# MIDDLESEX UNIVERSITY
# COURSEWORK PART 2

2009/10

BIS3226

AI Techniques in Information Management

Roman V Belavkin

This assignment is worth 25% of the overall grade. The submission date is **Friday, March 26, 2010**. You should do the assignment **individually**.

## Contents

# 1 Aims and Objectives

The aim of this part of the coursework is to develop and evaluate data-driven models based on mean-square linear regression and on feed-forward artificial neural networks. The models should be trained on publicly available real-world datasets. Links to data repositories will be provided. The performance of the models should be evaluated against several criteria, such as ease of use, the ability to explain the relationships between variables in th data and the ability to predict and generalise to new cases. Your work will be assessed based on a written report, which should include the following components:

1. A description of the data, objectives of the analysis, methods as well as anticipated results;
2. A report of the results;
3. Analysis of the results and reflection on the performance of the models.

# 2 Software Required

You will need a copy of Microsoft Excel with the Data Analysis Toolpack and the BrainCel Add-Ins. The software will be installed and available on the University computers. It is advised to back up your work.

# 3 The Data

You are encouraged to choose a dataset that you will use in the coursework.[1] You can find data in the following publicly available repositories:

| | |
|---|---|
| `fx.sauder.ubc.ca` | ForEx data at The University of British Colombia |
| `archive.ics.uci.edu/ml/` | Machine Learning Repository at University of California, Irvine |
| `fimi.cs.helsinki.fi` | Frequent Itemset Mining Implementations Repository |

For example, on the first site you can retrieve the daily prices of world currencies, precious metals and oil. Table 1 shows prices in British Pounds (GBP) of 10 top currencies during 2006–2007. The table has 12 columns (date, weekday and names of ten currencies) and many rows (for two years period there will be about 600 rows).

Table 1: Daily exchange rates of top 10 currencies in price notation. Base currency: GBP. Retrieved from PACIFIC Exchange Rate Service by Werner Antweiler, University of British Columbia.

| Date | Wdy | USD | CAD | EUR | JPY | CHF | AUD | HKD | NZD | KRW | MXN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2006/01/03 | Tue | 0.57467 | 0.49665 | 0.68840 | 0.0049392 | 0.44410 | 0.42404 | 0.074115 | 0.39156 | 0.00057214 | 0.053986 |
| 2006/01/04 | Wed | 0.56853 | 0.49373 | 0.68752 | 0.0048879 | 0.44406 | 0.42441 | 0.073325 | 0.39118 | 0.00056927 | 0.053718 |
| 2006/01/05 | Thu | 0.56929 | 0.49005 | 0.68887 | 0.0049089 | 0.44575 | 0.42590 | 0.073422 | 0.39136 | 0.00057238 | 0.053759 |
| 2006/01/06 | Fri | 0.56508 | 0.48513 | 0.68641 | 0.0049396 | 0.44506 | 0.42527 | 0.072881 | 0.39043 | 0.00057100 | 0.053461 |
| 2006/01/09 | Mon | 0.56677 | 0.48475 | 0.68379 | 0.0049416 | 0.44297 | 0.42600 | 0.073123 | 0.39231 | 0.00057977 | 0.053662 |
| 2006/01/10 | Tue | 0.56689 | 0.48752 | 0.68370 | 0.0049449 | 0.44267 | 0.42444 | 0.073137 | 0.39279 | 0.00057722 | 0.053481 |
| 2006/01/11 | Wed | 0.56684 | 0.48950 | 0.68789 | 0.0049738 | 0.44520 | 0.42851 | 0.073128 | 0.39512 | 0.00057565 | 0.053404 |
| 2006/01/12 | Thu | 0.56785 | 0.48847 | 0.68337 | 0.0049673 | 0.44153 | 0.42639 | 0.073258 | 0.39351 | 0.00058275 | 0.053732 |
| 2006/01/13 | Fri | 0.56448 | 0.48607 | 0.68327 | 0.0049322 | 0.44082 | 0.42517 | 0.072821 | 0.39372 | 0.00057162 | 0.053371 |
| 2006/01/16 | Mon | 0.56619 | 0.48902 | 0.68614 | 0.0049244 | 0.44246 | 0.42701 | 0.073022 | 0.39498 | 0.00057607 | 0.053646 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 2007/12/17 | Mon | 0.49582 | 0.49222 | 0.71239 | 0.0043798 | 0.43025 | 0.42484 | 0.063584 | 0.37360 | 0.00053111 | 0.045683 |

You can use a similar or different dataset for your coursework. It is important, however, that your data is

[1] If this is your **resit** coursework, then you must use a different dataset on your second attempt.

**Multidimensional** — each case (or each event) is described by several variables. The variables (or dimensions) usually correspond to columns in the table. Note that you can create a multidimensional dataset by using multiple copies of one variable, but at different time moments, as used in autoregressive models.

**Representative** — the dataset must be have sufficient number of cases for the analysis had some statistical significance. The cases (or events) are usually the rows of your database table.

Thus, you need data that can be displayed in a table with several columns and usually large number of rows.

# 4 Assessment of the Components

## 4.1 Introduction of data, objectives and methods (30%)

First, introduce the data you are using. This introduction should include:

1. The name and the origin of the data. Do not forget to give appropriate credit to the owners/creators of the dataset.
2. Show in a sample table of the data, such as shown in Table 1. Do not print the whole data in your table, as it can be too large. It is sufficient to show only the first 10 rows of the table as in Table 1.
3. Describe the **variables**.
4. Describe the **cases**, what do they represent and how many cases there are.

Second, you need to explain what are the objectives of your analysis and what do you anticipate to find. For example, do you expect any particular dependency between the variables?

Third, you need to introduce the methods. In particular, you need to give some details about the models you are going to use, what are the input and output variables, what are the training procedures and how do you evaluate their performance.

## 4.2 Report the results (30%)

Here you need to report the results of your data-driven models. These results should include:

- Parameters of the linear model after training (i.e. intercept and regression coefficients).
- The average errors of the linear model on the training and on the testing sets.
- Parameters of the neural network model (i.e. the configuration of a neural network). You can also include the weights of the neural network after training.
- The average errors of the neural network on the training and on the testing sets.
- If you use the Naive model as a reference, then report its average errors for the training and testing sets.
- Show samples of the tables with the calculations of the errors (you can show only 10 rows of the tables).

You may use plots to facilitate the presentation of your results.

### 4.3 Analysis of the results and evaluation of the models (30%)

First, you need to critically analyse and reflect on the results you have obtained. You have to compare the results with your expectations. Is there any interesting information that these results provide about the data and the processes underlying the data? Have you discovered any new knowledge that you can use (e.g. in a business environment)?

Second, you need to evaluate the models. In particular, you need to think about the following questions:

- Which model do you think is better for your task and objectives?
- Which model is easier to use, and which produced better results?
- If some model did not perform very well on your data, what do you think are the reasons?
- Which model is better for explanation and understanding of the data, and which model is better for generalisation or prediction of new data?

### 4.4 Presentation (10%)

Your report should be well presented. A good guide is the *Publication Manual* of the American Psychological Association (e.g. see `http://www.apastyle.org/`). At the very least, your report should be clear, typed or nicely hand-written document with good spelling, grammar and easy to understand English. There is no word limit, but a useful report should be just long enough to describe the work. A sensible limit is about 10 pages of typed text. Beyond this, you are probably being a bit too verbose. Tables, graphs, careful labelling and numbering are all well established and effective presentation tools.

Things to avoid are:

- Including images or diagrams that you did not create yourself or did not obtain the permission to use from the author (even if the image is from the Internet).
- Including graphs or diagrams that you do not describe in the text.
- Forgetting to label the axes on the charts.
- Using 3D charts to display 2D information.
- Including material irrelevant to the work.

Note also that you do not need to use coloured charts, as these can be quite expensive to print. A lot of of information can be displayed using black and white patterns or gradations of grey.

## 5 Assignment Submissions

Submit your report to the the Computing Science Student office, room TG18 by **Friday, March 26, 2010**, 16:00 hours. Do not include a disk or any other materials. Ensure that your work is clearly labelled with your name, student number, campus, course and the name of the module leader. Ensure that it is securely bound and easy to open. You should attach a coursework feedback form which will be dated and receipted. You should keep your receipt — it is for your own protection. Do not hand the coursework directly to your tutor.

# A Supplementary Material

The main theoretical material for this coursework is described in units on linear models, feed-forward neural networks and clustering algorithms. This section provides additional information on how to create linear and artificial neural network models in MS Excel with the BrainCel add-in.

## A.1 Tips on working with the data in MS Excel

If you use the MS Excel spreadsheet for your coursework, then this section will give you some advice on how to prepare and organise the data in the spreadsheet and how to do the main experiments with the models. Usually, all the variables in your dataset are distinguished either as the independent (input) or the dependent (output) variables. We shall denote the independent variables as $x_1$, $x_2$ and so on, while the dependent as $y_1$, $y_2$. In fact, for this coursework you can just consider one output variable, and we shall denote it simply as $y$. For example, if you are using ForEx data, as in Table 1, you can choose to predict EUR based on the values of all other currencies. In this case, variable $y$ represents the values of EUR, and $x_1, \ldots, x_m$ are the other currencies in the table. Usually, the output variable is placed in the last column of the table (column $y$) as shown below:

| $x_1$ | $x_2$ | $\cdots$ | $x_m$ | $y$ |
|---|---|---|---|---|
| $X_{11}$ | $X_{12}$ | $\cdots$ | $X_{1m}$ | $Y_1$ |
| $X_{21}$ | $X_{22}$ | $\cdots$ | $X_{2m}$ | $Y_2$ |
| $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $X_{n1}$ | $X_{n2}$ | $\cdots$ | $X_{nm}$ | $Y_n$ |

It is hoped that variable $y$ depends on the input variables $x_1, \ldots, x_m$, and this is why we refer to $y$ as the 'dependent' variable. However, usually it is not known how much $y$ depends on $x_1, \ldots, x_m$, and even what kind of dependency it is. The goal of many data analysis and data miming techniques is to understand and simulate this dependency as good as we can. This is done by creating a model (mathematical model) of this dependency from the data available, which means that one tries to find a function $y \approx f(x_1, \ldots, x_m)$ that computes the value of $y$ from the values of $x_1, \ldots, x_m$ (the sign $\approx$ here means 'approximately equals').

It may be a good idea to create not one, but several different models and compare their performance on the same data. For example, you can use a linear model and a neural network model. The following two sections will give you some tips on creating these models. Each model has some advantages and some disadvantages, and it is good to understand which they are from your own experience. For example, one model can give a better explanation of the dependency, while other can be more precise. Section A.4 will give you tips on how to evaluate the precision of the models by computing the errors and their average errors.

To test the predictive power of a model, you should split the data into two parts — one for training and one for testing the model. We shall call these two parts of data the *training set* and the *testing set*. However, before you split the data, it is advised to prepare the table with all the functions necessary to do the computations. These are the functions for the forecasts using the models and also all the functions to compute the absolute errors and the average errors. The table below is an example of such a table. It is advised that you first put all the formulae in the first row, and after making sure that they compute correctly, copy and paste them across the remaining table.

After you have prepared such a table, you can split it into the training and testing sets. Normally, we use about 70% of data for training and 30% for testing. Sometimes you can just use the upper 70% part of the table for training and the lower part for testing. However, in some datasets, such as describing time-series, such a split would mean that the data for training and testing has very different properties (e.g. they may refer to quite different time periods). Therefore, it is usually much better to split the data randomly. The BrainCel software provides a function called *Random Select and Move* under the Preprocessing menu which can split the table into two subsets. You are

Table 2: An example of how you can prepare the table for an experiment.

| $x_1$ | $x_2$ | ... | $x_m$ | $y$ | Linear model | Abs.e | ANN model | Abs.e |
|---|---|---|---|---|---|---|---|---|
| $X_{11}$ | $X_{12}$ | ... | $X_{1m}$ | $Y_1$ | ? | ? | ? | ? |
| $X_{21}$ | $X_{22}$ | ... | $X_{2m}$ | $Y_2$ | ? | ? | ? | ? |
| $X_{31}$ | $X_{32}$ | ... | $X_{3m}$ | $Y_3$ | ? | ? | ? | ? |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | | | | |
| $X_{n1}$ | $X_{n2}$ | ... | $X_{nm}$ | $Y_n$ | ? | ? | ? | ? |
| | | | Average error: | | | ? | | ? |

advised to use this function to split the data by moving randomly $\approx 70\%$ of the data, which will be your training set, and use the remaining $\approx 30\%$ for testing.

To split the table, you should select the whole table including all the formulae and define a name for it (Insert/Name/Define in MS Excel). You can use the named region to randomly select and move 70% of rows for the training set. The remaining 30% will be your testing set, and it will already contain all the necessary functions for computing the forecasts, the absolute errors and the average errors. The advantage of using this method is that all the models use exactly the same testing set to make their predictions. Thus, the comparison of their errors is more reliable than using different testing sets for each model.

## A.2 Linear models in MS Excel

Models are used to explain and predict data. One of the most common type of models is the linear model. The main assumption of this model is that the relationship between the variables in the data can be explained by linear functions. A linear function between two variables describes a line (hence the name linear), linear dependency between three variables describes a plane, linear functions between more than three variables describe linear manifolds also known as *hyperplanes*.

You probably remember that a straight line is described by the following formula:

$$y = a + bx \ ,$$

where $x$ and $y$ are the pair of variables, and $a$ and $b$ are the parameters of the line. Variable $x$ is sometimes referred to as the independent or the input variable, while $y$ as the dependent or the output variable (the one we are trying to explain or predict based on $x$). Of course here, the choice which is the input and which is the output is purely symbolic (we can always invert the linear function $x = (y - a)/b$).

Parameter $a$ is called the *intercept*, and it defines the point at which the line crosses the $y$ axis (i.e. when $x = 0$). Parameter $b$ is called the *slope* (or the *gradient*), and it defines how steep the line is. Positive $b$ means that the line goes up (positive trend), while negative $b$ means that the line goes down (negative trend). Parameters $a$ and $b$ uniquely define the line — when the numbers $a$ and $b$ are known, we can find the value of $y$ for any value of $x$ (this is how we make the prediction — compute $y$ for the desired value $x$).

In order to build such a linear model, we need to know $n$ data points — pairs of $x$ and $y$ values: $(X_1, Y_1)$, $(X_2, Y_2)$,..., $(X_n, Y_n)$. Using this data, we can find intercept $a$ and slope $b$ of a line that 'fits' these points as good as possible. Here, we understand a good fit in terms of a small average error. The most simple way of building the linear model is the *mean square linear regression*, and MS Excel has several functions that are based on this method. These functions are `INTERCEPT` (to compute $a$ based on several known data points) and `SLOPE` (to compute $b$), and they can be used for a linear model between two variables. You are advised to refer to the following website:
`http://www.bized.ac.uk/timeweb/crunching/crunch_analysis_illus.htm`

which has good instructions on how to use these functions in MS Excel. In addition, MS Excel

provides a function `FORECAST`, which combines the computations of $a$, $b$ and the forecast $y = a + bx$ in one step.

A linear function of multiple variables can be used to create a linear model for several variables. The linear function of $m$ input variables defines a hyperplane, and it is described by the following formula

$$y = a + b_1 x_1 + \cdots + b_m x_m$$

As you can see, the only difference here is that now we have $m$ slopes: $b_1$, $b_2$,...,$b_m$. To compute these parameters, we need at least $n = m + 2$ data points

| $x_1$ | $x_2$ | $\cdots$ | $x_m$ | $y$ |
|---|---|---|---|---|
| $X_{11}$ | $X_{12}$ | $\cdots$ | $X_{1m}$ | $Y_1$ |
| $X_{21}$ | $X_{22}$ | $\cdots$ | $X_{2m}$ | $Y_2$ |
| $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $X_{n1}$ | $X_{n2}$ | $\cdots$ | $X_{nm}$ | $Y_n$ |

This is why datasets should contain at least as many cases (rows) as the number of variables (columns) plus one.

The MS Excel Data Analysis package has the multiple **regression** function that computes parameters $a$ (the intercept) and $b_1, \ldots, b_m$ (the slopes) based on the given dataset. These parameters can be used to build a linear model with $m$ input variables as follows.

The regression function will ask you to input the data range of your spreadsheet containing the data (usually your training set). If everything is done correctly, the output of the regression function is a table containing values of the intercept and the slopes. You can copy these values and use them in the linear equation to value of the output variable. Note that if you are testing model's prediction, then you should compute the parameters using only the training set, but test the model's forecast on the testing set using the same parameters.

Below are the steps that will help you to create multiple regression model.

1. Prepare a column somewhere in your spreadsheet to store the parameters of the multiple regression model:

   $$\begin{matrix} a \\ b_1 \\ \vdots \\ b_m \end{matrix}$$

   It is advised to use a column because the Data Analysis package outputs the values in a column, and it is easier to copy and paste the values.

2. In your main table, add an extra column to enter the multiple linear formula for the prediction. The formula is

   $$y = a + b_1 x_1 + \cdots + b_m x_m$$

   where $m$ is the number of input variables, $x_1, \ldots, x_m$. In MS Excel, you can use relative references to the cells on the same row. Parameters $a$, $b_1, \ldots, b_m$ are the values computed by the regression function in step 1. In MS Excel, you use absolute references to the cells in Step 1, where you will copy the parameters.

If you are using the model only to explain the data, then you can simply use the regression function to compute the parameters based on the whole data. However, if you want also to evaluate how well can the model predict new data, then you should only use the training set to compute the model's parameters, and then use the model to predict the values in the testing set. The following procedure can be used:

1. Enter the linear regression formula in the extra column of your table, as explained above.

2. To evaluate the precision of your model, enter additional formulae, such as the absolute error and the average error, as explained in Section A.4.

3. When the table is ready and contains all the necessary formulae (including the other models if you are using them), you can randomly select and move the **training set** from the table (use the BrainCel preprocessing menu). The remaining table will be your **testing set**.

4. Use the regression function of the Data Analysis package to find the model parameters based only on the training set.

5. Copy and paste the values of the parameters of the model into the column you have prepared earlier. Your table with the testing set should automatically compute the predictions and all the errors.

By examining the regression coefficients of the model (parameters $b_1, \ldots, b_m$), it is possible to evaluate how significant is the dependency between the input and the output variables. If the value of $b_i$ is close to zero, then it suggests that there is almost no linear dependency between variable $x_i$ and $y$. Remember, however, that lack of linear dependency does not yet imply independence (the variables can be related non-linearly). Conversely, high absolute values suggest strong linear dependency. Moreover, positive values mean positive correlation, while negative values mean anti-correlation between $x_i$ and $y$. When making conclusions about the dependency, remember that correlation does not imply causation.

### A.3 Neural networks as models

As you will learn in the course, artificial neural networks can be trained and learn the relationship between the input and the output variables. Thus, you can train a neural network on your data and use it as a model. Although each neuron in the network can implement a linear model, a network of several neurons with at least one hidden layer can simulate non-linear relationships. This can be an advantage as many real-world processes are non-linear.

Recall that a neural network usually takes several inputs and has several outputs. If you use a dataset with $m$ input variables and only one output variable, then you need a net with $m$ inputs and only one output. We can consider the network as some function of $m$ variables that we shall use to model our data:

$$y = f(x_1, \ldots, x_m)$$

This function is more complex than the linear regression model, described above, and it has more parameters. Indeed, neural networks can have different topologies (i.e. the number of layers and the number of nodes in each layer); the nodes in the network can use different activation functions (e.g. linear, step or sigmoid); the network can be trained with different values of the training parameters (e.g. the learning rate).

A neural network model can be created in MS Excel using the BrainCel software. You can create a network (called 'expert' in BrainCel) with $m$ input nodes and one output node (to predict one variable based on $m$ input values). The number of hidden nodes is up to you to decide.

Before you can train the network on your data and use it for prediction, you need to name the regions in your spreadsheet where you have the training and the testing sets. Please, refer to Section A.1 for information on preparing the data in MS Excel.

After the spreadsheet has been prepared and the network created, you may train the network on the training set (the 70% of your data) and make the predictions using the testing set (the remaining 30%). If you have earlier created all the formulae in your table for computing the errors, then the spreadsheet should also compute all the results automatically.

Because the neural model has many parameters, the results will depend greatly on finding good values of these parameters, and there is a lot of room for experimentation. In order to achieve better results with neural networks (e.g. small average errors of prediction), try to experiment with the following parameters:

- Network topology (i.e. the number of nodes in the hidden layers);
- Different transfer functions (linear or log);
- Different learning schedule (i.e. the learning rate parameter, the number of training cycles).

If the network predicts the data well, then it means it has learnt the relationships between the variables. You can evaluate these relationships by examining the weights of individual nodes. Like the regression coefficients in linear models, weights that have high absolute values suggest high dependency between the variables. For example, if variable $x_i$ is connected to $i$th input of a particular node, and weight $w_i$ corresponding to this variable has high absolute value after training, then it means that variable $x_i$ has significant influence on the output of the node. This, however, does not mean yet that variable $x_i$ is related significantly to the output variable $y$. This is because in neural networks the input variables are not connected directly to the output variables (as in linear models), but there are nodes of the hidden layers between the input and the output variables. For this reason, it is more difficult to understand the dependency between the variables in neural network models, even if they produce good results. Sometimes, however, it is possible to say which variables do not have significant influence. Indeed, if some weights after training have values close to zero, then it means that the variables connected to them have little if no influence on the output variables.

### A.4 How to evaluate the models

The quality of the model can be assessed by computing the error between the known data ($y$) and model's prediction of this data ($f(x)$). The difference between the actual and the predicted value can be use to compute the *absolute error*:

$$c(y, f(x)) = |y - f(x)|$$

Here $|\cdot|$ means the absolute value (i.e. the positive part of the number). In MS Excel, you should use function `ABS` of the difference between the data and the model's forecast. The absolute error measures the distance between the data and the models' prediction (distance is always positive, hence the use of absolute values).

Sometimes it is convenient to show the error as a percentage of the data that the model predicted. You can do this by dividing the absolute error by the value of $y$ and then instructing MS Excel to format the cell as a percentage:

$$Error\% = \frac{|y - f(x)|}{y} 100\%$$

You can decide yourself which format to use. However, make sure that you are using and comparing the errors represented in the same format.

In order to understand how the model performs in a long term, we need to compute the errors for many cases and then take its average (the mean error). The *average absolute error* over $k$ predictions is computed as

$$Mean(|y - f(x)|) = \frac{|y_1 - f(x_1)| + \cdots + |y_k - f(x_k)|}{k}$$

In MS Excel, you can use function `AVERAGE` to find the mean error. By computing and comparing the average errors of different models you can evaluate which models perform better (i.e. make better predictions).