Statistical analysis of medical data with IBM SPSS Modeler

Cancer diagnosis support system

Alex Torquato Souza Carneiro
Marcelo Franceschi de Bianchi

October 22, 2012

IBM SPSS provides the main algorithms to recognize patterns that are identified in scientific literature about statistical data analysis, such as artificial neural networks, supporting vector machines, decision trees, and clustering algorithms. This article presents an application of SPSS Modeler software, as a medical diagnosis support system, helping identify both benign and malign tumors. Testing procedures are based on a set of reliable scientific data available in the http://archive.ics.uci.edu/ml/ repository (open to the general public) were employed to devise this article. This article was first published by developerWorks Brazil and is an example of developerWorks worldwide offerings.

Introduction

In the last few decades, computers and computing sciences evolve constantly. The constant change effects several industry segments, because it hinders the ability to analyze the data that is generated and stored in computers that are too often dependent of manual techniques. (See Related topics for more information on the evolution of technology.) The health industry, in particular, such as hospitals and clinics, generates large amounts of data without, however, appropriate resources to analyze such data in a fast and efficient manner. In this sense, there are systems capable of analyzing large quantities of data and providing system users with information such as data behavior trends or different data classes, helping the decision-making process, which is why they are known as decision support systems. The clinical decision support system (CDSS) is a specific diagnosis support system category for the medical industry. (See Related topics for more on the CDSS).

IBM offers, in its portfolio, the IBM SPSS software package (Statistical Package for Social Sciences), which was initially conceived to analyze social data, such as analyzing human behaviors or trends. However, its pattern recognition and statistical analysis algorithms allow the SPSS to be applied in any area or segment that require extracting relevant information from large quantities of data.
IBM SPSS Modeler as a supervised data classification system

Testing procedures conducted for this article were based on the SPSS Modeler software, currently under version 14.1, with the goal of implementing a decision support system to help medical diagnosis through supervised classification concepts and algorithms. According to TNM Classification of Malignant Tumors from the National Cancer Institute, Ministry of Health in Brazil (see Related topics), properly identifying tumors in patients as benign or malign is a decisive factor that is associated with the success of treatment procedures.

The SPSS Modeler software, presented in Figure 1, encompasses a graphic environment that creates flows composed of reading, preparation, analysis, modeling, and data writing elements (known as nodes).

Figure 1. SPSS screen executing one of the example files

Figure 1 presents a reading node, data preparation nodes, and data analysis nodes, in which each analysis node represents a single pattern recognition algorithm, and along with analysis nodes are nodes that are associated with obtained models, known as nuggets. Each nugget has an associated analysis node, with punctuated and non-directional connections; nuggets contain specifications of the associated algorithm that is applied to the data set, which varies between different data sets. In addition to algorithm specifications, nuggets also provide information about the performance of the algorithm to be applied over a certain data set, such as success rates in classification issues or error measurements in clustering or regression issues.
The algorithms that are employed in this article are MLP and SVM neural networks, which are used as supervised classification algorithms whose goal is to generalize each of the data classes. Figure 2 presents a spread chart for a set of two-dimensional data that shows three different classes, and the result of a supervised classification process obtained based on an algorithm similar to the ones indicated.

**Figure 2. Example of supervised classification result**

The lines that separate the data represent decision borders, which allows each data class to be identified as the region between borders. Each supervised classification algorithm presents a different result that is based on the position of the decision borders, which is why it is essential to assess more than one algorithm before you choose the most appropriate technique. (See *Pattern recognition: statistical and neural methods* under Related topics.)

**Introduction to supervised classification algorithms**

Scientific literature refers to several supervised classification algorithms, most of which are implemented in the IBM SPSS Modeler software. The following algorithms were selected for this procedure: MLP and SVM neural networks. The concepts of both algorithms that are employed in this article are detailed in this section.

**Multilayer perceptron neural networks (MLP)**

In 1999, Simon Haykin introduced artificial neural network algorithms that are based on the function principles of biological neural structures, as indicated in Figure 3. Neural computing
researches attempt to organize mathematical models similarly to the structures and organization of neurons in biological brains to achieve similar processing abilities, in addition to the inherent capacities of biological brains, such as learning based on examples, trial and error, and knowledge generalization, among many others. (See Neural networks: a comprehensive foundation under Related topics)

**Figure 3. Biological neural structure presenting the structural elements of neurons (dendrites, axon) and synapse**

![Biological neural structure](image)

Based on such principle of a model similar to animal brains, a neural network comprises elements responsible for processing activities, which are known as neurons that are interconnected through synapses and capable of processing information that is supplied to the network over time.

Artificial neurons can extract information from input data via the connections that are established between them, according to the layer layout in which neurons are organized, where the output of a neuron composes the input of neurons in the following layer.

**Figure 4** presents an artificial perceptron neuron model with N inputs (\{x_1, x_2, ..., x_N\}), in which each input \(x_i\) has an associated synapse \(w_i\), and an output \(y\). There is also an additional neuron parameter, named \(w_0\), known as bias that can be interpreted as the weight associated to an input \(x_0 = -1\). The output of the neuron \((y)\) is based on the product between weights, including the bias \((w_0)\). The neuron then formalizes such values through its transference function, also known as activation function\(^1\).
Figure 4. Artificial neuron model with inputs \{x_0, x_1, \ldots, x_N\}, weights \{w_0, w_1, \ldots, w_N\} and output \(y\)

The artificial neuron model that presented is feedforward, that is, connections are directed from inputs \{x_0, x_1, \ldots, x_N\} to output \(y\) of the neuron. Figure 5 presents the layout of a neural network, in which there are two neuron layers, one hidden and one output. Regarding the neural network shown in Figure 5, each neuron of the hidden layer is connected to each neuron in the output layer. Therefore, inputs of the output layer neurons correspond to the outputs of hidden layer neurons. The analyst that uses the artificial neural network algorithm must choose how many neurons to use in the hidden layer, considering the set of input data, since with a low number of neurons in the hidden layer it is impossible to generalize each class’ data. However, a high number of neurons in the hidden layer prompt the overfitting phenomenon, in which the neural network exclusively learns training data, and does not generalize learning for data classes.
Figure 5. Example of neural network with hidden layer

The neural network training process is conducted based on a backpropagation algorithm, in which a set of input data is determined for the desired neural network output. Such data is repeatedly presented to the neural network and the output provided is compared to the desired output, thus generating an error value. The error value is applied to adjust reverse neural network weights, from the output to inputs (backpropagated), reducing the error that is generated in each iteration and, therefore, promoting results more similar to the desired output. The most common method applied for data classification errors is to use an output neuron that corresponds to each of the input element classes, allowing the class that is identified by the neural network for each element to be established by the output layer neuron with the highest activation value.

In addition to the neural network algorithm, this article also encompasses the SVM algorithm.

Support Vector Machine (SVM)

V. Vapnik, 2000, introduces the SVM algorithm as a technique based on finding a hyperplain of separation between two data classes, which do not overlap, with the highest margin for each class's border elements possible. Figure 6 illustrates a case in which input data are two-dimensional and belong to two classes that are separated by a two-dimensional hyperplain.
(straight line), represented by the line between two sets of points. (See *The nature of statistical learning theory* in the Related topics section.)

**Figure 6. Example of two two-dimensional data classes that are separated by a straight line**

The elements of each class that are closer to the hyperplain that separates both classes are known as supporting vectors because they establish the path of the hyperplain, isolating each class with the largest margin possible.

For situations in which data cannot be separated by a hyperplain, as indicated in Figure 7, the SVM algorithm proposes the application of a non-linear transformation function over such data, separating them by a hyperplain in the input space that is obtained after the non-linear transformation.
The analyst that uses the SVM algorithm must choose which non-linear transformation function is the most appropriate for a certain data set, among the scientific literature proposals available in the IBM SPSS Modeler software: Gaussian function, polynomial function, linear function, or logistic sigmoid function. Figure 8 presents a possible classification result obtained based on a polynomial transformation function.
The following section presents the input data set for two algorithms that are used in this article.

**Database presentation**

The database that is used on this article is available to the general public at http://archive.ics.uci.edu/ml/, maintained by the Amherst Massachusetts University (http://rexa.info/) in collaboration with the National Science Foundation (US). Among the several databases available, the breast-cancer-wisconsin.data file presented in Figure 9 was chosen to create and analyze statistical models with the IBM SPSS Modeler software.
As indicated in Figure 9, the file that is used in this article is organized in 11 columns that are separated by commas, in which each line represents data that is collected from a patient. The first column of the file presents an identification code for each patient; the next 9 columns represent the attributes that are used to analyze each patient; and the last column presents a code to identify the type of cancer (2 for benign tumors and 4 for malign tumors). Overall, the file contains data of 699 patients that are diagnosed with cancerous tumors. The following attributes that are associated with the tumors, obtained from medical analysis, are analyzed for each patient:

- tumor thickness
- uniformity of cell sizes
- uniformity of cell forms
- marginal adhesion
- size of the simple epithelial cell
- Bare nuclei
- smooth chromatin
- nucleolus normality
• mitosis

All of these attributes are numerical variables with values between 1 and 10, representing information that is obtained via lab exams or medical assessments.

The next stage of this article is to prepare such data to be analyzed by SPSS Modeler algorithms. The following section establishes how data is prepared and applied, and how MLP and SVM algorithm statistical models are generated and analyzed by the SPSS Modeler.

Methodology

The first activity performed in this article is to adjust data to facilitate handling statistical models that are generated by SPSS. For such the breast-cancer-wisconsin.data file was converted into an electronic spreadsheet file \(^2\) (cancer.xls), as provided in Figure 10.

Figure 10. Spreadsheet with input data for statistical models

After the data is organized in a spreadsheet, it can be imported and manipulated by the elements of the SPSS Modeler software. Figure 11 presents the stream file that is created to import, prepare, and analyze the data.
Figure 11. Stream file that is developed to analyze the data, introducing the MLP and SVM algorithms

Figure 11 shows the file that uses data preparation elements to create statistical models:

- **Select**: Element that is used to eliminate samples that lack information
- **Derive (out)**: Creates a column that represents benign tumor = true and malign tumor = false
- **Type**: Element that is used to adjust data types that are based on analysis elements

In addition to data input elements, data preparation, and statistical algorithms, there are also analyses elements, which provide information such as success rates and confusion matrices for each algorithm.

The elements that contain MLP and SVM statistical algorithms were tested based on the following configurations:

- **SVM**
  - Default configuration for fixed parameters, **Figure 12**
  - Transformation functions used: radial base, sigmoid and polynomial function
- **MLP**
  - Default configuration for fixed parameters, **Figure 13**
  - Number of neurons in the hidden layer: 3, 6 and 10 (empirical values)
Figure 12. Default MLP algorithm configuration

(a)
The following section presents results that are obtained by the MLP and SVM algorithms for each configuration tested.

**Results**

Results were generated by using 683 records, removing 16 records that lacked data. The removal of such records occurs automatically with the Select element by filtering records with missing information.

Each of the MLP and SVM algorithms that were tested has specific aspects and approaches; the analyst must test different configurations to obtain the best results possible based on the adopted criteria.

This article presents success rates for each algorithm, corresponding to the percentage of tumors that were correctly identified within the data set that was used for the testing procedures; and the confusion matrix, which indicates the algorithm's success and error performance, presenting actual data classes as a matrix, previously known (malign or benign tumors), and how the algorithm rated testing data, accounting for successes and errors.

**Results that are obtained with the MLP algorithm**

The MLP algorithm has certain parameters, such as alpha factor and training interruption conditions, which, for the purposes of this article, are configured with default SPSS Modeler values. However, the number of neurons in the hidden layer was specified in 3, 6, and 10 (empirical values) to allow analyzing the results under different neural network architectures. The achieved results are shown in the following tables.
• 3 neurons in the hidden layer

<table>
<thead>
<tr>
<th></th>
<th>(algorithm output)</th>
<th>False</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>(malign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>234</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>(benign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>12</td>
<td>432</td>
<td></td>
</tr>
</tbody>
</table>

Success rate: 97.51%

• 6 neurons in the hidden layer

<table>
<thead>
<tr>
<th></th>
<th>(algorithm output)</th>
<th>False</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>(malign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>236</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(benign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>8</td>
<td>436</td>
<td></td>
</tr>
</tbody>
</table>

Success rate: 98.39%

• 10 neurons in the hidden layer

<table>
<thead>
<tr>
<th></th>
<th>(algorithm output)</th>
<th>False</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>(malign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>231</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>(benign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>14</td>
<td>430</td>
<td></td>
</tr>
</tbody>
</table>

Success rate: 96.78%

Results that are obtained with the SVM algorithm

The SVM algorithm varies according to the selected transformation function. In this article, the following transformation functions are presented: radial base function, sigmoid function, and polynomial function; each transformation function has its own set of characteristics, such as the sigma factor of a sigmoid function, or the order of a polynomial function.

• radial base function

<table>
<thead>
<tr>
<th></th>
<th>(algorithm output)</th>
<th>False</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>(malign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>231</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>(benign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>10</td>
<td>434</td>
<td></td>
</tr>
</tbody>
</table>

Success rate: 97.36%

• sigmoid function

<table>
<thead>
<tr>
<th></th>
<th>(algorithm output)</th>
<th>False</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>(malign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>114</td>
<td>125</td>
<td></td>
</tr>
<tr>
<td>(benign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>73</td>
<td>371</td>
<td></td>
</tr>
</tbody>
</table>

Success rate: 71.01%

• polynomial function

<table>
<thead>
<tr>
<th></th>
<th>(algorithm output)</th>
<th>False</th>
<th>True</th>
</tr>
</thead>
<tbody>
<tr>
<td>(malign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>False</td>
<td>233</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>(benign tumors)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Analysis of results achieved

The greatest success rate was obtained by using the SVM algorithm that is based on a polynomial transformation function, according to the presented specifications, in which the algorithm correctly identified 98.98% of cases. The smallest success rate was also obtained by using the SVM algorithm along with a sigmoid transformation function, in which the algorithm correctly identified 71.01% of cases.

Despite reaching a similar success rate, the MLP algorithm with 6 hidden layer neurons presents a lower success rate compared to the SVM algorithm based on a polynomial transformation function, correctly identifying 98.39% of the cases. However, the confusion matrices indicate that the SVM algorithm presents a larger error rate for malign tumors, identifying 5 cases of malign tumors as being benign tumors, while the MLP algorithm presents a larger error rate for benign tumors, identifying 8 cases of benign tumors as being malign tumors.

The decision of which algorithm is most appropriate, in this case, falls to a medical field expert, since only such qualified professional is able to assess which error would entail greater damages to patients, considering that a benign tumor erroneously identified as a malign tumor might cause psychological damages to patients, and most malign tumor treatments have severe side effects, while an erroneous malign tumor diagnosis, identified as a benign tumor, might delay treatment, causing the patient to lose valuable time in his/her recovery process. (See TNM Classification of Malignant Tumors under Related topics.)

Conclusion

This article summarizes an application of the SPSS Modeler software with the goal of assisting medical diagnosis of cancer diseases, identifying whether patients have malign or benign cancer. Considering the greatest average success rates, 98.39% for the MLP algorithm and 98.98% for the SVM algorithm, IBM SPSS Modeler is consolidated as an alternative base for a complete diagnosis support system, which can be implemented in hospitals, medical clinics, or any other healthcare institutions, thus reducing the probability of incorrect diagnosis.

1. Scientific literature traditionally adopts hyperbolic tangent or logistical sigmoid functions as a neuron's activation function.
2. The choice of an .xls (Excel) file pattern was based in the physician's knowledge of the Excel software and compatibility with the SPSS Modeler.
3. The polynomial transformation function used in this case is of order 3.
Related topics

- **Evolution of technology**: Learn more from this Wikipedia entry.
- **Clinical decision support system**: Learn more about the CDSS from Wikipedia.
- **Neural Networks: A Comprehensive Foundation** (2nd Edition), Simon Haykin: This book is one of the most comprehensive treatments of neural networks from an engineering perspective.
- **The nature of statistical learning theory** by Vladimir N. Vapnik: Learn more about the fundamental ideas that lie behind the statistical theory of learning and generalization from this book.
- **developerWorks on Twitter**: Join today to follow developerWorks tweets.
- **IBM product evaluation versions**: Get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

© Copyright IBM Corporation 2012

**Trademarks**
(www.ibm.com/developerworks/ibm/trademarks/)