GLPK Case Study 4 - 60016 Operations Research

We consider a public dataset[1] concerning Italian wines, called wines.dat. The dataset gives chemical analyses for a set of wines grown in the same region in Italy, but obtained from two different cultivars. For each cultivar, the analysis reports concentrations of alcohol, magnesium, flavanoids, wine color, protein concentration, and several other features. The dataset describes N_1 wines from the first cultivar and N_2 wines from the second cultivar. The question is whether it is sufficient to measure these chemical features to tell if a given wine comes from a cultivar or from the other.
Write a GMPL program that solves this problem for wines.dat. Solve it with glpsol and discuss what procedure has been used by the solver to find the answer.

Solution This problem goes under the name of linear separability. Suppose that we are given two sets of points $x^{(i)}$, $i = 1, \ldots, N$, and $y^{(j)}$, $j = 1, \ldots, M$ in a d-dimensional space. We need to find an hyperplane that divides the two sets. The idea is that we can use these two sets of points to train a classifier that works as follows. For any new point u that we wish to classify in one of the two groups, if u falls on the side of the hyperplane where the $x^{(i)}$ points lie, it will be classified as belonging to the first group, otherwise it will be classified in the second group of the $y^{(j)}$ points.

Let $a_1 x_1 + \ldots + a_d x_d = b$ be the equation that defines the separating hyperplane in $\mathbb{R}^d$. Also, let $x_k^{(i)}$ and $y_k^{(j)}$ be the coordinates in the k-th dimension of the points $x^{(i)}$ and $y^{(j)}$. Then we seek for a tuple $(a_1, a_2, \ldots, a_d, b)$ such that

$$\begin{aligned} a_1 x_1^{(i)} + \ldots + a_d x_d^{(i)} &< b \qquad \forall i = 1, \ldots, N \\ a_1 y_1^{(j)} + \ldots + a_d y_d^{(j)} &> b \qquad \forall j = 1, \ldots, M \end{aligned}$$

Let $\epsilon > 0$ be a small positive quantity, then the requirement can be equivalently written as

$$\begin{aligned} a_1 x_1^{(i)} + \ldots + a_d x_d^{(i)} &\leq b - \epsilon \qquad \forall i = 1, \ldots, N \\ a_1 y_1^{(j)} + \ldots + a_d y_d^{(j)} &\geq b + \epsilon \qquad \forall j = 1, \ldots, M \end{aligned}$$

or dividing both sides by ϵ as

$$\begin{aligned} a'_1 x_1^{(i)} + \ldots + a'_d x_d^{(i)} &\leq b' - 1 \qquad \forall i = 1, \ldots, N \\ a'_1 y_1^{(j)} + \ldots + a'_d y_d^{(j)} &\geq b' + 1 \qquad \forall j = 1, \ldots, M \end{aligned}$$

where $a'_k = a_k/\epsilon$ and $b' = b/\epsilon$. Finally, this can be rewritten as

$$\begin{aligned} a'_1 x_1^{(i)} + \ldots + a'_d x_d^{(i)} - b' &\leq -1 \qquad \forall i = 1, \ldots, N \\ -a'_1 y_1^{(j)} - \ldots - a'_d y_d^{(j)} + b' &\leq -1 \qquad \forall j = 1, \ldots, M \end{aligned}$$

A GMPL program that implements these requirements is given in Listing 1.

[1] The dataset comes from a public repository: https://archive.ics.uci.edu/ml/datasets/Wine.

Listing 1: wines.mod

```
set Class;
set Feature;

param N; # number ok points
param x {i in 1..N, k in Feature};

param M; # number ok points
param y {j in 1..M, k in Feature};

var a {k in Feature};
var b;

minimize z: 1;

s.t.
conx {i in 1..N}: sum {k in Feature} a[k] * x[i,k] - b <= -1;
cony {j in 1..M}: sum {k in Feature} -a[k] * y[j,k] + b <= -1;

solve;
display a;
display b;

end;
```

An interesting feature of this program is that the objective is constant, since we are only interested in find a feasible solution, given by the a and b decision variables. GLPK finds the following solution:

```
a[Alcohol].val = -0.420502091163753
a[Malicacid].val = -0.467875775381339
a[Ash].val = -3.31278830763806
a[Alcalinityofash].val = 0.391127337664156
a[Magnesium].val = -0.0303749371112767
a[Totalphenols].val = -0.0486367299837253
a[Flavanoids].val = -0.405477516966032
a[Nonflavanoidphenols].val = 1.84323151648662
a[Proanthocyanins].val = 1.01500270962373
a[Colorintensity].val = 0.00466280298848366
a[Hue].val = 1.28520933369491
a[OD280].val = -1.56249653138697
a[Proline].val = -0.00533640263407847
b.val = -16.461878775057
```

Thus, the problem is linearly separable and the classification procedure based on the hyperplane is justified by the data.

GLPK reports 63 iterations for the Simplex Algorithm. However, since the objective function is constant, what iterations is glpsol doing? In other words, what procedure is it using to solve this problem? The answer is that these are all Phase-1 iterations. The problem under consideration is simply the search for a feasible solution, hence only the Phase-1 algorithm is required.

Finally, note that as a validation test, one may copy a point from the set x^i to the set y^j and check that the program has now become infeasible (remember to increase the M value by one unit since the set y^j has one more point).