**Machine Learning from Data – IDC**

**HW5 – Support Vectors Machine**

**\*\*\* This assignment can be submitted in pairs.**

In this assignment you will use Weka’s implementation of an SVM classifier (called SMO). First you will divide the data to training and test set. Then you will select the best kernel according to the TPR and FPR (with ). In addition you will plot the result of each kernel in a ROC graph in excel. After selecting the best kernel you will try different values for the parameter C (the slack regularization) and once again you will plot the result for each value in a ROC graph.

Finally, you will answer some theoretical questions regarding kernel functions and Lagrange multipliers (at the end of the document).

Finding the best kernel:

1. Load the cancer data (cancer.txt).
2. Divide the data to training and test set – 80% training and 20% test.
3. For each kernel value, build the SVM classifier on the training set using the SMO WEKA class.
4. Calculate & print to the console the TPR and FPR on the test set.
5. Select the best kernel according to the best TPR-FPR (with ).
6. The possible values for the kernel are:
   1. Polynomial Kernel – with one of the following degrees
   2. RBF Kernel – with one of the following gamma values
7. In the attached excel file, fill the TPR and FPR for each kernel in the appropriate yellow cells. Add a parallel line (to the line from the origin to (1,1), which is already there) that shows the best kernel on the graph.

Finding the best C value (the slack regularization):

1. For the selected kernel, try different C values.
2. For each C value, build the SVM classifier with the selected kernel on the training set using the SMO WEKA class.
3. Calculate & print to the console the TPR and FPR on the test set.
4. The possible C values are all the combinations of , where and
5. In the attached excel file, fill the selected kernel in the gray cell and fill the TPR and FPR for each C value in the appropriate green cells. The ROC graph will update automatically. Does your graph make sense?

In order to do the above you need to first install WEKA:

1. See instructions in HW1.

Prepare your Eclipse project:

1. Create a project in Eclipse called HomeWork5.
2. Create a package called HomeWork5.
3. Move the SVM.java and MainHW5.java that you downloaded from the Moodle into this package.
4. Add WEKA to the project (see instructions in HW1).

The following methods are mandatory methods and you can't change their signatures, but you can add additional methods:

1. void setKernel: Setting the Weka SMO classifier kernel.
   1. Input: Kernel object.
2. void setC: Setting the C value for the Weka SMO classifier.
   1. Input: double.
3. double getC: Getting the C value for the Weka SMO classifier.
   1. output: double.
4. Int[] calcConfusion: Calculate the TP, FP, TN, FN for a given instances object.
   1. Input: Instances object.
   2. Output: int array of size 4 in this order [TP, FP, TN, FN].

SMO code examples:

Creating an SVM classifier:

SMO smo = new SMO();

PolyKernel kernel = new PolyKernel();

smo.setKernel(kernel);

smo.buildClassifier(data);

Setting the degree of a polynomial kernel:

PolyKernel kernel = new PolyKernel();

kernel.setExponent(degree);

Choosing RBF kernel:

RBFKernel kernel = new RBFKernel();

smo.setKernel(kernel);

smo.buildClassifier(data);

Setting the gamma parameter of an RBF kernel:

RBFKernel kernel = new RBFKernel();

kernel.setGamma(0.3);

Classify a new instance:

smo.classifyInstance(someInstance);

Define the kernel object for the SVM

smo.setKernel(kernel);

Set the parameter of the slack variable

m\_smo.setC(c); }

Your complete output should look like this:

For PolyKernel with degree <the degree of the polynomial kernel> the rates are:

TPR = <TRP\_Result>

FPR = <FPR\_Result>

For RBFKernel with gamma <the gamma value> the rates are:

TPR = <TRP\_Result>

FPR = <FPR\_Result>

The best kernel is: <Poly or RBF> <kernel parameter (degree or gamma)> <TPR / FPR>

For C <C value> the rates are:

TPR = <TRP\_Result>

FPR = <FPR\_Result>

The yellow part should occur many times, one for each polynomial kernel.

Similarly, the red part should occur for each RBF kernel.

And the green part should occur for each C value.

Theoretical questions:

1. Use Lagrange Multipliers to find the maximum and minimum values of the function subject to the given constraints:
   1. ; constraint: 2
   2. ; constraint:
   3. ; constraints:
2. Show that any Kernel function is symmetric, i.e.
3. Find the Kernel function for the following mapping:

You should hand in a SVM.java, MainHW5.java, the hw5 excel file filled with the relevant results and this word document with answers to the theoretical questions. All of these files should be placed in a hw\_5\_##id1##\_##id2##.zip folder with the id of both members of the team.

\*\*\* Submitting in groups on Moodle does not work. Please only submit one zip folder per pair

**\*\* Note: We asked Ben about dividing by alpha(when using Lagrange Multipliers), and Ben said that we can assume that the weight of each constraint does not equal zero. Therefor a division by alpha is legit.**

1. Use Lagrange Multipliers to find the maximum and minimum values of the function subject to the given constraints:

* 1. ; constraint: 2

2



So  can’t be a solution

; constraint:

Constraint  thus 



 then 

We have 2 Possibilities: , but y=0 is not held by the constraint

We got 

 We get: 

2 Points:

, => Max

, => Min

c. ; constraints:

We got 2 solutions: and

1. Show that any Kernel function is symmetric, i.e.



1. Find the Kernel function for the following mapping:

Answer: 

Proof:





