Comparative Analysis of Data Mining Algorithms for Predicting Inpatient Length of Stay

Peng Liu  
Shanghai University of Finance and Economics

Elia El-Darzi  
University of Westminster

Christos Vasilakis  
University of Westminster

Panagiotis Chountas  
University of Westminster

Wei Huang  
University of Westminster

See next page for additional authors

Follow this and additional works at: http://aisel.aisnet.org/pacis2004

Recommended Citation

Liu, Peng; El-Darzi, Elia; Vasilakis, Christos; Chountas, Panagiotis; Huang, Wei; and Lei, Lei, "Comparative Analysis of Data Mining Algorithms for Predicting Inpatient Length of Stay" (2004). PACIS 2004 Proceedings. 86.
http://aisel.aisnet.org/pacis2004/86

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2004 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
Comparative Analysis of Data Mining Algorithms for Predicting Inpatient Length of Stay

Peng Liu¹, Elia El-Darzi², Christos Vasilakis², Panagiotis Chountas², Wei Huang², Lei Lei¹

¹Department of Information Systems, Shanghai University of Finance and Economics, Shanghai, P.R. China, 200433
Email: liupeng@mail.shufe.edu.cn
²Health Care Computing Group, School of Computer Science, University of Westminster, London, Northwick Park, HA1 3TP, UK
Email: eldarze@wmin.ac.uk

Abstract

It is well documented that efficient measurement of inpatient length of stay (LOS) can greatly enhance the planning and management of hospital resources. This paper describes the use of a data mining approach to predict the LOS using historical data. The data are first pre-processed using attribute-aggregation, attribute-generalization and relevance analysis procedures to reduce the numbers of attributes and compress the training data set. Two classifiers, Naïve Bayesian and decision tree, are used for predicting LOS. Further a new concept, named “prediction profit”, is introduced to compare the performance of these classifiers. We demonstrate that this new measure of performance can better discriminate between alternative classifiers when compared to existing ones such as the overall accuracy. The empirical tests show that the proposed data mining approach for predicting LOS is a viable one.

Keywords: Data Mining, Inpatient Length of Stay, Decision Tree, Naïve Bayesian Classifier

1. Introduction

The provision and planning of hospital resources has always been a matter of great interest for researchers and practitioners alike. One of the main concerns in this area is the measurement of flow of patients through hospitals and other health care facilities. Coté (2000) describes patient flow from two perspectives, the clinical and the operational. From the clinical perspective, patient flow represents the progression of a patient’s health status. From the operational perspective it represents the movement of patients through a set of locations in a health care facility. A measure frequently used in describing operational patient flow is the average length of stay which although easy to quantity and calculate but its simplicity often leads to suboptimal decision as in many cases the data is skewed (Millard, 1994).

The problem of measuring LOS becomes more complicated when applied to hospital departments that deal with heterogeneous patient populations. Existing modelling frameworks have been proposed to categorize the data into short, medium and long stay (Harrison and Millard, 1991; Harrison, 1994). These methods, although capable of highlighting the interaction between the different streams of patient flow, are based on single
day bed censuses and thus exclude a large number of the available data points. Data mining algorithms start to emerge as viable tools to address this important heterogeneous patient population problem (Isken and Rajagopalan, 2002; Harper, 2002). These tools have been very successful in addressing other areas of healthcare such as diagnosis, prognosis or overall patient management. The major attractive features of these techniques (Kononenko et al., 1997) include good performance, the ability to deal with missing data and to process noisy data (errors in data), and the ability to provide knowledge that can guide decisions.

In this paper, we propose a framework for evaluating the suitability of two different data mining models for the prediction of inpatients LOS. Two data sets are used as running examples. A new concept named “prediction profit” is introduced to discriminate between the data mining classifiers. Finally, we report our results and conclusions.

2. The Data Sets
The data mining algorithms proposed in this paper are tested on two data sets, which are summarized below.

2.1 Clinics Data Set
The Clinics data set contains data from a clinical computer system that was used, between 1994 to 1997, for the management of patients in the Department of Geriatric Medicine at St. George’s Hospital in London (Marshall, 2001). It contains 4722 patient records including patient demographic details, admission reasons, discharge details, outcome and LOS. Patient LOS has an average of 85 days and a median of 17 days. For further analysis, the duration of stay variable was categorized into three groups: 0-14 days, 15-60 days and 61+ days. The boundaries of the duration of stay groups were chosen in agreement with clinical judgement. These categories were chosen to model the stages of patient survival in hospital. The first category is for patients in acute care, admitted in a critical condition and only staying in hospital for a short period of time. The second category includes patients who undergo a period of rehabilitation. The third category refers to the 11% of patients who stay in hospital for a long period of time.

2.2 Stroke Data Set
This data set originates from the English Hospital Episode Statistics (HES) database and concerns all finished consultant episodes (FCE) of stroke patients, aged 65 and over, discharged from all English hospitals between April 1st, 1994 and March 31st, 1995 (105,765 FCE’s) (Vasilakis and Marshall, 2003). A spell is qualified as stroke if contains a stroke related diagnosis code anywhere in the diagnostic chain (stroke related diagnoses are between codes 430 and 438 in the International Classification of Diseases, Injuries and Causes of Death-Revision 9, ICD-9). No information that identified individual patients was supplied. Patient LOS has an average of 14 days and a median value of 7 days. The variable LOS is categorized into six groups: 0 days, 01-06 days, 07-20 days, 21-41 days, 42-97 days and 98+ days. These categories, as before, are chosen for modelling the stages of patient survival in hospitals and are based on clinical and managerial judgement.
3. Attribute-aggregation and Generalization-based Data Mining Application

The proposed framework is made up of the following four steps.

a) Attribute-aggregation to compress the attribute numbers of training data set.
b) Attribute-oriented generalization to compress the training data (tuples).
c) Relevance analysis to remove irrelevant data attributes.
d) Data mining algorithms to estimate LOS.

3.1 Aggregation of Attributes

Health care data often have many characteristics and attributes. Some of these attributes are irrelevant and can be removed and some are closely related and thus can be grouped. A few of the attributes in Clinics data set are illustrated in Fig.1.

For example, in the Clinics data set the Barthel score is composed of various indices that assess the patient’s ability to do every day activities and their dependence on others for support (Mahony and Barthel, 1965). The score consists of ten different elements - feeding, grooming, bathing, mobility, stairs, dressing, transfer, toilet, bladder and bowels. Each patient is assessed using scales of dependency range from 0 to 3. A low score generally means a high level of patient dependency on others while a high score reflects those patients who are independent and require little assistance from medical staff.

In order to simplify this scoring system for patient dependency, a grouped Barthel score is introduced as in (Marshall, 2001). This revised scoring system is defined as follows: heavily dependent (Barthel score ≤1), very dependent (Barthel score: 2-10), slightly dependent (Barthel score: 11-19) and independent (Barthel score ≥20).

The application of attribute-aggregation procedure has reduced the total number of attributes in the original data set without loosing any critical information. The undesirable dependency between these attributes is also greatly reduced.

3.2 Attribute-oriented Generalization

In many real-life applications such as health care, data mining algorithms are often applied using large training data. Applying attribute-oriented generalization prior to classification
can substantially reduce the computational burden of this data-intensive process.

The attribute-oriented generalization method compresses the data set by replacing the attribute values with more general information in a form of a concept hierarchy. This procedure assumes that each attribute can be generalized independently of others thus, allowing large data sets to be efficiently generalized (Carter and Hamilton, 1998).

A tree structure is used in this method, with the most general concept is placed at the root of the tree while its particulars constitute the leaves (See Fig.2). For example in the Strokes data set, the attribute “Diag” represents the patient diagnostic codes and the attribute “Age” represents the age of patients. In the case of continuous attributes the leaves or nodes in concept hierarchies are represented as a range of values, as shown in Fig.2. The purpose of providing a set of concept hierarchies is to summarize the training data set.

![Figure 2. Concept hierarchies for attributes “Diag” and “Age”](image)

Attribute-oriented generalization can be applied to either remove or generalize attributes. If an attribute has a large number of distinct values but no higher level concept associated with it, or its higher level concept has been represented by another attribute, then it can be removed. On the other hand, if an attribute has a corresponding concept hierarchy then its value can be replaced by relevant higher level attribute.

The basic ideas of attribute-oriented generalization can be illustrated using the Stoke data set. In Table 1 the attribute “DHA” represents the District Health Authority of treatment, while the attribute “RHA” represents the Regional Health Authority. For instance the attribute
“Yorkshire RHA” is made up of seven DHA values. Hence, the attribute “DHA” can be removed, as it can be identified by its higher level values (RHA). The attribute “ProdMut” provides the first three characters of healthcare provider code. This attribute can be removed from the training set due to the large number of distinct values without any higher level concept.

The generalized result using the concept hierarchies is shown in Table 2. Identical tuples in Table 1 were merged while collecting the count information. The “count” column counts the number of ungeneralized tuples summarized in the generalized tuples. The generalization process is limited by an attribute threshold, which specifies the maximum number of distinct concept values of an attribute. If the number of distinct values of an attribute is less than or equal to this threshold, then further generalization of this attribute is halted. In our data set, after some initial experimentation, the threshold values were set in the range of 3-14. After generalization of the Stroke data set the number of tuples is reduced from 105,765 to 22,736.

### Table 1. Sample data

<table>
<thead>
<tr>
<th>Age</th>
<th>DHA</th>
<th>RHA</th>
<th>ProdMut</th>
<th>Diag</th>
<th>…</th>
<th>LOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>65</td>
<td>J25</td>
<td>Wessex RHA</td>
<td>RN1</td>
<td>43301:OBAWI</td>
<td>…</td>
<td>13</td>
</tr>
<tr>
<td>83</td>
<td>B31</td>
<td>Yorkshire RHA</td>
<td>RBN</td>
<td>4351:VAS</td>
<td>…</td>
<td>6</td>
</tr>
<tr>
<td>71</td>
<td>C03</td>
<td>Trent RHA</td>
<td>RNT</td>
<td>43300:OBAWOI</td>
<td>…</td>
<td>14</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

### Table 2. Training data set: generalized to an intermediate level concept, with count

<table>
<thead>
<tr>
<th>Age</th>
<th>RHA</th>
<th>Diag</th>
<th>…</th>
<th>LOS</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>65-74</td>
<td>Wessex RHA</td>
<td>433:P.O.</td>
<td>…</td>
<td>07-20</td>
<td>7</td>
</tr>
<tr>
<td>75-84</td>
<td>Yorkshire RHA</td>
<td>435:T.C.I</td>
<td>…</td>
<td>01-06</td>
<td>104</td>
</tr>
<tr>
<td>65-74</td>
<td>Trent RHA</td>
<td>433:P.O.</td>
<td>…</td>
<td>07-20</td>
<td>3</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td></td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

### 3.3 Relevance Analysis

The third step of this process is attribute relevance analysis, which aims to further reduce the size of the training data. Although there are a number of techniques for relevance analysis, in our paper the applied statistical property, named “information gain”, has been used for selecting the best attributes (Kamber et al., 1997).

One problem with information gain is that it favours attributes with many divisions and thus may lead to overfitting. This problem, however, can be overcome by taking into account the cardinality of each division. In this case the GainRatio will be used as opposed to information gain.

To apply the relevance analysis procedure, we first calculate the information gain for each attribute and then the GainRatio for those attributes with above average gain. The attributes are subsequently rearranged in descending order according to their GainRatio and
information gain values. Only the first \( n \) attributes, which are thought to be the most relevant, are retained, where \( n \) is a user-defined threshold. This threshold value can also be defined as a percentage of the original attribute numbers.

### 3.4 Data Mining Applications

In this section we give a brief introduction to the Naive Bayesian classifier and the decision tree algorithm, which are used to build prediction models using the reduced data. These two algorithms are implemented in practical experiments by a data mining system called weka (Mitchell, 1997; Witten and Frank, 2000), developed at the University of Waikato in New Zealand.

#### 3.4.1 Naive Bayesian Classifier

The Naive Bayesian classifier is a simple and efficient classifying algorithm. Sometimes it outperforms other data mining techniques such as decision tree and neural network algorithms (Mitchell, 1997; Witten and Frank, 2000). Naive Bayesian classifier is based on Bayes Theorem which allows us to derive the posterior probability \( P(C|X) \), from the prior probability \( P(C) \) and \( P(X) \) and the conditional probability \( P(X|C) \) by using the following relation.

\[
P(C|X) = \frac{P(X|C)P(C)}{P(X)}
\]

Given a training set, the Naive Bayesian classifier consists of deriving the probability \( P(C_j) \) of each class \( C_j \) and \( P(X_i|C_j) \) for each of the attribute values \( X_i \) and \( P(X_i) \). Assuming that the contribution by all attributes is independent, the following posterior probability \( P(C_j|X) \) can be used for each class:

\[
P(C_j|X) = \frac{P(X|C_j)P(C_j)}{P(X)}
\]

where

\[
P(X|C_j) = \prod_i P(X_i|C_j)
\]

A new tuple is classified into the class with the maximum probability. As \( P(X) \) is a constant for each class, we only need to calculate the maximum value of \( P(X|C_j)P(C_j) \) from the training data set.

#### 3.4.2 Decision Tree

The decision tree approach is another useful tool for classification problems. With this method, a tree is constructed to model the classification process. As applying the tree to a new data set is relatively straightforward, most applications and research tends to focus on the decision tree construction. In our application we adopt the widely recognized decision tree algorithm C4.5 (Quinlan, 1992).

The main idea of constructing a decision tree can be expressed recursively as follows: Firstly, choose an attribute to be placed at the root node and then make one branch for each possible
value. The process can be repeatedly applied for each branch. The creation of the tree is completed after the training data at a node is perfectly classified. The Gain and/or GainRatio criteria are used to select the optimal attribute for splitting. The attribute with the maximum GainRatio value is selected as the best attribute for splitting.

4. Accuracy Rate and Prediction Profit

After a classifier has been successfully built, a widely acceptable measure for classifier efficiency is the prediction accuracy. Prediction accuracy measures the correctness of the classifier when used to predict new values (Mitchell, 1997).

Prediction accuracy can be applied at two levels: class level and model level. In the class level the prediction accuracy, $\alpha_c$, can be obtained from the following relation and ideally should be higher than 50%.

$$\alpha_c = \frac{\text{Number of correctly categorized patients in the class}}{\text{Total number of patients in the class}} \times 100\%$$

The model-level prediction accuracy or overall accuracy rate, $\alpha$, is expressed below. Again high overall accuracy rate is desirable (Lim et al., 2000).

$$\alpha = \frac{\text{Number of correctly categorized patients}}{\text{Total number of patients}} \times 100\%$$

The overall accuracy, $\alpha$, is a relative value, which is quite different from the class-level accuracy, $\alpha_c$. In fact the class probability distribution can greatly influence the value of $\alpha$. Because the overall accuracy is used for comparing and selecting mining algorithms, it can be occasionally misleading due to the influence of the class probability distribution on the value of $\alpha$. In order to eliminate this influence the new concept “prediction profit” is introduced below.

Definition: let $\alpha$ be the overall accuracy of a classifier, $\beta_i$ is the $i$th class distribution where $\sum_i \beta_i = 1$, $\beta = \max \{\beta_i\}$ is the majority class probability distribution. The prediction profit, $p$, is then expressed as follows:

$$p = \frac{\alpha - \beta}{\beta} \times 100\%$$

Prediction profit provides the rate of information derived from a classifier. It is clear that for a classifier to be useful or effective $p$ must be positive, i.e. $\alpha$ must be greater than $\beta$.

To illustrate the importance of prediction profit, we have applied it to re-measure the effectiveness of four different neural network (NN) models introduced to predict LOS (Walczak et al., 2003). The initial results are presented in Table 3 with the added prediction profit as the last column. Although the overall prediction accuracy rates are as high as 71.94% in the first model and 71.8% in the last model, the benefit of the information gathering is as low as 0.2% and 0% respectively. The prediction profits for the second and
third models are negative. By using prediction profit as opposed to overall accuracy, one can conclude that the overall results are not good at all. The analysis in Table 3 is used to illustrate the importance of the prediction profit as a tool to compare the performance of different methods. Although in many other applications NN has proved to be a powerful tool in health care data mining, in this case NN did not perform too well.

Table 3. Neural network prediction of acute pancreatitis LOS (Walczak et al., 2003)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Network Type</th>
<th>1-7 days</th>
<th>8-14 days</th>
<th>&gt;14 days</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Backpro (1 H layer)</td>
<td>86.11</td>
<td>29.41</td>
<td>45.5</td>
<td>71.94</td>
</tr>
<tr>
<td>2</td>
<td>Backpro (2 H layer)</td>
<td>25.00</td>
<td>100</td>
<td>0.00</td>
<td>34.85</td>
</tr>
<tr>
<td>3</td>
<td>RBF</td>
<td>68.63</td>
<td>16.67</td>
<td>50.00</td>
<td>57.74</td>
</tr>
<tr>
<td>4</td>
<td>Fuzzy ARTMAP</td>
<td>100</td>
<td>0.00</td>
<td>0.00</td>
<td>71.80</td>
</tr>
<tr>
<td>N/A</td>
<td>Patient distribution</td>
<td>71.8</td>
<td>16.9</td>
<td>11.3</td>
<td>N/A</td>
</tr>
</tbody>
</table>

5. Practical Experiments
5.1 Experimenting with the Naive Bayesian Classifier
In these experiments, both Clinics and Stroke data sets are randomly partitioned for each run into two subsets: the training subset which constitutes 2/3 of the data set and the testing subset which constitutes the remaining 1/3. The procedure is repeated 10 times and the results are shown in table 4.

Table 4. Naive Bayesian classifier: experimental results

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Stroke</th>
<th>Clinics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
<td>Accuracy rate (%)</td>
<td>Prediction profit (%)</td>
</tr>
<tr>
<td>1</td>
<td>52.559</td>
<td>34.746</td>
</tr>
<tr>
<td>2</td>
<td>52.061</td>
<td>34.832</td>
</tr>
<tr>
<td>3</td>
<td>52.520</td>
<td>35.162</td>
</tr>
<tr>
<td>4</td>
<td>52.314</td>
<td>33.982</td>
</tr>
<tr>
<td>5</td>
<td>52.349</td>
<td>34.626</td>
</tr>
<tr>
<td>6</td>
<td>52.339</td>
<td>33.808</td>
</tr>
<tr>
<td>7</td>
<td>52.363</td>
<td>34.031</td>
</tr>
<tr>
<td>8</td>
<td>52.314</td>
<td>34.593</td>
</tr>
<tr>
<td>9</td>
<td>52.229</td>
<td>35.047</td>
</tr>
<tr>
<td>10</td>
<td>52.162</td>
<td>34.166</td>
</tr>
<tr>
<td>Average</td>
<td>52.321</td>
<td>34.499</td>
</tr>
<tr>
<td>Overall data set</td>
<td>52.425</td>
<td>34.617</td>
</tr>
<tr>
<td>Diff. (±%)</td>
<td>0.1988</td>
<td>0.3412</td>
</tr>
</tbody>
</table>

From Table 4 we observe that the accuracy rate and prediction profit in each run for the Stroke data set are very stable and reliable. This is due to the fact that this data set is very large. The average accuracy rate and the prediction profit of both data sets are very close to the overall values. It shows that the overall data set could be used to build and estimate the model directly in case of a very large data set. Since this method seems to work equally well for both data sets, results from smaller data sets can also be generalized.

5.2 Experimenting with the Decision Tree
The decision tree models were built using the reduced Clinics and Stroke data sets
respectively. Ten-fold cross-validation estimates of the performances were used (Witten and Frank, 2000). Initial results for Stroke data set were very poor for the three groups 21-41 days (11.09%), 42-97 days (4.48%) and 98+ days (0.93%) this indicates that data quantity has a great influence on data mining results. We believe that both Naïve Bayesian and decision tree classifiers favour major classes. These three groups constitute a small percentage of the data set. Therefore, we decided to merge them in one group (21+ days). The prediction results for both classifiers for both data sets are summarised in tables 5 and 6.

Table 5. Decision Tree: experiment results on Clinics data set

<table>
<thead>
<tr>
<th>Models</th>
<th>Accuracy rate (%)</th>
<th>Prediction Profit (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0-14 days</td>
<td>15-60 days</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>49.10</td>
<td>69.20</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>35.60</td>
<td>76.50</td>
</tr>
<tr>
<td>Patient Distribution</td>
<td>39.52</td>
<td>49.03</td>
</tr>
</tbody>
</table>

Table 6. Decision Tree: experiment results on Stroke data set

<table>
<thead>
<tr>
<th>Models</th>
<th>Accuracy rate (%)</th>
<th>Prediction Profit (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 day</td>
<td>1-6 days</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>75.80</td>
<td>43.67</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>75.98</td>
<td>50.68</td>
</tr>
<tr>
<td>Patient Distribution</td>
<td>12.34</td>
<td>32.21</td>
</tr>
</tbody>
</table>

It can be seen that the decision tree model outperforms the Naïve Bayesian classifier in the Stroke data set. However, the Naïve Bayesian classifier outperforms the decision tree model in the Clinics data set. This can be attributed to the fact that the Clinics data set has some missing values and some noise. The two models produce very similar overall prediction accuracy rates, prediction profits and accuracy rates for each class. This implies that either one can in principle be the used to predict inpatient LOS. However, Naïve Bayesian is better suited for noisy data, while decision tree is more appropriate for “clean” data.

6. Conclusions

In this paper we have demonstrated the applicability and suitability of Naïve Bayesian and decision tree algorithms for the prediction of inpatient LOS. We have also shown that these methods performed much better than a commonly used Neural Network approach. The pre-processing procedures, i.e. attribute aggregation, generalization and relevance analysis are found to be very effective in reducing the size of the data sets without loosing any critical information. A new concept, the “prediction profit”, has been introduced to compare the performance of different prediction models. The advantage of using prediction profit alongside the overall accuracy is also highlighted. Overall, the prediction results of the Naïve Bayesian and decision tree algorithms are, in our views, acceptable. We believe that data mining tools can be used by hospital planners to estimate the likelihood of patients LOS at least in the short and medium stay groups. This is because these models seem to perform better when the number of samples in a group is reasonably high. These results, however, lead us to believe that using a larger data set would allow revealing results to be found using data mining techniques. Although the last LOS group of each data set constitutes a small percentage of the patient population, it represents the long stay patients. An optimal (or near
optimal) prediction of inpatient LOS in this group could lead to better overall capacity planning. Further work is needed to improve the accuracy rate of the groups with a small sample size by experimenting with larger data sets.

References
Marshall, A. H. "Bayesian Belief Networks Using Conditional Phase-Type Distributions," Faculty of Informatics, University of Ulster. 2001.