MONASH University

Information Technology

FIT5186 Intelligent Systems

Lecture 5

Classification and Prediction with Case Studies

Learning Objectives

- Understand
  - the techniques and methodologies used in NN models via case studies of classification and prediction.
  - the analysis issues for solving classification or prediction problems using NN models
  - the data pre-processing techniques
  - the performance measures for testing NN models
- Be able to
  - pre-process the data for building NN models
  - conduct experiments for building better NN models
  - perform result analysis for building better NN models

Classification vs. Prediction

- **Classification** considers the features of an object in order to assign it to a pre-defined class.
  - Examples:
    - Spotting fraudulent insurance claims
    - Identifying high-value customers

- **Prediction** is the classification of *future events*.
  - Historical data is used to build a model that “explains” outputs for known inputs.
  - The model is applied to new inputs to predict future outputs.
    - Examples:
      - Predicting which customers won’t default on a loan.
      - Predicting which footy team will win this weekend.

Case Studies of Classification and Prediction

- This lecture examines two case studies to demonstrate the techniques and methodologies discussed in the previous lecture.
- The case studies are based on Smith (1999) Chapter 4 Case Studies, pp. 58-67.
  - **Case study 1: Classification**
    - Involves learning to classify loan applicants as good or bad credit risks.
  - **Case study 2: Prediction**
    - Involves learning to predict the daily exchange rate of the Australian dollar against the US dollar.
- Both case studies are constructed from the initial raw data sets, showing the required pre-processing and decisions in relation to NN architecture and parameters.

Analysis Issues

- For solving classification/prediction problems using MFNN models, the analysis issues include:
  - Best choice of input/output variables (and the number of input/output variables);
  - How much data to use (if a great amount is available);
  - Number of hidden neurons;
  - Choice of parameters (learning rate, momentum factor, activation function parameter, stopping criterion);
  - Test set extraction method;
  - Method of input pattern presentation (random/rotation);
  - Performance measure: classification accuracy rate, R^2, or RMS error.

Case Study 1:
Loan Applicant Classification

- The Problem:

  Given 1,000 *previous* loan applications at a bank, together with those applicants' subsequent classification as "good" or "bad" customers, can we learn to classify *new* applicants as "good" or "bad" based only on their applications?

  – i.e. can we learn which combinations of responses indicate a "good" credit risk, and which indicate a "bad" credit risk?

The Data

- See Smith (1999) Appendix C (pp. 150-152). See also Week 6 Tutorial.
  - Questions for credit application example (German data)
- credit.txt contains the original responses to these questions (1,000 rows = 700 good and 300 bad).
- The original data needs to be pre-processed, since neural networks require numerical data as inputs and the data needs to contain a sense of scale and order.
- credit.xls contains the pre-processed data (1,000 rows).

Pre-processing the Data

- Original data has 20 columns (variables).
  - 7 numerical data columns (i.e. age or credit amount) and
  - 13 qualitative (categorical) data columns (i.e. loan purpose)
- Qualitative (categorical) data
  - Nominal: no order (e.g. blood type: A, B, AB, O)
  - Ordinal: order (e.g. grade: fail, pass, credit, distinction, HD)
- The categorical data needs to be converted to numerical data.
  - usually achieved by translating a single column categorical data into multiple columns with binary variables.
  - **1-out-of-N Encoding** technique.

    A variable with N possible values of no inherent order is replaced with N variables of binary values.

Pre-processing the Data (continued)

- As such
  - Question 4 (credit purpose) becomes
    - 1 0 (new car), 0 1 (used car) and 0 0 (other)
  - Question 10 (other debtors/guarantors)
    - 1 0 (none), 0 1 (coapplicant), 0 0 (guarantor)
  - Question 15 (Housing type) becomes
    - 1 0 (renting), 0 1 (owner), 0 0 (for free)
  - Question 17 (employment type) becomes
    - 1 0 0 (unemployed), 0 1 0 (unskilled), 0 0 1 (skilled), 0 0 0 (highly skilled)

Pre-processing the Data (continued)

- Once these categorical data columns have been replaced, the processed data file credit.xls contains 24 columns.
- The known classification of each customer in the data set as "good" or "bad" is coded as
  - (a) Using a single column (with one output neuron) as:

    credit class: **1 (good) or 2 (bad)**
    - 0 (good) or 1 (bad) is used in the further experiments by Smith (1999) → see Slide 13.
  - (b) Using 2 columns (with two output neurons) as

    good: 1 0 and **bad: 0 1**
- See Table 4.1 (p. 60) and credit.xls.

Constructing the MFNN Model

- Input variables: 24
- Output neurons:
  - 1 (with a value of 1 for good or 2 for bad)
  - 2 (with a value of (1 0) for good or (0 1) for bad)
  - It is generally better to use the same number of output neurons as the number of classes for classification problems.
- Hidden neurons: 30 or ?
- Experiments with different architectures and model parameters.

Experiment 1

- MFNN architecture
  - 24 Input variables
  - 30 hidden neurons
  - 2 output neurons ((1 0) for good and (0 1) for bad))
- Learning rate $c = 0.1$
- Momentum factor $\alpha = 0.1$
- Initial weights: 0.3
- 20% of the data for the test (out-of-sample) set extracted randomly
- Input patterns presented to the MFNN model randomly
- Stopping criteria
  - Measure network performance on the test set every 200 epochs. In NeuroShell 2, save training on “best test set” and set Calibration interval = 200.
  - Stop if error approaches 0, or if error has not improved within 20,000 epochs.
- Result analysis – see Table 4.2 (p. 61).

Experiment 2

- Use a subset of input variables which contribute more to the output of the MFNN.
- NeuroShell 2 has the ability to reveal the contributions of the input variables to the output.
- Only 10 input variables which make significant contributions to the output are used, including 1, 2, 3, 5, 7, 10, 15, 17, 18 and 19.
- Use the same architecture and parameters.
- Result analysis – see Table 4.3 (p. 62).

Further Experiments

- Vary the number of hidden neurons.
- Change the learning rate and momentum factor.
- Use a single output neuron (0 for good and 1 for bad).
- Alter the discrete decision threshold value

  (bad if greater than 0.5, 0.25 or 0.1).
- Result analysis – see Tables 4.4 – 4.8 (pp. 62-63).

Case Study 2:
Foreign Exchange Rate Predication

- The Problem:
  - Predict the tomorrow's (daily) spot exchange rate between the Australian dollar (AUD) and the US dollar (USD), using today's and previous days' spot rates, some moving averages* and commodity prices.
  - Predict the movement of the exchange rate (i.e. the AUD/USD will increase or decrease, regardless of the magnitude of the movement).
  - These results will help make buying and selling decisions.

* Moving averages are used to smooth out short-term fluctuations, thus highlighting longer-term trends.

The Data

- 1 January 1993 to 24 April 1998.
- 1377 daily data points, with values ranging from 63.26 to 81.79.
- Both the minimum and maximum values occur during the 1997-1998 potion of the data.
- To ensure good generalisation of the model performance, the test set extracted should be statistically similar to the training set.

Constructing the MFNN Model

- 9 input variables:
  - Today's AUD/USD spot rate
  - 1 day lag of AUD/USD spot rate
  - 2 day lag of AUD/USD spot rate
  - 3 day lag of AUD/USD spot rate
  - 4 day lag of AUD/USD spot rate
  - 5 day moving average of daily AUD/USD spot rate
  - 10 day moving average of daily AUD/USD spot rate
  - Today's All Ordinaries Index
  - Today's Gold spot rate
- 1 output neuron: AUD/USD spot rate
- Hidden neurons: the default formula used by NeuroShell 2 $\frac{1}{2}(N + K) + \sqrt{P}$

Experiments

- 9 experiments with different architectures and test sets – see Table 4.9 (p. 66).
- Learning rate *c* = 0.05; Momentum factor α= 0.5.
- Each experiment was trained for 100 epochs.
- Result analysis – see Table 4.10 (p. 66).
- Performance measure used is the *coefficient of multiple determination* R^2 (*R* Squared).
- See FIT5186_Lecture 5_R2.pdf for the meaning of R^2 and another performance measure, the *RMS error*.

Experiments (continued)

- A better result is achieved if the MFNN model is used to predict whether or not the spot rate will move up or down.

  - See Table 4.11 (p. 67).

- An alternative approach – formulating as a classification problem.

  Construct an MFNN model to learn to classify the movement of the daily spot rate as up or down.

NN Application Papers

- Flitman, A.M. (1997). Towards analysing student failures: neural networks compared with regression analysis and multiple discriminant analysis. *Computers & Operations Research,* 26(4), 367-377.
  - Compares NN to 2 other techniques.
  - Preprocessing and normalising of input data.
  - Tries different architectures, and reports the optimal configurations based on best R^2.
  - Describes clearly the extraction method (random), and training decisions.
  - Reports final relevant inputs.

NN Application Papers (continued)

- Jagielska, I. and Jacob, A. (1993). A neural network model for sales forecasting.
  - Predicts sales levels for Tatersall’s
  - Sales level depends on draw type (specials), jackpot level, prize level, economy, advertising.
  - Architecture = 28 input variables, 1 hidden layer, one output neuron (=sales $’s for current draw).
  - Training set: July 89 - December 91
  - Test set: January 91 - September 92
  - Correlation = 0.979; average error = 3.48%
  - NN better compared to expert predictions.

NN Application Papers (continued)

- Thiesing, F.M., et al. (1995). Short term prediction of sales in supermarkets.
- Kong, J.H.L. and Martin, G.P. (1995). A backpropagation neural network for sales forecasting.
- Lai, H.-H., Lin, Y.-C. and Yeh, C.-H. (2005). Form design of product image using grey relational analysis and neural network models.
- Yeh, C.-H., Lin, Y.-C. and Chang, Y.-H. (2007). A neural network approach to website design.
- Lin, Y.-C., Yeh, C.-H. and Wei, C.-C. (2013). How will the use of graphics affect visual aesthetics? A user-centered approach for web page design.
- Lin, Y.-C. and Yeh, C.-H. (2015). Grey relational analysis based artificial neural networks for product design: A comparison study.

NN Application Papers (continued)

- Have a look at papers published in relevant journals and conferences.
  - Monash Library online databases, e.g. IEEE Xplore or Scopus.
- You should be looking for
  - How they describe the problem;
  - How they discuss the relevant issues;
  - Their decisions and choice of analysis issues;
  - Their conclusions.
- No matter what topic you choose, similar work probably has already been done. Read it.

Other Classification Methods

- k-nearest neighbour
- Decision trees
- Support vector machines
- Case-based reasoning
- Fuzzy logic
- Rough set theory

Week 5 Tutorial

- NeuroShell 2 on-line Tutorial Example 2
  – NYSE prediction
  – Larger and more complex
- NeuroShell 2 - **Advanced** Neural Networks System
  Same as Beginners' system but with extra features
  – Range of network designs (architectures): beginners' just has 3-layer backpropagation network.
  – Ability to determine contributions of inputs.
    - Apply Neural Network/Contribution Factors
  – Graphing facilities.
  – Graph pattern variables and/or error convergence.
  – Symbol translation for inputs.
  – If then else rules.