COMP3308/COMP3608 Introduction to Artificial Intelligence

Week 10 Tutorial exercises
Support Vector Machines. Ensembles of Classifiers.

This week we have a smaller number of tutorial exercises. We will use the remaining time for questions about Assignment 2.

Regarding Assignment 2: Please do not underestimate the report! It is worth 12/24 marks = 50% of your mark for this assignment. Sometimes students spend too much time on the code and then there is not enough time left to write a good quality report.

Exercise 1. (Homework)

This is a set of 5 multiple choice questions to be answered online on Canvas. To complete the homework, please go to Canvas -> Quizzes->w10-homework-submission. Remember to press the "Submit" button at the end. **Only 1 attempt is allowed.**

Here are the questions:

Q1. The problem of finding a decision boundary in SVM can be formulated as an optimisation problem using Lagrange multipliers. What is the goal?

a) To maximize the margin of the decision boundary.
b) To minimize the margin of the decision boundary.

Q2. After SVM learning, each Lagrange multiplier *lambda_i* takes either zero or non-zero value. Which one is correct:

a) A non-zero *lambda_i* indicates that example *i* is a support vector.
b) A zero *lambda_i* indicates that example *i* is a support vector.
c) A non-zero *lambda_i* indicates that the learning has not yet converged to a global minimum.
d) A zero *lambda_i* indicates that the learning process has identified support for example *i*.

Q3. In linear SVM, during training we compute dot products between:

a) training vectors
b) training and testing vectors
c) support vectors
d) support vectors and Lagrange multiplayers

Q4. Bagging is only applicable to classification problems and cannot be applied to regression problems.

a) True
b) False

Q5. Boosting is guaranteed to improve the performance of the single classifier it uses.

a) True
b) False

Explanation:

Q1: The goal is to find the decision boundary (hyperplane) with maximum margin => the optimization problem is to maximize the margin.

Q2: This is the definition of a support vector.

Q3: During training, we compute dot products between pairs of training vectors; during testing – between the testing example and the support vectors. Note that d) can be immediately ruled out as it is a product of a vector and coefficient; dot product is a product of 2 vectors.

Q4: Bagging can be applied to both classification and regression problems. In regression problems, the individual predictions will be averaged (see slide 14 of lecture10b).

Q5: There is no guarantee that an ensemble of classifiers (including Boosting) will produce better accuracy than the single classifier it uses. In practice, ensembles often perform better but there is no guarantee.

<u>Exercise 2.</u> Bagging, Boosting and Random Forest

a) What are the similarities and differences between Bagging and Boosting?

Answer: See the lecture slides (slide 31)

b) What are the 2 main ideas that are combined in Random Forest?

Answer: Bagging and random selection of features

<u>Exercise 3</u> Boosting

Continue the example from slides 24-25. The current probabilities of the training examples to be used by AdaBoost are given below:

$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	$p(x_5)$	$p(x_6)$	$p(x_7)$	$p(x_8)$	$p(x_9)$	$p(x_{10})$
0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.17	0.17	0.17

From these, a training set T_2 has been created and using T_2 a classifier C_2 has been generated. Suppose that C_2 misclassifies examples x_2 and x_9. Show how the probabilities of all training examples are recalculated and normalized.

Solution:

1. Initial probabilities:

$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	$p(x_5)$	$p(x_6)$	$p(x_7)$	$p(x_8)$	$p(x_9)$	$p(x_{10})$
0.07	0.07	0.07	0.07	0.07	0.07	0.07	0.17	0.17	0.17

2. Examples 2 and 9 are misclassified => their errors *e* are 1, while the errors of the other 8 examples are 0
   The weighed error of the classifier C_2 will be: ϵ_2=0.07*1+0.17*1=0.24
3. The term β for the probability modification will be: $\beta_2=\epsilon_2/(1-\epsilon_2)$=0.24/0.76=0.32
4. Calculating the new probabilities by modifying (reducing) only the probabilities of the correctly classified examples by: $p(x_j)=p(x_j)*\beta_2$

   The new probabilities are:

$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	$p(x_5)$	$p(x_6)$	$p(x_7)$	$p(x_8)$	$p(x_9)$	$p(x_{10})$
0.07*0.32	0.07	0.07*0.32	0.07*0.32	0.07*0.32	0.07*0.32	0.07*0.32	0.17*0.32	0.17	0.17*.32

$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	$p(x_5)$	$p(x_6)$	$p(x_7)$	$p(x_8)$	$p(x_9)$	$p(x_{10})$

0.022	0.07	0.022	0.022	0.022	0.022	0.022	0.054	0.17	0.054

5. Normalization:
   Sum of all probabilities: 0.022*6+0.054*2+0.07+0.17=0.132+0.108+0.24=0.48

   Normalized probabilities:

$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	$p(x_5)$	$p(x_6)$	$p(x_7)$	$p(x_8)$	$p(x_9)$	$p(x_{10})$
0.022/0.48	0.07/0.48	0.022/0.48	0.022/0.48	0.022/0.48	0.022/0.48	0.022/0.48	0.054/0.48	0.17/0.48	0.054/0.48

$p(x_1)$	$p(x_2)$	$p(x_3)$	$p(x_4)$	$p(x_5)$	$p(x_6)$	$p(x_7)$	$p(x_8)$	$p(x_9)$	$p(x_{10})$
0.045	0.146	0.045	0.045	0.045	0.045	0.045	0.113	0.354	0.113

We can see that the weights of the misclassified examples (x_2 and x_9) have increased while the weights of the correctly classified examples (the remaining 8 examples) have decreased. In the next iteration x_2 and x_9 will have a higher chance to be selected for the new training set, i.e. the next classifier will focus on learning to classify these more difficult examples.

Exercises using Weka

***Exercise 4.* SVM in Weka – Binary classification**

1. Load the diabetes data (diabetes.arff) or another binary classification data. The two classes is the diabetes data are "tested_positive" and "tested_negative".

Choose 10-fold cross validation. Run the SVM (SMO located under "functions"). This implementation of SVM uses John Platt's sequential minimal optimization algorithm to solve the optimization problem (i.e. to train SVM classifier) and shows the computed decision boundary.

Examine the output.
a) What type of SVM is used – linear or non-linear? What type of kernel function is used? Where is the equation of the decision boundary?

Answer:
Linear SVM. This is evident from "Linear Kernel: K(x,y) = <x,y>". This is the default option in Weka.

The equation of the decision boundary is:

```
      1.3614 * (normalized) preg
+     4.8764 * (normalized) plas
+    -0.8118 * (normalized) pres
+    -0.1158 * (normalized) skin
+    -0.1776 * (normalized) insu
+     3.0745 * (normalized) mass
+     1.4242 * (normalized) pedi
+     0.2601 * (normalized) age
-    5.1761
```

b) Let's try non-linear SVM. Edit the properties of SMO (by clicking on "SMO"). Keep the same type of kernel (Poly Kernel, it is a general type of kernel and we can use it to create both linear and no-linear SVMs) but change the exponent from 1 to 2 (by clicking on "Poly kernel"). Now the kernel function is nonlinear: Poly Kernel: K(x,y) = <x,y>^2.0. Run the classifier, observe the new equation of the decision boundary and the new accuracy. Compare the accuracy of the linear and non-linear SVM, which one is better? Experiment with other types of kernels (e.g. other polynomial, by increasing and exponent, or RBF).

3) Compare SVM's accuracy with the backpropagation neural network and the other classifiers we have studied.

Exercise 5. ***SVM in Weka – Multi-class classification***

Load the iris data (iris.arff). Choose 10-fold cross validation and run the SVM classifier.

a) Examine the output. Not 1 but 3 SVM are generated. Do you know why?

Answer:
This is because the task is a multi-class problem (3-class, there are 3 types of irises). SVM handles multi-class problems by transforming them into several binary problems. In general, there are 2 methods to do this: 1-versus-rest and one-versus-one. In the first case, k classifiers are constructed, where k is the number of classes; each of them learns a hyperplane to separate 1 class versus all the remaining classes. A new example is assigned to the class for which the distance to the margin is maximal. In the second case, k(k-1)/2 classifiers (i.e. hyperplanes) are constructed, 1 class versus another class. To classify a new example, a decision function using some ad-hoc voting scheme is used.

b) Examining again the output, which of the 2 methods does Weka use?

Answer: The second one, 1 class vs another:

```
Classifier for classes: Iris-setosa, Iris-versicolor …
Classifier for classes: Iris-setosa, Iris-versicolor …
Classifier for classes: Iris-versicolor, Iris-virginica …
```

The voting scheme to combine these 3 classifiers is not clear from the Weka's output.

Exercise 6. ***Ensembles of classifiers in Weka – Bagging, Boosting and Random Forest***

1. Load iris data (iris.arff). Choose 10-fold cross validation. Run the decision trees (J48) single classifiers.

2. Run bagging of decision trees (J48). It is under "Meta". The default base classifier is REPTree, you need to change it to j48. Compare the performance with the single DT classifier from 1).

3. Run boosting (AdaBoost) of decision trees (J48). It is also under "Meta" and you need to change the default base classifier. Compare performance with the single classifiers from 1) and also with the bagged classifier from 2).

4. Run Random Forest of decision trees (J48). It's under "Trees" and you need to change the default base classifier. Compare performance with the previous classifiers.

5. Use AdaBoost with OneR as a base classifier. Experiment with different number of hypotheses M. What happens with the accuracy on the training data as M increases? What about the accuracy on test data? Is this consistent with the AdaBoost theorem?

Exercise 7. Do you have any questions about Assignment 2? Ask your tutor!