minima coincides with the set of solutions of the scalarized problems (P_{α_1}), where $0 \leq \alpha_1 \leq 1$, i.e.

$$\begin{cases} \min\ \alpha_1(x_1 + 2x_2^2) + (1-\alpha_1)(x_1^2 + 4x_2^2) =: \psi_{\alpha_1}(x) \\ x_1^2 + 2x_2^2 - 4 \leq 0 \end{cases}$$

For $0 \leq \alpha_1 < 1$, ψ_{α_1} is strongly convex so that (P_{α_1}) admits a unique optimal solution which is a minimum of (P).

We note that (P_{α_1}) is convex and differentiable and ACQ holds at any $x \in X$; then the KKT system provides a necessary and sufficient condition for an optimal solution of (P_{α_1}). KKT system is given by:

$$\begin{cases} (1-\alpha_1)2x_1 + \alpha_1 + 2\lambda x_1 = 0 \\ 2(4-2\alpha_1)x_2 + 4\lambda x_2 = 0 \\ \lambda(x_1^2 + 2x_2^2 - 4) = 0 \\ 0 \leq \alpha_1 \leq 1,\ \lambda \geq 0 \\ x_1^2 + 2x_2^2 - 4 \leq 0 \end{cases}$$

We obtain:

$$\begin{cases} x_1 = -\dfrac{\alpha_1}{2(1-\alpha_1+\lambda)},\ 1-\alpha_1+\lambda \neq 0 \\ x_2 = 0 \\ \lambda(x_1^2 - 4) = 0 \\ 0 \leq \alpha_1 \leq 1,\ \lambda \geq 0 \\ x_1^2 - 4 \leq 0 \end{cases}$$

Notice that $1 - \alpha_1 + \lambda = 0$ implies $\lambda = 0$, $\alpha_1 = 1$, which is impossible by the first equation in the KKT system.

From the complementarity condition $\lambda(x_1^2 - 4) = 0$, we have the two cases: $I)$ $\lambda = 0$, $II)$ $x_1 = \pm 2$.

In case $I)$, $\lambda = 0$, $0 \leq \alpha_1 < 1$, we obtain:

$$x_1 = -\frac{\alpha_1}{2(1-\alpha_1)},\ x_2 = 0$$

so that $Min(P) \supseteq \{(x_1, x_2) : -2 \leq x_1 \leq 0,\ x_2 = 0\}$, noticing that P_0 has a unique optimal solution.

In case $II)$, for $x_1 = 2$, the first equation is impossible, taking into account that $\lambda \geq 0$.

For $x_1 = -2$, we obtain the point $(-2, 0)$, already considered.

It remains to consider the case $\alpha_1 = 1$, $\lambda \neq 0$. The system becomes:

$$\begin{cases} 1 + 2\lambda x_1 = 0 \\ 4x_2(1+\lambda) = 0 \\ x_1^2 + 2x_2^2 - 4 = 0 \\ \lambda > 0 \end{cases}$$

with the unique solution $(x_1, x_2) = (-2, 0)$, $\lambda = \frac{1}{4}$.

In conclusion: $Weak\ Min(P) = Min(P) = \{(x_1, x_2) : -2 \leq x_1 \leq 0,\ x_2 = 0\}$.

4) Consider the following bimatrix game:

$$C_1 = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 2 & 1 & 0 \end{pmatrix} \qquad C_2 = \begin{pmatrix} 0 & 2 & 1 \\ 2 & 1 & 3 \\ 1 & 2 & 0 \end{pmatrix}$$

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

(b) Find a mixed strategy Nash equilibrium.

SOLUTION

(a) Strategy 3 of Player 1 is dominated by Strategy 1, so that row 3 in the two matrices can be deleted. Now Strategy 3 of Player 2 is dominated by Strategy 1 and column 3 in the two matrices can be deleted. The reduced game is given by the matrices

$$C_1^R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad C_2^R = \begin{pmatrix} 0 & 2 \\ 2 & 1 \end{pmatrix}$$

Now, it is simple to show that no pure strategies Nash equilibria exist. Indeed, considering Player 1, the possible couples of pure strategies Nash equilibria could be $(2, 1)$ and $(1, 2)$, while considering Player 2 the pure strategies Nash equilibria could be $(1, 1)$ and $(2, 2)$. Since there are no common couples, 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

$$\begin{cases} \min\ x^T C_1^R y = x_1 y_1 + x_2 y_2 \\ x_1 + x_2 = 1 \\ x_1, x_2 \geq 0 \end{cases} \equiv \begin{cases} \min\ (2y_1 - 1)x_1 + 1 - y_1 \\ 0 \leq x_1 \leq 1 \end{cases} \qquad (P_1(y_1))$$

Then, the best response mapping associated with $P_1(y_1)$ is:

$$B_1(y_1) = \begin{cases} 0 & \text{if } y_1 \in [1/2, 1) \\ [0,1] & \text{if } y_1 = 1/2 \\ 1 & \text{if } y_1 \in [0, 1/2) \end{cases}$$

Similarly, the optimization problem associated with Player 2 is

$$\begin{cases} \min\ x^T C_2^R y = 2x_2 y_1 + (2x_1 + x_2) y_2 \\ y_1 + y_2 = 1 \\ y_1, y_2 \geq 0 \end{cases} \equiv \begin{cases} \min\ (1 - 3x_1)y_1 + x_1 + 1 \\ 0 \leq y_1 \leq 1 \end{cases} \qquad (P_2(x_1))$$

Then, the best response mapping associated with $P_2(x_1)$ is:

$$B_2(x_1) = \begin{cases} 0 & \text{if } x_1 \in [0, 1/3) \\ [0,1] & \text{if } x_1 = 1/3 \\ 1 & \text{if } x_1 \in (1/3, 1] \end{cases}$$

The only couple $(\bar{x}_1, \bar{y}_1)$ such that $\bar{x}_1 \in B_1(\bar{y}_1)$ and $\bar{y}_1 \in B_2(\bar{x}_1)$ is $\bar{x}_1 = \frac{1}{3}, \bar{y}_1 = \frac{1}{2}$, so that

$$\bar{x} = (\frac{1}{3}, \frac{2}{3}, 0), \quad \bar{y} = (\frac{1}{2}, \frac{1}{2}, 0)$$

is the unique mixed strategy Nash equilibrium for the given game, which also shows that no pure strategies Nash equilibrium exists.

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

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

into four distinct clusters under the L_2 distance. Write a `Matlab` code that heuristically solves the clustering problem using the k-means 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_2 norm requires finding four vectors $c_p \in \mathbb{R}^2$, $p = 1, \dots, 4$ that solve

$$\min\{\, f(c) = \textstyle\sum_{1,\dots,12} \min_{1,\dots,4} \| c_p - X_i \|_2^2 \,:\, c \in \mathbb{R}^{4\times 2} \}$$

or equivalently, with the introduction of binary variables,

$$\begin{array}{lr} \min \; \sum_{i=1,\dots,12} \sum_{p=1,\dots,4} z_{ip} \| c_p - X_i \|_2^2 & \\ \quad \sum_{p=1,\dots,4} z_{ip} = 1 & i = 1, \dots, 12 \\ \quad z_{ip}\{\,0\,,\,1\,\} & p = 1, \dots, 4\;,\; i = 1, \dots, 12 \end{array}$$

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 $k \in \{1, \dots, 4\}^{12}$ 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 mean, e.g.,

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

(note the importance of the ", 1" parameter, without which the when the cluster is a singleton the result is a single number: the mean 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 , : ) , 2 , 2 );
    [ ~ , ci ] = min( dist );
    clusters( i ) = ci;
end
```

The process ends when the objective value, computed as in

```
v = sum( vecnorm( X - c( clusters , : ) , 2 , 2 ) .^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 1.2467 and terminating at 0.24833, with the final centroids

c			
0.6750	0.2000	0.7667	0.2250
0.8000	1.0000	0.1667	0.4000

and the corresponding final clusters

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

(note how cluster 2 is in fact a singleton corresponding to $X_6 = [0.1, 1.0]$).

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

$$
\begin{array}{lll}
\min & \sum_{i=1,\dots,12} \sum_{p=1,\dots,4} \| v_{ip} \|_2^2 & \\
& (\overline{x} - X_i) z_{ip} \geq v_{ip} \geq (\underline{x} - X_i) z_{ip} & p = 1, \dots, 4 \,,\ i = 1, \dots, 12 \\
& c_p - X_i z_{ip} - \underline{x}(1 - z_{ip}) \geq v_{ip} \geq c_p - X_i z_{ip} - \overline{x}(1 - z_{ip}) & p = 1, \dots, 4 \,,\ i = 1, \dots, 12 \\
& \overline{x} \geq c_p \geq \underline{x} & p = 1, \dots, 4 \\
& \sum_{p=1,\dots,4} z_{ip} = 1 & i = 1, \dots, 12 \\
& z_{ip} \in \{ 0 \,,\, 1 \} & p = 1, \dots, 4 \,,\ i = 1, \dots, 12
\end{array}
$$

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 MIQP solver. Doing so would reveal that the optimal solution has value 0.24833, i.e., exactly the one obtained by the k-means heuristic, which in this case is therefore exact.

2) Let f be a C^2 function. Define the concepts of L-smoothness and τ-convexity. Then state and prove what is the optimal fixed stepsize for the gradient method for an L-smooth and τ-convex function, and what is the corresponding convergence rate.

SOLUTION

f is L-smooth if the gradient of f is Lipschitz continuous, i.e.,

$$|| \nabla f(x) - \nabla f(y) || \leq L|| x - y || \qquad \forall x \, , \, y$$

f is τ-convex if it is strongly convex modulus τ, i.e., $f(x) - \tau/2 \| x \|^2$ is convex $\forall x$, or alternatively $\alpha f(x) + (1 - \alpha)f(z) \geq f(\alpha x + (1 - \alpha)z) + \tau/2\alpha(1 - \alpha)\| z - x \|^2$ for all x, z and $\alpha \in [0 \, , \, 1]$. For a convex C^2 function the two conditions are also equivalent to

$$\tau I \preceq \nabla^2 f(x) \preceq LI \;\equiv\; \tau \leq \lambda^n \leq \lambda^1 \leq L \qquad \forall x$$

where as usual λ^1 and λ^n are the maximum and minimum eigenvalue of $\nabla^2 f(x)$, respectively.
The generic iterate of the gradient method with fixed stepsize reads

$$x^{k+1} = x^k - \alpha \nabla f(x^k)$$

Owing to the fact that $\nabla f(x_*) = 0$, we can then write

$$x^{k+1} - x_* = x^k - x_* - \alpha(\nabla f(x^k) - \nabla f(x_*))$$

Since $f \in C^2$, ∇f is in particular continuous and we can apply the Mean Value Theorem on ∇f to establish that

$$\exists \, w \in [x^k \, , \, x_*] \text{ s.t. } \nabla f(x^*) - \nabla f(x_*) = \nabla f^2(w)(x^k - x_*)$$

This finally yields

$$x^{k+1} - x_* = x^k - x_* - \alpha \nabla f^2(w)(x^k - x_*) = (I - \alpha \nabla f^2(w))(x^k - x_*)$$

whence,

$$\| x^{k+1} - x_* \| \leq \| I - \alpha \nabla f^2(w) \| \| x^k - x_* \|$$

If for some α one has $r = \| I - \alpha \nabla f^2(w) \| < 1$ then the algorithm is linearly convergent with rate r. In particular, then, the optimal stepsize will be the one solving

$$\min\{ \| I - \alpha \nabla f^2(w) \| \, : \, \alpha \geq 0 \}$$

Now, by definition of the L_2 norm for matrices,

$$\| I - \alpha \nabla f^2(w) \| = \max\{ | 1 - \alpha \lambda_1(\nabla f^2(w)) | \, , \, | 1 - \alpha \lambda_n(\nabla f^2(w)) | \}$$

It is easy to check that when $1-\alpha\lambda_n \geq 1-\alpha\lambda_1 \geq 0$, increasing α decreases the max. Symmetrically, when $0 \leq \alpha\lambda_n-1 \leq \alpha\lambda_1-1$, decreasing α decreases the max. Thus, the optimal α must be s.t. $1-\alpha\lambda_n > 0$ and $1-\alpha\lambda_1 < 0$, i.e., $r = \max\{ -1+\alpha\lambda_1 \, , \, 1-\alpha\lambda_n \}$. Of course λ_1 and λ_n depend on w and are unknown in general, but $L \geq \lambda_1$ and $\tau \leq \lambda_n$ whence $r \leq \bar{r} = \max\{ -1+\alpha L \, , \, 1-\alpha\tau \}$. Now, clearly the two terms in the max behave symmetrically: if one grows the other decreases and vice-versa. It is therefore obvious that the optimal α is the one where they are equal (for otherwise one could decrease the larger one and increase the smaller one). It is easy to check that this happens when $\alpha = 2 / (L + \tau)$. In fact,

$$-1 + 2L / (L + \tau) = (-L - \tau + 2L) / (L + \tau) = (L - \tau) / (L + \tau)$$

$$1 - 2\tau / (L + \tau) = (L + \tau - 2\tau) / (L + \tau) = (L - \tau) / (L + \tau)$$

All in all, with $\alpha = 2 / (L + \tau)$ one has $\| x^{k+1} - x_* \| \leq r^k \| x^1 - x_* \|$ with $r = (L - \tau) / (L + \tau) < 1$, i.e., the algorithm converges linearly. An alternative way of writing the result is

$$\bar{r} = (L - \tau) / (L + \tau) = (\bar{\kappa} - 1) / (\bar{\kappa} + 1)$$

with with $\bar{\kappa} = L / \tau$ the worst-case condition number of ∇f^2. This yields the intuitive result that the convergence rate worsens the more ∇f^2 can be ill-conditioned.

3) Consider the following multiobjective optimization problem:

$$\begin{cases} \min\ (3x_1^2 + x_2^2 - x_1x_2,\ 2x_1 - x_2) \\ -2x_1 + x_2 - 2 \leq 0 \end{cases}$$

(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.

SOLUTION

(a) Since the feasible set X is nonempty, closed and the objective function f_1 is strongly convex then the problem admits at least a (Pareto) minimum point.

(b) We preliminarly observe that the problem is convex, since the objective and the constraint functions are convex. Therefore the set of weak minima coincides with the set of solutions of the scalarized problems (P_{α_1}), where $0 \leq \alpha_1 \leq 1$, i.e.

$$\begin{cases} \min\ \alpha_1(3x_1^2 + x_2^2 - x_1x_2) + (1-\alpha_1)(2x_1 - x_2) =: \psi_{\alpha_1}(x) \\ -2x_1 + x_2 - 2 \leq 0 \end{cases}$$

For $0 < \alpha_1 \leq 1$, ψ_{α_1} is strongly convex so that P_{α_1} admits a unique optimal solution which is a minimum of (P).

We note that (P_{α_1}) is convex and differentiable and ACQ holds at any $x \in X$; then the KKT system provides a necessary and sufficient condition for an optimal solution of (P_{α_1}). KKT system is given by:

$$\begin{cases} 6\alpha_1x_1 - \alpha_1x_2 + 2 - 2\alpha_1 - 2\lambda = 0 \\ 2\alpha_1x_2 - \alpha_1x_1 - 1 + \alpha_1 + \lambda = 0 \\ \lambda(-2x_1 + x_2 - 2) = 0 \\ 0 \leq \alpha_1 \leq 1,\ \lambda \geq 0 \\ -2x_1 + x_2 - 2 \leq 0 \end{cases}$$

By eliminating λ in the first equality, we obtain:

$$\begin{cases} \alpha_1(4x_1 + 3x_2) = 0 \\ \lambda = -2\alpha_1x_2 + \alpha_1x_1 + 1 - \alpha_1 \\ \lambda(-2x_1 + x_2 - 2) = 0 \\ 0 \leq \alpha_1 \leq 1,\ \lambda \geq 0 \\ -2x_1 + x_2 - 2 \leq 0 \end{cases} \tag{1}$$

For $\alpha_1 = 0$, we obtain $\lambda = 1$, $-2x_1 + x_2 - 2 = 0$, so that

$$\text{Weak Min(P)} \supseteq \{(x_1, x_2) : -2x_1 + x_2 - 2 = 0\}.$$

For $0 < \alpha_1 \leq 1$ the system (1) becomes

$$\begin{cases} 4x_1 + 3x_2 = 0 \\ \lambda = -2\alpha_1x_2 + \alpha_1x_1 + 1 - \alpha_1 \\ \lambda(-2x_1 + x_2 - 2) = 0 \\ 0 < \alpha_1 \leq 1,\ \lambda \geq 0 \\ -2x_1 + x_2 - 2 \leq 0 \end{cases} \tag{2}$$

From the complementarity condition $\lambda(-2x_1+x_2-2) = 0$, we have the two cases: *I*) $\lambda = 0$, *II*) $\lambda \neq 0$ (*or*, $-2x_1+x_2-2 = 0$). In case *I*), $\lambda = 0$, by the first two equations we obtain:

$$x_1 = \frac{3}{11}\left(1 - \frac{1}{\alpha_1}\right),\ \ x_2 = -\frac{4}{11}\left(1 - \frac{1}{\alpha_1}\right)$$

Imposing the feasibility condition given by the last inequality, and recalling that $0 < \alpha_1 \leq 1$, we obtain $\frac{5}{16} \leq \alpha_1 \leq 1$ so that

$$Min(P) \supseteq \{(x_1, x_2) : x_2 = -\frac{4}{3}x_1, -\frac{3}{5} \leq x_1 \leq 0\}.$$

In case II), $\lambda \neq 0$ the the system (2) becomes

$$\begin{cases} 4x_1 + 3x_2 = 0 \\ \lambda = -2\alpha_1 x_2 + \alpha_1 x_1 + 1 - \alpha_1 \\ -2x_1 + x_2 - 2 = 0 \\ 0 < \alpha_1 \leq 1, \ \lambda > 0 \end{cases} \tag{3}$$

and we obtain the unique solution $(x_1, x_2) = (-\frac{3}{5}, \frac{4}{5})$, a point that we had already found as a minimum.

In conclusion:

$$\text{Weak Min(P)} = \{(x_1, x_2) : -2x_1 + x_2 - 2 = 0\} \quad \cup \{(x_1, x_2) : x_2 = -\frac{4}{3}x_1, -\frac{3}{5} \leq x_1 \leq 0\}.$$

$$Min(P) \supseteq \{(x_1, x_2) : x_2 = -\frac{4}{3}x_1, -\frac{3}{5} \leq x_1 \leq 0\}.$$

It is possible to show that $Min(P)$ actually coincides with the found minima.

4) Consider the following matrix game:

$$C = \begin{pmatrix} 5 & 4 & 3 & 5 \\ 6 & 7 & 8 & 2 \\ 5 & 3 & 4 & 4 \end{pmatrix}$$

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

(b) Find a mixed strategy Nash equilibrium which is not a pure strategy Nash equilibrium, if any. Alternatively, show that no Nash equilibrium of such kind exists.

SOLUTION (a) Considering Player 1, the possible couples of pure strategies Nash equilibria could be $(1, 1)$, $(3, 1)$, $(3, 2)$, $(1, 3)$ and $(2, 4)$ (minimal components on the columns), while considering Player 2 the pure strategies Nash equilibria could be $(1, 1)$, $(1, 4)$,$(2, 3)$ and $(3, 1)$ (maximal components on the rows). The common couples $(1, 1)$ and $(3, 1)$ are pure strategies Nash equilibria.

(b) Consider the linear optimization problem associated with Player 1:

$$\begin{cases} \min \quad v \\ 5x_1 + 6x_2 + 5x_3 \leq v \\ 4x_1 + 7x_2 + 3x_3 \leq v \\ 3x_1 + 8x_2 + 4x_3 \leq v \\ 5x_1 + 2x_2 + 4x_3 \leq v \\ x_1 + x_2 + x_3 = 1 \\ x \geq 0 \end{cases} \tag{4}$$

Pure strategies for Player 1 correspond to the solutions $(x^1, v^1) = (1, 0, 0, 5)$ and $(x^2, v^2) = (0, 0, 1, 5)$ and $y = (1, 0, 0, 0, 5)$ is a dual solution of (4) associated with the pure strategy $(1, 0, 0, 0)$ of Player 2. Since the problem is linear then any convex combination

$$\alpha(x^1, v^1) + (1 - \alpha)(x^2, v^2), \quad \alpha \in [0, 1]$$

is an optimal solution of (4).

For example, for $\alpha = \frac{1}{2}$ we obtain the solution $\hat{x} = (\frac{1}{2}, 0, \frac{1}{2}, 5)$ so that

$$\hat{x} = (\frac{1}{2}, 0, \frac{1}{2}), \quad \hat{y} = (1, 0, 0, 0)$$

is a mixed-strategies Nash equilibrium.

1) Consider the (univariate) regression task defined by the data

x	-5	-4	-3	-2	-1	0	1	2	3	4	5
y	-284	-168	-88	-38	-12	-4	-8	-18	-28	-32	-24

Write down the dual formulation of SVR (the one allowing for the kernel trick). Then, write a `Matlab` code that implements the training of the SVR and succinctly describe it (the fundamental steps, leaving aside the unnecessary details). Train the SVR with Gaussian kernel using the code with hyperparameters $C = 100$, $\epsilon =$ `1e-6` and $\sigma = 0.5$. Knowing that the data is generated by the polynomial $p^3 - 6p^2 + p - 4$, run the trained SVR on the interval $[-5\,,\,5]$ with small stepsize (say, `-5 : 0.01 : 5`) and compare graphically and/or algebraically the prediction with the underlying original data commenting on its quality.

SOLUTION

The dual formulation of SVR is

$$\begin{array}{lll} \max & \sum_{i\in I} y^i\alpha_i - \varepsilon\sum_{i\in I} v_i - \frac{1}{2}\sum_{i\in I}\sum_{j\in I}\alpha_i\kappa(\,x^i\,,\,x^j\,)\alpha_j & \\ & \sum_{i\in I}\alpha_i = 0 & \\ & v_i \geq \alpha_i \;\;,\;\; v_i \geq -\alpha_i & i \in I \\ & -\,C \leq \alpha_i \leq C & i \in I \end{array}$$

where $I = \{\,1, \ldots\, 11\,\}$ is the set of indices of data points, and $\kappa(\,\cdot\,,\,\cdot\,)$ is the kernel function. For the Gaussian kernel,

$$\kappa(\,x\,,\,z\,) = e^{-\|\,x-z\,\|^2\,/\,(2\sigma^2)}$$

Solving the dual formulation requires either an ad-hoc library with MATLAB interface, like LIBSVM, or an optimization library capable of solving convex Quadratic Programs, possibly with a modelling system as the interface. Using the YALMIP system the above formulation can be easily implemented as

```
alpha = sdpvar( n , 1 );
v = sdpvar( n , 1 );

F = [ - C <= alpha <= C ];
F = F + [ v >= alpha , v >= - alpha ];
F = F + [ ( ones( 1 , n ) * alpha == 0 ):'simplex' ];

c = (1/2) * alpha' * H * alpha - y' * alpha + ( eps * ones( 1 , n ) ) * v;
```

where `n = 11` and `H` is the Gram matrix, computed (for the Gaussian kernel) by

```
H = zeros( n , n );
s2 = 2 * sigma * sigma;
for i = 1 : n
    H( i , i ) = 1;
    for j = 1 : i - 1
        H( i , j ) = exp( - norm( X( i ) - X( j ) )^2 / s2 );
        H( j , i ) = H( i , j );
    end
end
```

Then, the optimization can be obtained by just calling

```
optimize( F , c , ops );
```

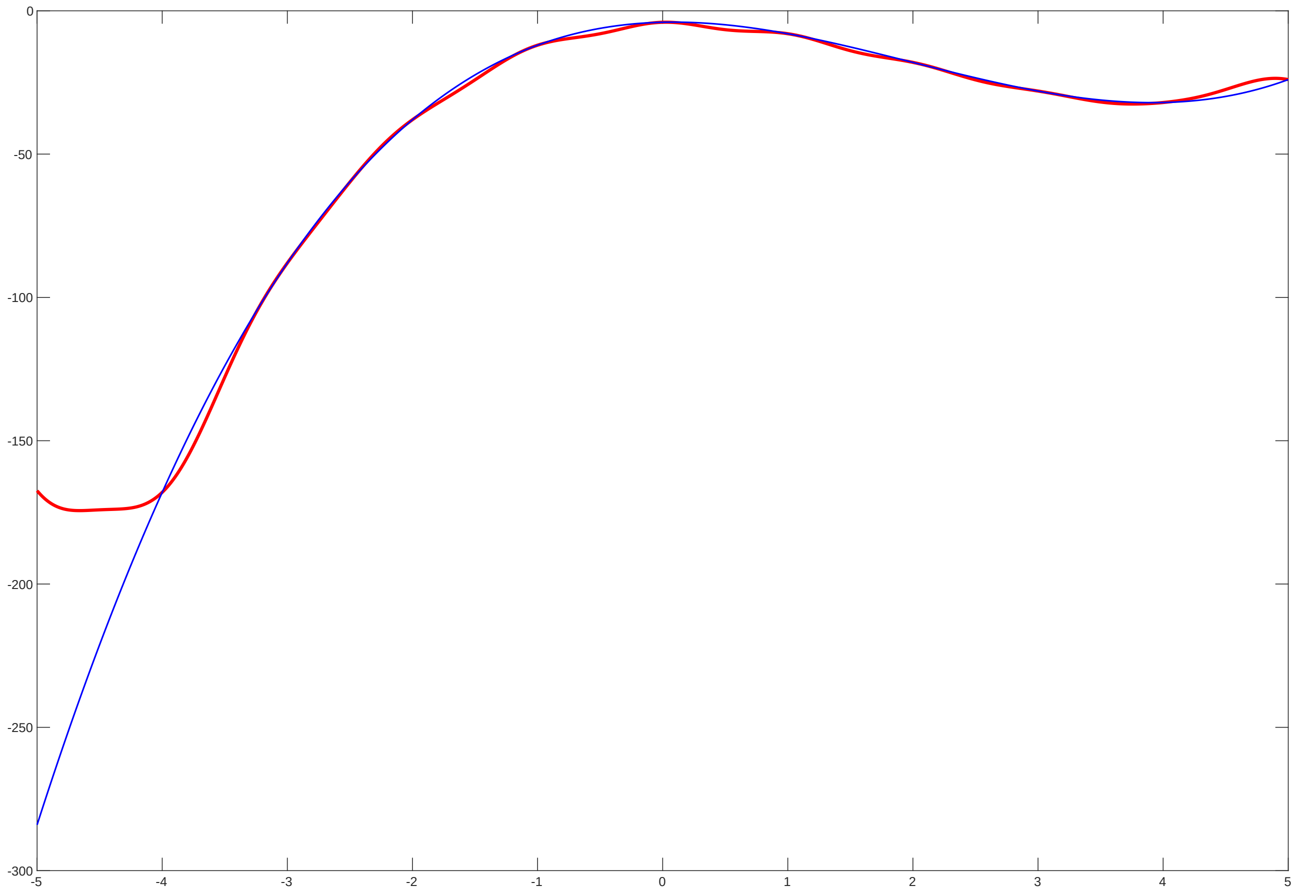
(having specified in `ops` a proper solver, e.g., `QUADPROG`), and the optimal solution can be retrieved by

```
alpha = value( alpha );
b = dual( F( 'simplex' ) );
```

Once this is done, the prediction `v` at a point `x` is computed by

```
v = b;
for i = 1 : n
    v = v + alpha( i ) *  exp( - norm( x - X( i ) )^2 / s2 );
end
```

This could in principle be made a bit more efficient by only using the support vectors, i.e., the indices such that `alpha( i ) > 0`, but in this case all original points should turn out to be support vectors. Doing so for all the points in `-5 : 0.01 : 5` and plotting the results against the original polynomial should produce a picture like the one below:



The SVR thereby produces a good interpolation of the real polynomial, except towards the left endpoint of the interval. By changing the hyperparameters better (or much worse) results may be obtained.

2) Solve the box-constrained quadratic optimization problem

$$\min\left\{ x^TQx/2 + qx \,:\, 0 \leq x \leq u \right\} \quad \text{with data} \quad Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1001 & -999 \\ 0 & -999 & 1001 \end{bmatrix} \quad , \quad q = \begin{bmatrix} -4 \\ -1 \\ -1 \end{bmatrix} \quad , \quad u = \begin{bmatrix} 3 \\ 4 \\ 3 \end{bmatrix}$$

using the projected gradient method (the "standard form projecting over the tangent cone of the active constraints) starting from the point "in the middle of the box" $x^0 = u/2$. Write a `Matlab` code that implements the algorithm and succinctly describe it (the fundamental steps, leaving aside the unnecessary details), then 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 comment on the number of iterations and the convergence rate of the approach.

SOLUTION

Starting from the initial iterate `x = u / 2`, the projected gradient algorithm initially computes the descent direction as the anti-gradient, after which it is then projected over the (tangent cone of the) active constraint. This is especially simple in the box-constrained case in that it boils down to remove any negative component corresponding to an active lower bound constraint and any positive component corresponding to an active upper bound constraint:

```
d = - ( Q * x + q );
d( u - x <= 1e-12 & d > 0 ) = 0;
d( x <= 1e-12 & d < 0 ) = 0;
```

The norm of the projected (anti-)gradient being zero is the stopping condition; in practice one should always use some tolerance, which is here taken as `1e-6`. One can then compute the unconstrained minimum along it with the usual formula

```
alpha = ( - g' * d ) / d' * Q * d;
```

(where in general one should account for the possibility of the denumerator is zero, but again this is not happening in our instance). However, such a step may violate some box constraint, hence one must first compute the maximum step that retains feasibility, i.e., the minimum among the steps that make each constraint active. Again, due to the trivial form of the constraints one should only consider the upper bound constraint $x_i \leq u_i$ if $d_i > 0$ and the lower bound one $x_i \geq 0$ if $d_i < 0$, i.e.,

```
ind = d > 0;
maxt = min( ( u( ind ) - x( ind ) ) ./ d( ind ) );
ind = d < 0;
maxt = min( [ maxt min( - x( ind ) ./ d( ind ) ) ] );
```

Then, the next iterate

```
x = x + min( [ alpha maxt ] * d;
```

is guaranteed to be feasible and the process can be repeated, eventually leading to convergence. With a tolerance of `1e-6` the process should take around 200 iterations and return a solution close to

```
x = [ 3.0000 ; 0.5000 ; 0.5000 ] , v = -8.0000
```

Indeed, this point has zero projected gradient, in that

```
Q * x + q = [ -1.0000 ; 0.0000 ; 0.0000 ]
```

but the constraint $x_1 \leq 3$ is active, which means that the first component of the anti-gradient (1) is zeroed as well from the projection.

For the optional part one can notice that Q is separable: the first variable has no cross-product terms with the second and third. Thus, the optimization is basically separable in the two groups of variables. The eigenvalues of the 2×2 block are 2 and 2000, which gives the convergence rate in the unconstrained case

$$[(2000 - 2)/(2000 + 2)]^2 \approx 0.996$$

Now, the second and third components of the optimal point are in the interior of the feasible region; basically, one should expect the algorithm to run as an unconstrained one for those. Since the initial function value is 122.75 and the final is 8, $a^0 = f(\, x^1\,) - f_* = 114.75$. The expected number of iterations is

$$k \geq [\, 1 \,/\, \log(\, 1 \,/\, r\,)\,] \log(\, a^0 \,/\, \epsilon\,)$$

where ϵ is the desired final error. Using this formula with $\epsilon =$ `1e-6` one obtains an estimate of roughly 4500 iterations, while in practice the algorithm takes much less than that, partly due to a number of approximations done during the analysis: the value of a^0 also takes into account the first component, the stopping condition with tolerance `1e-6` does not directly measure the gap, the algorithm is not exactly the same as what it would be if it ran on the last two components alone, and so on. Besides, the estimate in the formula is worst-case, and faster convergence is always possible.

3) Consider the following unconstrained multiobjective optimization problem:

$$\begin{cases} \min\ (x_1^4 + 2x_2^2,\ -x_1 + x_2^2) \\ (x_1, x_2) \in \mathbb{R}^2 \end{cases}$$

(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.

SOLUTION

(a) It is enough to notice that the objective function f_1 admits a unique global minimum point, which, therefore, is a (Pareto) minimum point for the given problem.

(b) - (c) We preliminarly observe that the problem is convex, since the objective and the constraint functions are convex. Therefore the set of weak minima coincides with the set of solutions of the scalarized problems (P_{α_1}), where $0 \le \alpha_1 \le 1$, i.e.

$$\begin{cases} \min\ \alpha_1(x_1^4 + 2x_2^2) + (1 - \alpha_1)(-x_1 + x_2^2) =: \psi_{\alpha_1}(x) \\ (x_1, x_2) \in \mathbb{R}^2 \end{cases}$$

We note that (P_{α_1}) is convex and differentiable; then the following system provides a necessary and sufficient condition for an optimal solution of (P_{α_1}):

$$\begin{cases} 4\alpha_1 x_1^3 - 1 + \alpha_1 = 0 \\ 2\alpha_1 x_2 + 2x_2 = 0 \\ 0 \le \alpha_1 \le 1, \end{cases}$$

We obtain:

$$\begin{cases} x_1 = \sqrt[3]{\dfrac{1 - \alpha_1}{4\alpha_1}},\ \alpha_1 \neq 0 \\ x_2 = 0 \\ 0 \le \alpha_1 \le 1, \end{cases}$$

Notice that for $\alpha_1 = 0$ the previous system is impossible.

Then, $Weak\ Min(P) = \{(x_1, x_2) : x_1 = \sqrt[3]{\dfrac{1 - \alpha_1}{4\alpha_1}},,\ x_2 = 0, 0 < \alpha_1 \le 1\}$, i.e.,

$$Weak\ Min(P) = \{(x_1, x_2) : x_1 \ge 0,\ x_2 = 0\}.$$

Moreover,

$$Min(P) \supseteq \{(x_1, x_2) : x_1 = \sqrt[3]{\frac{1 - \alpha_1}{4\alpha_1}},\ x_2 = 0, 0 < \alpha_1 < 1\}.$$

Since, for $\alpha_1 = 1$, we obtain the point $(0, 0)$ which is a minimum point, as we have already observed, in conclusion:

$$Weak\ Min(P) = Min(P) = \{(x_1, x_2) : x_1 \ge 0,\ x_2 = 0\}.$$

4) Consider the following bimatrix game:

$$C_1 = \begin{pmatrix} 4 & 3 & 2 \\ 2 & 1 & 5 \end{pmatrix} \qquad C_2 = \begin{pmatrix} 5 & 4 & 3 \\ 7 & 2 & 6 \end{pmatrix}$$

(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 3, so that column 1 in the two matrices can be deleted. The reduced game is given by the matrices

$$C_1^R = \begin{pmatrix} 3 & 2 \\ 1 & 5 \end{pmatrix} \qquad C_2^R = \begin{pmatrix} 4 & 3 \\ 2 & 6 \end{pmatrix}$$

Now, it is simple to show that (2,2) and (1,3) are pure strategies Nash equilibria. Indeed, the minima on the columns of C_1^R, (i.e., 1 and 2), are obtained in correspondence of the minima on the rows of C_2^R, (i.e., 2 and 3) and are related to the components $(C_1)_{22}$ $(C_1)_{13}$ of the given matrices C_1 and C_2.

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

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

Then, the best response mapping associated with $P_1(y_2)$ is:

$$B_1(y_2) = \begin{cases} 0 & \text{if } y_2 \in (3/5, 1] \\ [0,1] & \text{if } y_2 = 3/5 \\ 1 & \text{if } y_2 \in [0, 3/5) \end{cases}$$

Similarly, the optimization problem associated with Player 2 is

$$\begin{cases} \min\ x^T C_2^R y = x_1(4y_2 + 3y_3) + x_2(2y_2 + 6y_3) \\ y_1 + y_2 = 1 \\ y_1, y_2 \geq 0 \end{cases} \equiv \begin{cases} \min\ (5x_1 - 4)y_2 - 3x_1 + 6 \\ 0 \leq y_2 \leq 1 \end{cases} \qquad (P_2(x_1))$$

Then, the best response mapping associated with $P_2(x_1)$ is:

$$B_2(x_1) = \begin{cases} 0 & \text{if } x_1 \in (4/5, 1] \\ [0,1] & \text{if } x_1 = 4/5 \\ 1 & \text{if } x_1 \in [0, 4/5) \end{cases}$$

The couples (x_1, y_2) such that $x_1 \in B_1(y_2)$ and $y_2 \in B_2(x_1)$ are

1. $x_1 = 0, y_2 = 1$,
2. $x_1 = \frac{4}{5}, y_2 = \frac{3}{5}$,
3. $x_1 = 1, y_2 = 0$,

so that, recalling that $y_1 = 0$,

- $(x_1, x_2) = (0, 1)$, $\quad (y_1, y_2, y_3) = (0, 1, 0)$, is a pure strategies Nash equilibrium,
- $(x_1, x_2) = (\frac{4}{5}, \frac{1}{5})$, $\quad (y_1, y_2, y_3) = (0, \frac{3}{5}, \frac{2}{5})$, is a mixed strategies Nash equilibrium,
- $(x_1, x_2) = (1, 0)$, $\quad (y_1, y_2, y_3) = (0, 0, 1)$, is a pure strategies Nash equilibrium.

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 $c_p \in \mathbb{R}^2$, $p = 1, \dots, 4$ that solve

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

or equivalently, with the introduction of binary variables,

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

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 $k \in \{1, \dots, 4\}^{12}$ 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

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

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

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

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

$$
\begin{array}{c|cccc}
\multirow{2}{*}{c} & 0.8 & 0.9 & 0.1 & 1.0 \\
 & 1.0 & 0.5 & 0.8 & 0.1
\end{array}
$$

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

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

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 $\lambda^+ \geq 0$ and $\lambda^- \geq 0$, $\psi(\lambda^+, \lambda^-) \leq \nu(P)$, which immediately leads to the definition of the Lagrangian Dual

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

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:

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

(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.

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

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

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

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

(P_{α_1}):

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

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 $0 < \alpha_1 \le 1$, $x(\alpha_1) = (0, -1, 1)$ 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

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

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

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

(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

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

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:

$$B_1(y_3) = \left\{ \begin{array}{cl} 0 & \text{if } y_3 \in [0, 1/2) \\ [0,1] & \text{if } y_3 = 1/2 \\ 1 & \text{if } y_3 \in (1/2, 1) \end{array} \right.$$

Similarly, the optimization problem associated with Player 2 is

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

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

$$B_2(x_2) = \left\{ \begin{array}{cl} 0 & \text{if } x_2 \in (1/5, 1] \\ [0,1] & \text{if } x_2 = 1/5 \\ 1 & \text{if } x_2 \in [0, 1/5) \end{array} \right.$$

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 $y_1 = 0$,

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

1) Consider the unconstrained optimization problem

$$\begin{cases} \min\ 3x_1^2 + x_2^2 + 2x_3^2 + 4x_4^2 + x_1x_2 + 2x_1x_4 + 2x_3x_4 - x_1 + 8x_2 + 6x_3 + 9x_4 \\ x \in \mathbb{R}^4 \end{cases}$$

(a) Apply the conjugate gradient method with starting point $x^0 = (0,0,0,0)$ and using $\|\nabla f(x)\| < 10^{-6}$ as stopping criterion. How many iterations are needed by the algorithm? Write the vector found at each iteration.

(b) Is the obtained solution a global minimum of the given problem? Justify the answer.

SOLUTION

(a) The objective function $f(x)$ is quadratic, i.e., $f(x) = (1/2)x^TQx + c^Tx$ with

$$Q = \begin{pmatrix} 6 & 1 & 0 & 2 \\ 1 & 2 & 0 & 0 \\ 0 & 0 & 4 & 2 \\ 2 & 0 & 2 & 8 \end{pmatrix} \qquad c^T = (-1, 8, 6, 9)$$

Matlab solution

```
Q = [6 1 0 2;1 2 0 0;0 0 4 2;2 0 2 8];
c = [-1 8 6 9]';

disp('Eigenvalues of Q:')
eig(Q)

x0 = [0,0,0,0]';        % Starting point
tolerance = 10^(-6);    % Parameters

x = x0;
X = [];
for ITER = 1:10
    v = 0.5*x'*Q*x + c'*x;
    g = Q*x + c ;

    X=[X;ITER,x',v];
    if norm(g) < tolerance   % stopping criterion
       break
    end

    % search direction
    if ITER == 1
       d = -g;
    else
       beta = (g'*Q*d_prev)/(d_prev'*Q*d_prev);
       d = -g + beta*d_prev;
    end

    t = (-g'*d)/(d'*Q*d);    % step size
    x = x + t*d;             % new point
    d_prev = d;
end
 X
 disp('Gradient norm:')
 norm(g)
```

We obtain the following solution:

```
Eigenvalues of Q:
ans =
    1.7070
    3.0000
    5.5077
    9.7853
```

```
X =
    1.0000         0         0         0         0         0
    2.0000    0.1670   -1.3358   -1.0018   -1.5028  -15.1945
    3.0000    1.6166   -3.9878   -1.4191   -1.1575  -26.2254
    4.0000    1.2719   -4.6713   -1.0360   -1.1446  -27.5798
    5.0000    1.3623   -4.6812   -0.8768   -1.2464  -27.6449
```

In particular, the gradient norm evaluated at the final point is:

```
ans =

   4.4409e-16
```

The effective iterations of the algorithm are 4, since in the first one we have considered the initial point x^0.

(b) The found point $x = (1.3623 - 4.6812 - 0.8768 - 1.2464)$ is a global minimum since the objective function is strongly convex: in fact the eigenvalues of the Hessian of f are all strictly positive.

2) Consider a binary classification problem with the data sets A and B given by the row vectors of the matrices:

$$A = \begin{pmatrix} 6.55 & 0.85 \\ 6.55 & 1.71 \\ 7.06 & 0.31 \\ 2.76 & 0.46 \\ 0.97 & 8.23 \\ 9.5 & 0.34 \\ 4.38 & 3.81 \\ 1.86 & 4.89 \\ 2.76 & 6.79 \\ 6.55 & 1.22 \end{pmatrix}, \quad B = \begin{pmatrix} 9.59 & 3.40 \\ 5.85 & 2.23 \\ 7.51 & 2.55 \\ 5.05 & 7 \\ 8.9 & 9.59 \\ 8.40 & 2.54 \\ 8.14 & 2.43 \\ 9.3 & 3.45 \\ 6.16 & 4.73 \\ 3.51 & 8.30 \end{pmatrix}$$

(a) Write the linear SVM model with soft margin to find the separating hyperplane;

(b) Solve the dual problem with parameter $C = 10$ and find the optimal hyperplane. Write explicitly the vector of the optimal solution of the dual problem;

(c) Find the misclassified points of the data sets A and B by means of the dual solution;

(d) Classify the new point $(5, 5)$.

SOLUTION

(a) Let $\ell = 20$, $(x^i)^T$ be the i-th row of the matrix A, $i = 1, .., 10$ and of the matrix B for $i = 11, .., 20$. For any point x^i, define the label:

$$y^i = \begin{cases} 1 & \text{if } i = 1, .., 10 \\ -1 & \text{if } i = 11, .., 20 \end{cases}$$

Given $C > 0$, the formulation of the linear SVM with soft margin is

$$\begin{cases} \min\limits_{w,b,\xi} \dfrac{1}{2}\|w\|^2 + C \sum\limits_{i=1}^{\ell} \xi_i \\ 1 - y^i(w^T x^i + b) \le \xi_i, & \forall\, i = 1, \dots, \ell \\ \xi_i \ge 0, & \forall\, i = 1, \dots, \ell \end{cases} \tag{1}$$

(b) The dual problem of (1) is

$$\begin{cases} \max\limits_{\lambda} \; -\dfrac{1}{2} \sum\limits_{i=1}^{\ell} \sum\limits_{j=1}^{\ell} y^i y^j (x^i)^T x^j \lambda_i \lambda_j + \sum\limits_{i=1}^{\ell} \lambda_i \\ \sum\limits_{i=1}^{\ell} \lambda_i y^i = 0 \\ 0 \le \lambda_i \le C \qquad i = 1, \dots, \ell \end{cases}$$

Matlab solution

```
A = [.......]; B = [.......]; C=10; nA = size(A,1); nB = size(B,1);

T = [A ; B];                          % training points
y = [ones(nA,1) ; -ones(nB,1)];       % labels
l = length(y);
Q = zeros(l,l);

for i = 1 : l
    for j = 1 : l
        Q(i,j) = y(i)*y(j)*(T(i,:))*T(j,:)';  % (minus) Dual Hessian
    end
end

% solve the problem
la = quadprog(Q,-ones(l,1),[ ],[ ],y',0,zeros(l,1),C*ones(l,1));

w = zeros(2,1);                     % compute vector w
for i = 1 : l
    w = w + la(i)*y(i)*T(i,:)';
end
```

```
indpos = find(la > 0.001) ;       % compute scalar b
ind = find(la(indpos) < C - 10^(-3));
  i = indpos(ind(1)) ;
 b = 1/y(i) - w'*T(i,:)'
```

We obtain the dual optimal solution:

```
la =

    0.0000
    6.5691
    0.0000
    0.0000
    0.0000
   10.0000
    0.0000
    0.0000
    7.8340
    0.0000
    0.0000
   10.0000
   10.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    4.4031
    0.0000


w =
   -1.0735
   -0.8009
b =
    9.4008
```

The optimal hyperplane has equation $w^T x + b = -1.0735x_1 - 0.8009x_2 + 9.4008 = 0$.

(b) Consider the dual optimal solution λ^* and denote by (w^*, b^*, ξ^*) an optimal solution of (1). By the complementary slackness conditions,

$$\begin{cases} \lambda_i^* \left[1 - y^i((w^*)^T x^i + b^*) - \xi_i^*\right] = 0 \\ (C - \lambda_i^*)\xi_i^* = 0 \end{cases} \tag{2}$$

it follows that a necessary condition for a point x^i to be misclassified is that $\lambda_i^* = C = 10$. We find that $\lambda_i^* = 10$, for $i = 6, 12, 13$, which correspond to the points

$$x^6 = (9.5, 0.34) \in A, \quad x^{12} = (5.85, 2.23) \in B, \quad x^{13} = (7.51, 2.55) \in B$$

The first two points are misclassified, being $w^T x^6 + b < 0$, $w^T x^{12} + b > 0$. Note that x^{13} is not misclassified being $w^T x^{13} + b < 0$, in fact in this case, even though $\lambda_{13}^* > 0$, we have that the error $\xi_{13}^* < 1$.

(c) The new point $\bar{x}^T = (5, 5)$ is labeled 1, since $w^T \bar{x} + b = 0.029 > 0$.

3) Consider the following multiobjective optimization problem:

$$\begin{cases} \min\ (x_1^2,\ -x_1 + x_2^2) \\ -x_1 - x_2 + 1 \leq 0 \end{cases}$$

(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.

SOLUTION

(a) We preliminarly observe that the problem is convex, since the objective and the constraint functions are convex. Therefore the set of weak minima coincides with the set of solutions of the scalarized problems (P_{α_1}), where $0 \leq \alpha_1 \leq 1$, i.e.

$$\begin{cases} \min\ \alpha_1 x_1^2 + (1-\alpha_1)(-x_1 + x_2^2) =: \psi_{\alpha_1}(x) \\ -x_1 - x_2 + 1 \leq 0 \end{cases}$$

with $0 \leq \alpha_1 \leq 1$ and for every $0 < \alpha_1 < 1$ we obtain a Pareto minimum. Note that in this last case the objective function $\psi_{\alpha_1}(x)$ of P_{α_1} is strongly convex so that P_{α_1} admits a unique global solution for every $0 < \alpha_1 < 1$, which is a Pareto minimum.

(b) - (c) We note that for $\alpha_1 = 0$, the problem P_{α_1} is unbounded, indeed, at any feasible point $(x_1, 0)$, $d = (1, 0)$ is a descent direction which is also a recession direction for the polyhedron K.

P_{α_1} can be solved by Matlab for $0 < \alpha_1 \leq 1$:

```
Q1 = [2 0; 0 0];
Q2 = [0 0; 0 2] ;
c1=[0 0]';
c2=[-1 0]';

A =[ -1 -1];
b = -1;

% solve the scalarized problem with  alfa1 in (0,1]

MINIMA=[ ]; LAMBDA=[ ]; % First column: value of alfa1

for alfa1 = 0.01 : 0.001 : 1
[x,fval,exitflag,output,lambda] = quadprog(alfa1*Q1+(1-alfa1)*Q2,alfa1*c1+(1-alfa1)*c2,A,b) ;
MINIMA=[MINIMA; alfa1 x'];
LAMBDA=[LAMBDA;alfa1,lambda.ineqlin'];
end

  plot(MINIMA(:,2),MINIMA(:,3), 'r*')
```

- For $0 < \alpha_1 \leq 1/3$, we obtain the set of points $\{(x_1, 0), x_1 \geq 1\}$ which are Pareto minima.
- For $1/3 < \alpha_1 < 1$, we obtain the set of points $\{(x_1, x_2), x_1 + x_2 = 1, 0 < x_1 < 1\}$ which are Pareto minima.
- For $\alpha_1 = 1$, we obtain the set of points $\{(0, x_2), x_2 \geq 1\}$ which are Weak Pareto minima.

We note that the previous solutions can also be obtained by solving the KKT conditions for (P_{α_1}) which is convex,differentiable and fulfils the Abadie constraints qualifications. Therefore, the following system provides a necessary and sufficient condition for an optimal solution of (P_{α_1}):

$$\begin{cases} 2\alpha_1 x_1 - 1 + \alpha_1 - \lambda = 0 \\ 2(1-\alpha_1)x_2 - \lambda = 0 \\ \lambda(-x_1 - x_2 + 1) = 0 \\ -x_1 - x_2 + 1 \leq 0, \lambda \geq 0 \\ 0 \leq \alpha_1 \leq 1, \end{cases}$$

In particular, notice that for $\alpha_1 = 0$ the previous system is impossible, which agrees with the fact that (P_{α_1}) has no solutions for $\alpha_1 = 0$, as already observed.

For $\alpha_1 = 1$, the system becomes:

$$\begin{cases} 2x_1 = \lambda \\ \lambda = 0 \\ -x_2 + 1 \leq 0, \end{cases}$$

and we obtain the set of points $\{(0, x_2), x_2 \geq 1\}$ which are Weak Pareto minima.

For $0 < \alpha_1 < 1$, the system becomes:

$$\begin{cases} x_1 = \frac{1-\alpha_1+\lambda}{2\alpha_1} \\ x_2 = \frac{\lambda}{2(1-\alpha_1)} \\ \lambda(-x_1 - x_2 + 1) = 0 \\ -x_1 - x_2 + 1 \leq 0, \lambda \geq 0 \\ 0 < \alpha_1 < 1, \end{cases} \tag{3}$$

For solving (3) we distinguish the cases $I)$ $\lambda = 0$ and $II)$ $\lambda > 0$. In case $I)$, (3) becomes:

$$\begin{cases} x_1 = \frac{1-\alpha_1}{2\alpha_1} \\ x_2 = 0 \\ -x_1 + 1 \leq 0 \\ 0 < \alpha_1 < 1, \end{cases}$$

which leads to the set of minimum points: $\{(x_1, 0), x_1 \geq 1\}$, with $0 < \alpha_1 \leq 1/3$.

In case $II)$,(3) becomes:

$$\begin{cases} x_1 = \frac{1-\alpha_1+\lambda}{2\alpha_1} \\ x_2 = \frac{\lambda}{2(1-\alpha_1)} \\ -x_1 - x_2 + 1 = 0 \\ \lambda > 0 \\ 0 < \alpha_1 < 1, \end{cases}$$

with solutions:

$$\begin{cases} x_1 = \frac{3(1-\alpha_1)}{2} \\ x_2 = \frac{3\alpha_1-1}{2} \\ \lambda = (3\alpha_1 - 1)(1 - \alpha_1) \\ \lambda > 0 \\ 0 < \alpha_1 < 1, \end{cases}$$

which leads to the set of minimum points: $\{(x_1, x_2), x_1 + x_2 = 1, 0 < x_1 < 1\}$, with $1/3 < \alpha_1 < 1$.

4) Consider the following bimatrix game:

$$C_1 = \begin{pmatrix} 2 & 0 & 2 \\ 1 & 3 & -1 \end{pmatrix} \qquad C_2 = \begin{pmatrix} 3 & 1 & 2 \\ 2 & 4 & 1 \end{pmatrix}$$

(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 3, so that column 1 in the two matrices can be deleted. The reduced game is given by the matrices

$$C_1^R = \begin{pmatrix} 0 & 2 \\ 3 & -1 \end{pmatrix} \qquad C_2^R = \begin{pmatrix} 1 & 2 \\ 4 & 1 \end{pmatrix}$$

Now, it is simple to show that (1,2) and (2,3) are pure strategies Nash equilibria. Indeed, the minima on the columns of C_1^R, (i.e., 0 and -1), are obtained in correspondence of the minima on the rows of C_2^R, (i.e., 1 and 1) and are related to the components $(1,2)$ $(2,3)$ of the given matrices C_1 and C_2.

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

$$\begin{cases} \min\ x^T C_1^R y = 3x_2y_2 + (2x_1 - x_2)y_3 \\ x_1 + x_2 = 1 \\ x_1, x_2 \geq 0 \end{cases} \equiv \begin{cases} \min\ (6y_2 - 3)x_2 - 2y_2 + 2 \\ 0 \leq x_1 \leq 1 \end{cases} \qquad (P_1(y_2))$$

Then, the best response mapping associated with $P_1(y_2)$ is:

$$B_1(y_2) = \begin{cases} 0 & \text{if } y_2 \in (1/2, 1] \\ [0,1] & \text{if } y_2 = 1/2 \\ 1 & \text{if } y_2 \in [0, 1/2) \end{cases}$$

Similarly, the optimization problem associated with Player 2 is

$$\begin{cases} \min\ x^T C_2^R y = (x_1 + 4x_2)y_2 + (2x_1 + x_2)y_3 \\ y_2 + y_3 = 1 \\ y_2, y_3 \geq 0 \end{cases} \equiv \begin{cases} \min\ (4x_2 - 1)y_2 + 2 - x_2 \\ 0 \leq y_2 \leq 1 \end{cases} \qquad (P_2(x_2))$$

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

$$B_2(x_2) = \begin{cases} 0 & \text{if } x_2 \in (1/4, 1] \\ [0,1] & \text{if } x_1 = 1/4 \\ 1 & \text{if } x_1 \in [0, 1/4) \end{cases}$$

The couples (x_2, y_2) such that $x_2 \in B_1(y_2)$ and $y_2 \in B_2(x_2)$ are

1. $x_2 = 0, y_2 = 1,$
2. $x_2 = \frac{1}{4}, y_2 = \frac{1}{2},$
3. $x_2 = 1, y_2 = 0,$

so that, recalling that $y_1 = 0$,

- $(x_1, x_2) = (0, 1)$, $\quad (y_1, y_2, y_3) = (0, 0, 1)$, is a pure strategies Nash equilibrium,
- $(x_1, x_2) = (\frac{3}{4}, \frac{1}{4})$, $\quad (y_1, y_2, y_3) = (0, \frac{1}{2}, \frac{1}{2})$, is a mixed strategies Nash equilibrium,
- $(x_1, x_2) = (1, 0)$, $\quad (y_1, y_2, y_3) = (0, 1, 0)$, is a pure strategies Nash equilibrium.

1) Consider the unconstrained optimization problem

$$\begin{cases} \min \ 2x_1^2 + x_2^2 - x_1x_2 + e^{x_1+2x_2} \\ x \in \mathbb{R}^2 \end{cases}$$

(a) Prove that the problem admits a global minimum;

(b) Apply the gradient method with an inexact line search, setting $\bar{t} = 1$,$\alpha = 0.1$, $\gamma = 0.8$, with starting point $x^0 = (-10, 8)$ and using $\|\nabla f(x)\| < 10^{-6}$ as stopping criterion. How many iterations are needed by the algorithm? Write explicitly the vectors found at the last three iterations.

(c) Is the obtained solution a global minimum of the given problem? Justify the answer.

SOLUTION

(a) The objective function $f(x) = f_1(x) + f_2(x)$, where $f_1 = 2x_1^2 + x_2^2 - x_1x_2$ is strongly convex and $f_2(x) = e^{x_1+2x_2}$ is convex, being $f_2 = \psi \circ g$, with $\psi(y) = e^y$ convex and $g(x) = x_1 + 2x_2$ linear. Therefore $f_1 + f_2$ is strongly convex being the sum of a strongly convex plus a convex function. Consequently, f admits a unique global minimum point.

(b) We notice that

$$\nabla f(x_1, x_2) = \begin{pmatrix} 4x_1 - x_2 + e^{x_1+2x_2} \\ -x_1 + 2x_2 + 2e^{x_1+2x_2} \end{pmatrix}$$

Matlab solution

```
%% Data

alpha = 0.1;
gamma = 0.8;
tbar = 1;
x0 = [-10;8];
tolerance = 10^(-6) ;

X=[ ];

ITER = 0 ;
x = x0;

while true
    [v, g] = f(x);

    X=[X;ITER,x(1),x(2),v,norm(g)];

    % stopping criterion
    if norm(g) < tolerance
        break
    end

    d = -g;                           % search direction


     t = tbar ;                        % Armijo inexact line search
    while f(x+t*d) > v + alpha*g'*d*t
        t = gamma*t ;
    end


    x = x + t*d ;                     % new point
    ITER = ITER + 1 ;
    end

disp('optimal solution')
x
v
norm(g)
ITER
```

```
function [v, g] = f(x)

v = 2*x(1)^2 + x(2)^2 - x(1)*x(2) + exp(x(1)+2*x(2)) ;

g = [4*x(1)-x(2)+exp(x(1)+2*x(2));
      -x(1)+2*x(2)+2*exp(x(1)+2*x(2))];

end
```

We obtain the following solution:

```
x =

   -0.1952
   -0.4393


v =

    0.5251


ans =

   6.6986e-07


ITER =

    24
```

In particular, the gradient norm evaluated at the final point is:

```
ans =

   6.6986e-07
```

The iterations of the algorithm are 24.

The vectors found at the last three iterations are:

```
-0.1952   -0.4393
-0.1953   -0.4393
-0.1952   -0.4393
```

(b) The found point $x = (-0.1952, -0.4393)$ is a global minimum since the objective function is strongly convex as shown in point (a).

2) Consider a regression problem with the following data set where the points $(x_i, y_i), i = 1, 31$, are given by the row vectors of the matrices:

$$\begin{pmatrix} -3.0000 & 4.58 \\ -2.8000 & 7.19 \\ -2.6000 & 8.22 \\ -2.4000 & 16.06 \\ -2.2000 & 16.42 \\ -2.0000 & 17.53 \\ -1.8000 & 11.48 \\ -1.6000 & 14.10 \\ -1.4000 & 16.82 \\ -1.2000 & 16.15 \\ -1.0000 & 11.68 \\ -0.8000 & 6.00 \\ -0.6000 & 7.82 \\ -0.4000 & 2.82 \\ -0.2000 & 2.71 \\ 0 & 1.16 \end{pmatrix} \qquad \begin{pmatrix} 0.2000 & -1.42 \\ 0.4000 & -3.84 \\ 0.6000 & -4.71 \\ 0.8000 & -8.15 \\ 1.0000 & -7.33 \\ 1.2000 & -13.64 \\ 1.4000 & -15.26 \\ 1.6000 & -14.87 \\ 1.8000 & -9.92 \\ 2.0000 & -10.50 \\ 2.2000 & -7.72 \\ 2.4000 & -11.78 \\ 2.6000 & -10.26 \\ 2.8000 & -7.13 \\ 3.0000 & -2.11 \end{pmatrix}$$

(a) Write the dual formulation of a nonlinear ε-SV regression model with $C = 5$, $\varepsilon = 3$ and a polynomial kernel $k(x, y) := (x^T y + 1)^4$;

(b) Solve the problem in (a) and find the regression function;

(c) Find the support vectors;

(d) Find the points of the data set that are outside the ε-tube, by making use of the dual solution.

SOLUTION

(a) Let $\ell = 31$, $(x_i, y_i))$, $i = 1, .., \ell$ be the i-th element of the data set, $C = 5$, $\varepsilon = 3$, $k(x, y) := (x^T y + 1)^4$. The dual formulation of a nonlinear ε-SV regression model is

$$\begin{cases} \max\limits_{(\lambda^+,\lambda^-)} \ -\dfrac{1}{2} \sum\limits_{i=1}^{\ell} \sum\limits_{j=1}^{\ell} (\lambda_i^+ - \lambda_i^-)(\lambda_j^+ - \lambda_j^-) k(x_i, x_j) \\ \qquad\qquad -\varepsilon \sum\limits_{i=1}^{\ell} (\lambda_i^+ + \lambda_i^-) + \sum\limits_{i=1}^{\ell} y_i (\lambda_i^+ - \lambda_i^-) \\ \sum\limits_{i=1}^{\ell} (\lambda_i^+ - \lambda_i^-) = 0 \\ \lambda_i^+, \lambda_i^- \in [0, C], i = 1, .., \ell \end{cases} \tag{1}$$

(b) **Matlab solution**

```
data = [
   -3.0000    4.58
   -2.8000     7.19
   -2.6000     8.22
   -2.4000    16.06
   -2.2000    16.42
   -2.0000    17.53
   -1.8000    11.48
   -1.6000    14.10
   -1.4000    16.82
   -1.2000    16.15
   -1.0000    11.68
   -0.8000     6.00
   -0.6000     7.82
   -0.4000     2.82
   -0.2000     2.71
         0     1.16
    0.2000    -1.42
    0.4000    -3.84
    0.6000    -4.71
    0.8000    -8.15
    1.0000    -7.33
    1.2000   -13.64
    1.4000   -15.26
    1.6000   -14.87
```

```
    1.8000   -9.92
    2.0000  -10.50
    2.2000   -7.72
    2.4000  -11.78
    2.6000  -10.26
    2.8000   -7.13
    3.0000   -2.11
        ];

x = data(:,1) ;
y = data(:,2) ;
l = length(x) ;

epsilon = 3 ;
C = 5;

X = zeros(l,l);
for i = 1 : l
    for j = 1 : l
        X(i,j) = kernel(x(i),x(j)) ;
    end
end
Q = [ X -X ; -X X ];
c = epsilon*ones(2*l,1) + [-y;y];

sol = quadprog(Q,c,[],[],[ones(1,l) -ones(1,l)],0,zeros(2*l,1),C*ones(2*l,1));
lap = sol(1:l);
lam = sol(l+1:2*l);

% compute b
ind = find(lap > 1e-3 & lap < C-1e-3);
if isempty(ind)==0
    i = ind(1);
    b = y(i) - epsilon;
    for j = 1 : l
        b = b - (lap(j)-lam(j))*kernel(x(i),x(j));
    end
else
    ind = find(lam > 1e-3 & lam < C-1e-3);
    i = ind(1);
    b = y(i) + epsilon ;
    for j = 1 : l
        b = b - (lap(j)-lam(j))*kernel(x(i),x(j));
    end
end


z = zeros(l,1);                                              % find regression and epsilon-tube
for i = 1 : l
    z(i) = b ;
    for j = 1 : l
        z(i) = z(i) + (lap(j)-lam(j))*kernel(x(i),x(j));
    end
end
zp = z + epsilon ;
zm = z - epsilon ;

sv = [find(lap > 1e-3);find(lam > 1e-3)];                     % find support vectors
sv = sort(sv);

plot(x,y,'b.',x(sv),y(sv),'ro',x,z,'k-',x,zp,'r-',x,zm,'r-');     % plot the solution

    disp('Support vectors')
```

```
[sv,x(sv),y(sv),lam(sv),lap(sv)]    % Indexes of support vectors, support vectors, lambda_-,lambda_+

function v = kernel(x,y)
p = 4 ;
v = (x'*y + 1)^p;
end
```

Let λ_- and λ_+ be the vectors given by the Matlab solutions lam, lap. In particular we find, $b = -0.008$.

The regression function is:

$$f(x) = \sum_{i=1}^{\ell}(\lambda_i^+ - \lambda_i^-)k(x_i, x) + b = \sum_{i=1}^{31}(\lambda_i^+ - \lambda_i^-)(x_i x + 1)^4 - 0.008$$

(c) We obtain the support vectors (columns 2-3) and corresponding λ_- and λ_+ (columns 4-5) :

```
ans =

    4.0000   -2.4000   16.0600    0.0000    1.2569
    7.0000   -1.8000   11.4800    5.0000    0.0000
    9.0000   -1.4000   16.8200    0.0000    1.3176
   10.0000   -1.2000   16.1500    0.0000    5.0000
   12.0000   -0.8000    6.0000    0.5720    0.0000
   23.0000    1.4000  -15.2600    4.7410    0.0000
   27.0000    2.2000   -7.7200    0.0000    5.0000
   29.0000    2.6000  -10.2600    2.2616    0.0000
```

(d) Consider the feasibility condition of the primal formulation of the regression problem:

$$y_i - f(x_i) - \varepsilon - \xi_i^+ \leq 0, \quad y_i - f(x_i) + \varepsilon + \xi_i^- \geq 0, \quad i = 1, .., \ell$$

If a point (x_i, y_i) is outside the ε-tube then $\xi_i^+ > 0$ or $\xi_i^- > 0$.

Given the dual optimal solution (λ_+, λ_-) of (1), we can find the errors ξ_i^+ and ξ_i^- associated with the point (x_i, y_i) by the complementarity conditions:

$$\begin{cases} \lambda_i^+ \left[y_i - f(x_i) - \varepsilon - \xi_i^+\right] = 0, & i = 1, .., \ell \\ \lambda_i^- \left[y_i - f(x_i) + \varepsilon + \xi_i^-\right] = 0, & i = 1, .., \ell \\ \xi_i^+(C - \lambda_i^+) = 0, & i = 1, .., \ell \\ \xi_i^-(C - \lambda_i^-) = 0, & i = 1, .., \ell \end{cases} \tag{2}$$

it follows that a necessary condition for a point (x_i, y_i) to be outside the ε-tube is that $\lambda_i^+ = C = 5$ or $\lambda_i^- = C = 5$. We find that $\lambda_i^- = 5$, for $i = 7$, $\lambda_i^+ = 5$, for $i = 10, 27$ which correspond to the points

$$(x_7, y_7) = (-1.8, 11.48), \quad (x_{10}, y_{10}) = (-1.2, 16.15), \quad (x_{27}, y_{27}) = (2.2, -7.72)$$

3) Consider the following unconstrained multiobjective optimization problem:

$$\begin{cases} \min \; f(x_1, x_2) = (x_1 x_2, \; x_1^2 + 2x_2^2 - 2x_2) \\ x \in \mathbb{R}^2 \end{cases}$$

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

(b) Find a suitable subset of Pareto minima, by means of the scalarization method.

SOLUTION

(a) We observe that the function f_2 is strongly convex, so that its minimum point is a Pareto minimum of the given problem.

(b) Consider the scalarized problem (P_{α_1}), where $0 \leq \alpha_1 \leq 1$, i.e.

$$\begin{cases} \min \; \alpha_1 x_1 x_2 + (1-\alpha_1)(x_1^2 + 2x_2^2 - 2x_2) =: \psi_{\alpha_1}(x) \\ x \in \mathbb{R}^2 \end{cases}$$

We note that the objective function $\psi_{\alpha_1}(x)$ is quadratic with Hessian given by

$$Q = \begin{pmatrix} 2(1-\alpha_1) & \alpha_1 \\ \alpha_1 & 4(1-\alpha_1) \end{pmatrix}$$

We observe that Q is positive definite for $0 \leq \alpha_1 < \frac{8-2\sqrt{2}}{7} \approx 0.7388$ so that for such values P_{α_1} is strongly convex and admits a unique global solution which is a Pareto minimum.

P_{α_1} can be solved by Matlab for $0 \leq \alpha_1 \leq 1$:

```
Q1 = [0 1; 1 0];
Q2 = [2 0; 0 4] ;
c1=[0 0]';
c2=[0 -2]';


MINIMA=[ ]; SOL=[ ];

for alfa1 = 0 : 0.001 : 1
eigQalfa1= eig(alfa1*Q1+(1-alfa1)*Q2);
eigQ=[eigQ;alfa1,eigQalfa1']; % Eigenvalues of the hessian of \psi_alfa1(x)

if (eigQ > 0.001)
[x,fval,exitflag,exitflag] = fminunc(@(x) 0.5*x'*(alfa1*Q1+(1-alfa1)*Q2)*x  +(alfa1*c1+(1-alfa1)*c2)'*x, [0,0]'

MINIMA=[MINIMA; alfa1 x'];

else

[x,fval,exitflag,exitflag] = fminunc(@(x) 0.5*x'(alfa1*Q1+(1-alfa1)*Q2)*x  +(alfa1*c1+(1-alfa1)*c2)'*x, [0,0]')
SOL=[SOL; alfa1 x'];
end
end

  plot(MINIMA(:,2),MINIMA(:,3), 'r*')
```

For every $0 \leq \alpha_1 \leq 0.738$, the problem admits a unique solution which is a Pareto minimum (MINIMA(:,2:3)). In particular,

- For $\alpha_1 = 0$ we obtain the point $(0, 0.5)$
- For $\alpha_1 = 0.736$ we obtain the point $(-24.4839, 17.5645)$;
- For $\alpha_1 = 0.737$ we obtain $(-38.0695, 27.1704)$
- For $\alpha_1 = 0.738$ we obtain $(-85.7834, 60.9085)$
- For $\alpha_1 \geq 0.739$ the problem P_{α_1} is unbounded.

The previous values lead us to conjecture that the sequence of minima diverges as $\alpha_1 \to \frac{8-2\sqrt{2}}{7} \approx 0.7388$

We note that the previous solutions can also be obtained by the stationarity conditions for (P_{α_1}) which are necessary for a weak minimum point. Then, the following system provides a necessary condition for an optimal solution of (P_{α_1}) and such condition is also sufficient for $0 \le \alpha_1 < \frac{8-2\sqrt{2}}{7}$:

$$\begin{cases} \alpha_1 x_2 + 2(1-\alpha_1)x_1 = 0 \\ \alpha_1 x_1 + (1-\alpha_1)(4x_2 - 2) = 0 \\ 0 \le \alpha_1 \le 1, \end{cases}$$

We obtain:

$$\begin{cases} x_1 = \dfrac{2\alpha_1(1-\alpha_1)}{-7\alpha_1^2 + 16\alpha_1 - 8}, \\ x_2 = \dfrac{4(1-\alpha_1)^2}{7\alpha_1^2 - 16\alpha_1 + 8} \\ 0 \le \alpha_1 \le 1, \\ \alpha_1 \ne \frac{8-2\sqrt{2}}{7} \end{cases} \tag{3}$$

In particular, for $\alpha_1 = \frac{8-2\sqrt{2}}{7}$ the previous system is impossible.

We obtain the set of points

$$\begin{cases} x_1 = \dfrac{2\alpha_1(1-\alpha_1)}{-7\alpha_1^2 + 16\alpha_1 - 8}, \\ x_2 = \dfrac{4(1-\alpha_1)^2}{7\alpha_1^2 - 16\alpha_1 + 8} \\ 0 \le \alpha_1 < \frac{8-2\sqrt{2}}{7} \end{cases}$$

which are Pareto minima.

For $1 \ge \alpha_1 > \frac{8-2\sqrt{2}}{7}$ is not a minimum of (P_{α_1}) being the Hessian matrix Q indefinite as can also be checked by the previous Matlab program.

4) Consider the following matrix game:

$$C = \begin{pmatrix} 1 & 4 & -1 & 5 & 2 \\ 2 & 1 & 3 & 3 & 5 \\ 2 & 3 & -2 & 3 & 1 \\ 1 & 1 & 5 & 2 & 3 \end{pmatrix}$$

(a) Find the dominated strategies and reduce the cost matrix accordingly;

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

(c) Find a mixed strategies Nash equilibrium.

SOLUTION

(a) Strategy 1 of Player 2 is dominated by Strategy 4, so that column 1 can be deleted. The reduced game is given by the matrix

$$C_{R1} = \begin{pmatrix} 4 & -1 & 5 & 2 \\ 1 & 3 & 3 & 5 \\ 3 & -2 & 3 & 1 \\ 1 & 5 & 2 & 3 \end{pmatrix}$$

Now, Strategy 1 of Player 1 is dominated by Strategy 3, so that row 1 can be deleted and the reduced matrix becomes:

$$C_{R2} = \begin{pmatrix} 1 & 3 & 3 & 5 \\ 3 & -2 & 3 & 1 \\ 1 & 5 & 2 & 3 \end{pmatrix}$$

(b) We observe that no minimum component on the columns of the reduced matrix is a maximum on the respective row, so that no pure strategies Nash equilibrium exists.

(c) The optimization problem associated with Player 1 is

$$\begin{cases} \min\ v \\ v \geq x_1 + 2x_2 + 2x_3 + x_4 \\ v \geq 4x_1 + x_2 + x_3 + x_4 \\ v \geq -x_1 + 3x_2 - 2x_3 + 5x_4 \\ v \geq 5x_1 + 3x_2 + 3x_3 + 2x_4 \\ v \geq 2x_1 + 5x_2 + x_3 + 3x_4 \\ x_1 + x_2 + x_3 + x_4 = 1 \\ x \geq 0 \end{cases} \tag{4}$$

The previous problem can be solved by Matlab.

Matlab solution

```
C=[1,4,-1,5,2; 2 1 3 3 5; 2 3 -2 3 1;1 1  5 2 3]

m = size(C,1);
n = size(C,2);
c=[zeros(m,1);1];
A= [C', -ones(n,1)]; b=zeros(n,1); Aeq=[ones(1,m),0]; beq=1;
lb= [zeros(m,1);-inf]; ub=[ ];
[sol,Val,exitflag,output,lambda] = linprog(c, A,b, Aeq, beq, lb, ub);
x = sol(1:m)
y = lambda.ineqlin
```

We obtain the optimal solution $(x, v) = (0, 0, \frac{3}{8}, \frac{5}{8}, 2.375)$. The optimal solution of the dual of (4) is given by $(y, w) = (0, 0, \frac{1}{8}, \frac{7}{8}, 0, 2.375)$. y can be found in the vector *lambda.ineqlin* given by the Matlab function linprog.

Therefore,

$$(x_1, x_2, x_3, x_4) = (0, 0, \frac{3}{8}, \frac{5}{8}), \quad (y_1, y_2, y_3, y_4, y_5) = (0, 0, \frac{1}{8}, \frac{7}{8}, 0),$$

is a mixed strategies Nash equilibrium.