CS524 – Problem Set #9

Due Date: Friday December 1, 2023 at 09:00AM

Instructions for Handing In Homework

Formulate the following problems in GAMS and solve them. Submit this assignment electronically using the instructions on the course web page. You should hand in a single zip file containing exactly 6 files with the following names: `hw9-1.gms, hw9-2.gms, hw9-1.lst, hw9-2.lst` Question 1 is worth 100 points, Question 2 is worth 50.

1 Support Vector Machines

Support vector machines are a popular method for classification within the machine learning community. Essentially, a linear classifier f is generated as $f(x) = w'x + \gamma$ such that if $f(x) > 0$ then x is in class P_+ whereas if $f(x) < 0$ then $x \in P_-$.

Once you have the classifier, you can use it with new data x to predict if $x \in P_+$ or $x \in P_i$ just by evaluating $f(x)$.

In this exercise, you should use the data that is provided in the abalone.gdx file, and you should attempt to predict whether the "number of rings" is greater than 10 or not. The data file provides 4177 samples, with the following features (including the number of rings):

For your information the data fields are:

Name	Data Type	Meas.	Description
Sex	nominal		M, F, and I (infant)
Length	continuous	mm	Longest shell measurement
Diameter	continuous	mm	perpendicular to length
Height	continuous	mm	with meat in shell
Whole weight	continuous	grams	whole abalone
Shucked weight	continuous	grams	weight of meat
Viscera weight	continuous	grams	gut weight (after bleeding)
Shell weight	continuous	grams	after being dried
Rings	integer		+1.5 gives the age in years

Note that the first nominal value has been converted to 3 binary values and can be treated as separate data for this assignment. The following gams code uses the set j to index the features, excluding the "rings" that is stored as a label y indicating whether the number of rings is greater than 10 or not.

Instead of using all the data to generate the classifier f, we use a subset of the samples i as training data, another set as tuning data to determine some tradeoff parameters and the remaining data as the testing data (to see how well we do). You should pretend that you do not know the values of y for the testing data since normally you won't!

```
set i(*), headr(*);
parameter A(i,headr);
$gdxin abalone.gdx
```

```
$LOAD i
$LOAD headr
$LOAD A=Data
$gdxin

set  j(headr)  index of independent variables;
j(headr) = yes$(not sameas(headr,'Rings'));

parameter y(i);
y(i) = -1 + 2$(A(i,'Rings') gt 10);

set train(i), tune(i), test(i);
train(i) = yes$(ord(i) le 3000);
tune(i) = yes$(ord(i) gt 3000 and ord(i) le 3500);
test(i) = yes$(ord(i) gt 3500);
```

To generate this classifier a quadratic optimization model is solved:

$$\begin{array}{ll} \min_{w,\gamma,\delta} & \frac{1}{2}w^T w + C\sum_i \delta_i \\ \text{s.t.} & y_i[A_{i\cdot}w - \gamma] \geq 1 - \delta_i, i = 1,\ldots,m \\ & \delta_i \geq 0 \end{array}$$

Here m runs over a set of training examples, and y_i is 1 if $i \in P_+$ and -1 if $i \in P_-$. $A_{i\cdot}$ are the values of the predictor variables for the ith sample. Note that C is a tradeoff parameter between errors δ_i and a measure of generalization ability w and we will choose an appropriate value of C. Initially set $C = 1$.

The dual problem of this quadratic program is:

$$\begin{array}{ll} \max_{\alpha} & \sum_i \alpha_i - \frac{1}{2}\sum_{i,k} \alpha_i\alpha_k y_i y_k A_{i\cdot}A_{k\cdot}^T \\ \text{s.t.} & \sum_i y_i\alpha_i = 0 \\ & C \geq \alpha_i \geq 0 \end{array}$$

1.1 Problem

You should formulate both these problems in GAMS. Note that for the dual problem, you should introduce some intermediate variables

$$v_j = \sum_i \alpha_i y_i A_{ij}$$

and express the objective function as

$$\sum_i \alpha_i - \frac{1}{2}\sum_j v_j^2$$

where j runs over the predictor features.

1.2 Problem

Solve both problems using the qcp solver of your choice, and then use the multiplier information from the dual solve to set the values of w and γ in a parameter as shown below.

```
parameter wsolns(*,headr), gsolns(*);
wsolns('primal',j) = w.l(j);
```

```
gsolns('primal') = gamma.l;
wsolns('dual',j) = XXX;
gsolns('dual') = YYY;
```

1.3 Problem

Utilize the code demonstrated in class for 5 fold cross validation to determine a good value for C (from the possible values "0.1", "1", "10", "100", "1000", "10000") and use this value subsequently. You should use the following code to set the train and tune data for each fold:

```
loop(folds,
    tune(i) = yes$(i.ord gt (folds.ord-1)*700 and i.ord le folds.ord*700);
    train(i) = yes$(not test(i) and (not tune(i)));
    solve svmmod  min pobj using qcp;
);
```

Save the values of the errors on each trial value into a parameter `errvec`.

1.4 Problem

Now with the fixed value of C from the above, solve the optimization problem (or its dual) using all the training data. Then determine the error rate that you would expect to make on unseen data (i.e. on the testing data: this is the first time you should use the test data, and you should not solve any optimization problem with that data, just see if your prediction is correct or not). Save this error rate as a scalar parameter `errorrate`.

Finally display all the required values at the end of your list file using

```
display wsolns, gsolns, errvec, C, errorrate;
```

2 Advertising Budget for Weasley's Wizard Wheezes

The Weasley's are buying advertising time on the radio in order to reach wizards and witches. There are a set N of adds that they could possibly purchase. Each minute of advertising of type $j \in N$ costs c_j. If they purchase x_j minutes of advertising of an add of type j, then they will ensure that $\alpha_j\sqrt{x_j}$ witches and $\beta_j\sqrt{x_j}$ wizards hear their spots.

They would like to find a least cost advertising strategy so that their ads will reach K_1 witches and K_2 wizards.

For both problems, please use the following gams code to create the instance:

```
sets N /ad1*ad20/;  alias(I,N) ;
parameters c(I) Cost,
  alpha(I) Witches proportionality constant,
  beta(I) Wizards proportionality constant ;
scalars K1, K2 ;

c(I) = normal(100,5) ;
alpha(I) = uniform(7,13) ;
beta(I) = 13-alpha(I) + 7 + 5$(uniform(0,1) < 0.3) ;
K1 = 5000; K2 = 8000;
```

2.1 Problem

Write a GAMS NLP model that will determine the number of minutes of each advertising spot to purchase. Store the amount of each ad purchased and the total time purchased using the code below:

```
solve w1 using nlp minimizing cost ;
parameter solution(*,i), totalAdTime(*);
totalAdTime('nlp') = sum(I, x.L(I)) ;
solution('nlp',i) = x.l(i);
```

2.2 Problem

Write a GAMS *qcp* model that will determine the number of minutes of each advertising spot to purchase. You will need to add some auxiliary variables. Display the amount of each ad purchased and the total time purchased using the code below:

```
option qcp = mosek;
solve w2 using qcp minimizing cost;
totalAdTime('qcp') = sum(I, x.L(I)) ;
solution('qcp',i) = x.l(i);
display solution, totalAdTime;
```