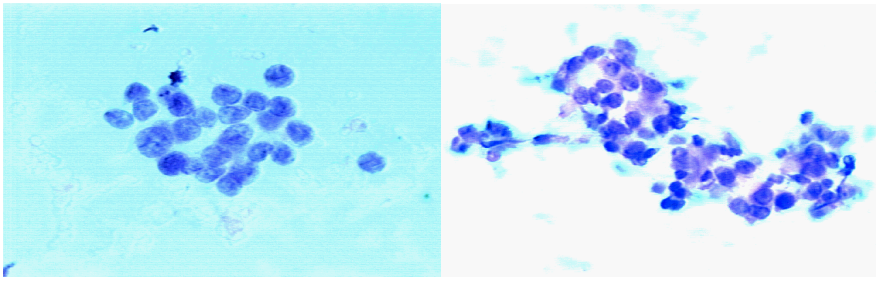
COMP3055 Machine Learning Coursework

Deadline: 4pm Friday Dec 21, 2018
Submit an electronic copy via Moodle

The coursework aims to make use of the machine learning techniques learned in this course to diagnose breast cancer using Wisconsin Diagnostic Breast Cancer (**WDBC**) dataset. WDBC contains 569 instances of breast cancer data collected in by professors in the University of Wisconsin. Each instance is either labeled as M (malignant) or B (benign). In others words, you are going to solve a binary classification problem. Features are computed by analyzing a digitized image of a fine needle aspirate (FNA) of a breast mass, instead of using pixels as raw input. They describe characteristics of the cell nuclei present in the image (see the following for example images).



In particular, the input include ten real-valued features for each cell nucleus (three in total):

a) Radius (mean of distance from center to points on the perimeter)
b) Texture (standard deviation of gray-scale values)
c) Perimeter
d) Area
e) Smoothness (local variation in radius lengths)
f) Compactness (perimeter^2 / area -1.0)
g) Concavity (severity of concave portions of the contour)
h) Concave points (number of concave portions of the contour)
i) Symmetry
j) Fractal dimension ("coastline approximate"-1)

In total, there are 30 features (feature dimension is 30) available for diagnosis. All features are recorded using four digits for precision.

You will perform the following tasks using Matlab or other languages at your choice (e.g. Python):

Task 1: You can find WDBC dataset file (wdbc.data) from moodle under coursework section. The data file is arranged in the way that each line represents an instance of the data. Within each line, the attribute values are separated by comma (,) and there are total 32 attributes. The first attribute is the patient's ID. The second attribute is the class label (either M or B). The rest of the attributes are the input features. Do the following:

1. Load the data from the file into data matrix for the subsequent tasks. In Matlab, you can use function `csvread` to do so. Note that you need to read the second attribute separately as class label and ignore the first attribute. Then you need to read the rest of attributes as features.

2. Split the data portions: a) select 169 samples as testing data and b) 400 samples for training.

Task 2: Design and implement a breast cancer diagnosis system using decision tree with dimension reduction. Do the following

1. Apply PCA to reduce the original input features into new feature vectors with different dimensions, 3, 5, 7, 9, 11.
2. Use training data to do 10-fold cross validation to train and validate your decision trees with different input feature vectors (original input and reduced input calculated in step 1). You can use default parameters for your decision trees according the library you use.
3. Using test data to compute f1 values for each model and Plot a figure showing result vs feature dimension.

Task 3: Design and implement a breast cancer diagnosis system using SVM. Do the following:

1. Use training data to do 10-fold cross validation to train and validate your models. For the input features, use the one that gives the best performance in task 2. You need to use linear, polynomial, and rbf kernels for your models. Note that each kernel has different parameters to set, for example, orders for polynomial model and sigma for rbf kernels. You can simply use the default parameters for each kernel.
2. Use test data to compute the classification error, precision, recall and f1 for your models with different kernels in step 1. In the rbf kernel case, draw an ROC curve with different parameters at your choice.

Task 4 (Optional): Find the best SVM model. You are required do a parameter search for each kernels and use cross validation to find the best performer. You should also use soft SVM with different penalty parameters. There are no rule-of-the-thumb on how you should search the best combination of parameters. Try your best to obtain the highest performance in terms of precision and recall (f1).

Task 5: Based on your experiences of performing task 2 and task 3 and findings therein, in your own words, compare and contrast the performances (error rate, precision and recall, f1), computational complexity (time), level of overfitting of the two approaches. To look at the level of overfitting, you can compare the performance of a given model on the training data with test data and see how different they are. State which one you think would be a better approach to this problem and explain why.

What to submit: A report of no more than 6 pages including all the figures and tables summarizing how above tasks are done, justification on your decisions involved, and the results of your analysis. A zipped file with all your source code. Note that you should properly organize your code with appropriate comments for easy of marking and running.

Marking scheme: this coursework takes 30% of your total marks in this module. The marking distribution is given in 100 scaling as follows:

1) Completeness of task 1 (10 marks)
2) Completeness of task 2 (30 marks)
3) Completeness of task 3 (30 marks)
4) Completeness of task 5 (10 marks)

5) Report writing (15 marks)
6) Coding with proper comments and organization (5 marks)

If you complete task 4, you will get 5 bonus marks in addition to the above marks.