CN550 Class Project: Benchmark Problems

Circle-in-the-Square (CIS)

Description
The CIS benchmark is a two-dimensional, noise-free, two-category classification task. A circle with area 0.5 (radius slightly less than 0.4) is drawn in the center of a unit square.

Learning systems, given training points \((a_1,a_2)\) in random order, learn to distinguish between two classes: inside the circle (IN - class 1), and outside the circle (OUT - class 0).

Problem \textit{CIS-small} trains on 100 randomly chosen points, and problem \textit{CIS-large} trains on 1,000 points.

The test set is a regular 100 \times 100 grid.

Reference

Data
CIS exemplar structure in the datasets

<table>
<thead>
<tr>
<th>Field #</th>
<th>Description</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(a_1) (x)-coordinate</td>
<td>real-values in ([0,1])</td>
</tr>
<tr>
<td>2</td>
<td>(a_2) (y)-coordinate</td>
<td>real values in ([0,1])</td>
</tr>
<tr>
<td>3</td>
<td>IN/OUT of circle (output)</td>
<td>binary (1 IN / 0 OUT)</td>
</tr>
</tbody>
</table>

Word (text) files:
- CIS-small (train 100 points) \(\text{http://cns.bu.edu/~gail/550\_CIS\_train\_100inputs.doc}\)
- CIS-large (train 1,000 points) \(\text{http://cns.bu.edu/~gail/550\_CIS\_train\_1000inputs.doc}\)
- CIS-small (test 10,000 points) \(\text{http://cns.bu.edu/~gail/550\_CIS\_test\_10,000inputs.doc}\)

Matlab and .txt files:
\(\text{http://cns.bu.edu/cn550/classproject/data/CIS/}\)

\text{http://cns.bu.edu/~jdoon/papers/550\_evalvis.doc}\)

function \texttt{cis\_vis.m}
Pass the Matlab function \texttt{cis\_vis(x)} a 10,000 element vector of 0 (OUT) and 1 (IN) values representing your classifier’s predictions on the 10,000-point test set. The order of the vector must correspond to the order of the test points in the file \texttt{550\_CIS\_test\_10,000inputs.doc}. 

CIS response plot visualization and ROC curves
\(\text{http://cns.bu.edu/~jdoon/papers/550\_evalvis.doc}\)
Financial Prediction

Description: In this benchmark, classifiers learn to predict whether the price of the exchange traded fund Diamonds Trust (DIA, which tracks the Dow Industrial Average) will go up or go down during the month of January, based on features extracted from trading data of the previous year. These features were constructed nearly exactly from those specified in Versace et al. (2004). The training data consist of 192 daily features from the year 2005 (237 days), and the testing data consist of features from January 2006 (21 days).

Binary classification problem: Predict whether the fund will go UP or DOWN on the following day.

Custom investment strategies: Try to make as much money as possible!

Reference:
Versace, Massimiliano, Bhatt, Rushi, Hinds, Oliver, & Shiffer, Mark (2004)
Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks.

Data
In text format (Word document):
http://cns.bu.edu/~gail/550Financial_Test_Jan2006_21days.doc
In Matlab, text, and text comma formats:
http://cns.bu.edu/cn550/classproject/data/FinancialDataset

See the sites below for results obtained by previous CN550 and CN710 classes:
CIS
Financial

For details, see the end of this document.
Evaluation and Visualization

Matlab routines
http://cns.bu.edu/~jdoon/papers/550_evalvis.doc (Word document) or
http://cns.bu.edu/cn550/classproject/contrib/

Evaluation criteria
- % correct
- C-index
- Response plot (CIS)
- Investment Strategies (Financial)
  Strategy 1: Custom measure which calculates how much money you would gain (or lose) if you were to buy one share every day your learning system predicts UP and sell (or short) one share every day your system predicts DOWN.
  \[ C = [P(d) - P(d-1)]h(d) \]
  where \( P(d) \) is the price of DIA on day \( d \), and
  \( h(d) \) is the hypothesis (output) of the classifier on day \( d \) (–1 for predicting DOWN, 1 for predicting UP).
  See http://cns.bu.edu/cn550/classproject/contrib/fin_cost_function.m
  Be sure to convert 0 (= DOWN in the dataset) to –1 for this calculation.

- Confusion matrix.

<table>
<thead>
<tr>
<th>confusion matrix</th>
<th># actual +</th>
<th># actual –</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td># predicted +</td>
<td>TP: true +</td>
<td>FP: false +</td>
<td>TP + FP</td>
</tr>
<tr>
<td># predicted –</td>
<td>FN: false</td>
<td>TN: true</td>
<td>TN + FN</td>
</tr>
<tr>
<td>total</td>
<td>TP + FN</td>
<td>FP + TN</td>
<td>whole test set</td>
</tr>
</tbody>
</table>


A sample ROC curve generated using the Matlab code. Real values (i.e., not class labels) predicted by the classifier and the true class labels are used to compute the ROC curve. The dashed line shows the ROC convex hull.

The ROC curve is parameterized by an output threshold \( \gamma \in [0,1] \).
A system predicts + if its positive prediction output \( \sigma > \gamma \).
\[ \gamma = 0: \text{All predictions are +} \]
\[ \gamma = 1: \text{All predictions are –} \]

The c-index \( \in [0,1] \) is the area under the ROC curve. Chance prediction produces a diagonal ROC curve and a c-index of 0.5.