Length of Stay Prediction and Analysis through a Growing Neural Gas Model

Luigi Lella[1], Antonio di Giorgio[2], Aldo Franco Dragoni[2]

[1]Azienda Sanitaria Unica Regionale delle Marche
Ancona, Italy
Luigi.Lella@sanita.marche.it

[2]Dipartimento di Ingegneria dell'Informazione (DII)
Università Politecnica delle Marche
Via Brecce Bianche 60131 Ancona, Italy
antonio.di.giorgio@email.it, a.f.dragoni@univpm.it

Abstract. Length of stay (LoS) prediction is considered an important research field in Healthcare Informatics as it can help to improve hospital bed and resource management. The health cost containment process carried out in Italian local healthcare systems makes this problem particularly challenging in healthcare services management.

In this work a novel unsupervised LoS prediction model is presented which performs better than other ones commonly used in this kind of problem. The developed model detects autonomously the subset of non-class attributes to be considered in these classification tasks, and the structure of the trained self-organizing network can be analysed in order to extract the main factors leading to the overcoming of regional LoS threshold.

Keywords: Business Intelligence in Health Care - LoS prediction - self organizing networks

1 Introduction

An accurate prediction of the length of stay (LoS) of recovered patients is considered a factor of strategic importance for the optimization of healthcare system resources [21,7]. This kind of information can be used to contain costs and eliminate waste by the reduction of hospital stays and readmission rates [4,15]. In Marche Region (Italy) the central maneuver of health cost containment led to an overall reorganization of healthcare system processes and to a heavy reduction in the number of hospital beds (and hospitals too). For this reason, the analysis of data on LoS becomes essential to effectively manage a hospital structure. Furthermore, the knowledge of the potential discharge date could improve also long term care activities or discharge activities planning [16]. This indeed can favor the continuity of care, a significant reduction of clinical risk together with the lowering of the related costs.

For all the above mentioned reasons it is considered extremely important to choose the right tools and methodologies to improve the prediction of LoS.

There has been a considerable effort in LoS prediction research to define the best solutions to cope with this problem. A first kind of methods is based on classic statistical algorithms such as *t-test, one-way ANalysis Of VAriance (ANOVA)* and *multifactor regression* [2].

A second kind of methods is based on AI techniques such as *decision trees* and *artificial neural networks* (ANN). ANN in particular have been successfully used in the context of postoperative phase of cardiac patients, or to identify patients at high risk of incur in prolonged intensive care [16]. Other ANN models have been used for LOS prediction in emergency rooms [20].

The best results have been obtained by the adoption of ensemble models and multilevel approaches making use of different clustering or categorization algorithms [9].

2 Methods

We are not interested here in the development of a new ensemble model. More exactly we are not interested in a mere predictive model. Our goal is not just to choose a good ANN model in hospital LoS prediction, but we are looking for a model or a methodology capable of explaining the acquired knowledge.

Most of learning techniques are oriented on a sort of structural representation of knowledge. This can be symbolic (e.g. acquired set of rules, decision trees etc.) or subsymbolic (e.g. associative networks, neural networks etc.). Subsymbolic models seem to reach the best results [17], but their structural representation need further analysis techniques in order to externalize the acquired knowledge.

Subsymbolic models can be further subdivided in *classification* learning algorithms (as feed forward networks and back-propagation models [9] [17]), *association* learning algorithms (as the Apriori algorithm [1]) and *clustering* learning algorithms (as the self-organizing networks [10] [18]).

In *classification* learning the system is trained to provide an output (a class) given a set of classified examples. For this reason, these algorithms are known in literature as "supervised". This kind of model is effective only if the correlation among the non-class attributes and all the possible classes are known beforehand. This is not the case of a dynamic model like the LoS prediction model. Our work is based on the assumption that almost every year scientific and technological discoveries lead to an improvement of care and a consequent reduction of hospital stays. Sometimes new therapies or diagnostic techniques can even lead to an increase of hospital stay. So it could be very hard and tricky trying to establish a set of classified examples of hospital stay, especially when precise guidelines or care pathways have not been defined.

In *association* learning there are not specified classes, the system just tries to find any interesting structure or correlation among data. The association rules can be used to predict every type of attribute, not just the class ones. Since we are interested in

LoS classes prediction, association learning models are not indicated for our problem. Association learning algorithms are probably more suited to implement expert systems capable to find correlation among clinical data and symptoms or to find complex symptomatologies.

Clustering algorithms, like association algorithms, are "unsupervised" ones, meaning that there is not a set of classified examples that can be used to train the system. But clustering algorithms try to define autonomously a set of classes. If we choose LoS as class attribute, the system can extrapolate different clusters related to the class attribute. In this way users are not committed to provide training sets of selected LoS examples, and the system could help the experimenters to find out the possible reasons leading to the overcoming of a given LoS threshold. In this phase the presence of human experts can be avoided making this solution more interesting and easy to implement.

Among the unsupervised algorithms, SOM have been effectively used in grouping data related to different lengths of treatments in emergency departments [22]. Nevertheless we think that SOM models are not particularly suited for LoS prediction.

In this kind of unsupervised learning task there is not a clear correlation among the class attribute and the other ones. In other terms the exact topology of the input space is unknown.

B. Fritzke in one of his works demonstrated that his Growing Neural Gas (GNG) model [5] is capable to identify exactly the local dimension of the input space. In other words on LoS prediction the GNG can find how many attributes in the defined input space are necessary to predict exactly the class attribute of hospital stay.

As it will be explained in the following section we have obtained a higher accurate prediction by the use of GNG in comparison with other algorithms which are commonly used in this kind of problem, in particular the J48 [19] algorithm which is one of the best algorithm based on the decision tree paradigm.

According to these assumptions we have choosen to use ZeroR, OneR, J48 and SOM as baseline approaches to compare with the GNG approach.

The first tested algorithm was the ZeroR [19]. ZeroR algorithm provides as a prediction always the majority class (in case of a nominal class attribute) or the average (in case of a numeric class attribute). This is considered the most simple predictive algorithm that is used to define a threshold for the accuracy. If other algorithms perform worse than this, probably they have been badly configured or more simply they are not suited for the class of problem to be dealt with.

The second tested algorithm was the OneR [19,8], which stands for "one rule". This method generates a decision tree with just one level. The training algorithm is quite simple. For each attribute a rule is created such that an attribute value is assigned to the most frequent class value correlated with it. For a numeric attribute a range of values is assigned with the most frequent class attribute, for a nominal attribute each value is assigned with the most frequent class attribute. Several rules are generated, but at last just one attribute is selected to make predictions, that is the one that produces the rules with the lowest error rate. Surprisingly this method has revealed a predictive power lower than few percentage points compared to other decision tree models.

The third tested algorithm was the J48 [19], which is the eighth version of C4.5 [14], corresponding to the last version distributed as free within this family of algorithms. J48 is based on the "divide and conquer" algorithm and the decision tree is recursively generated. Each time the node with the highest information quantity is selected and a branch for each of its possible values is created. This subdivides the data set in several subsets, one for every value of the attribute. This process is repeated for each branch but if all the instances belong to the same attribute class value the growth of the branch stops. The final tree can be downsized and simplified by pre-pruning or post-pruning techniques.

The fourth tested algorithm was the SOM [10]. A Self Organizing Map describes a mapping from a higher-dimensional input space to a lower-dimensional map space, typically a two-dimensional space like the one tested in this work. The training algorithm is designed to cause different parts of the network to respond similarly to certain input patterns. The training is based on competitive learning, meaning that for each input vector of the training set just a unit is selected as winner, that is the one whose weight vector is most similar to the input. The weights of the winner i and of the neurons i^* close to it in the SOM lattice are adjusted towards the input vector. The magnitude of the change decreases with time and with distance (within the lattice) from the winner according to the following update formula:

$$\Delta \boldsymbol{w}_i = \varepsilon(t)\Lambda(i, i^*, \sigma(t))(\boldsymbol{x} - \boldsymbol{w}_i)$$

Where ε(t) varies linearly with time from ε_{start} to ε_{end}, σ(t) varies linearly with time from σ_{start} to σ_{end}, and Λ is a Gaussian function centered on the winner unit i that includes all the neighbor i^* units.

The fifth tested algorithm was the GNG [5]. This algorithm is based on the Competitive Hebbian Learning (CHL) [11] and Neural Gas (NG) [12] algorithms. The former assumes an initial number of centers (units related to vectors having the same dimension of the input space) and successively inserts topological connections among them. For each input signal the two closest centers are connected by an edge. The other algorithm adapts the k nearest centers to each input which is being presented whereby k is decreasing from a large initial to a small final value.

In GNG algorithm the network topology of centers is generated incrementally by CHL and has a dimensionality which depends on the input data and may vary locally. The NG algorithm is used to move the nearest unit and its direct topological neighbors to the input signal by fractions ε_v and ε_n respectively of the total distance. For each input signal presented in the training phase a new connection is established between the first nearest unit and the second nearest unit and the local error variables of these two units are decreased multiplying them with a constant α. The age of all the edges connecting units are incremented by one and the edges with an age larger than a given threshold (α_{max}) are removed as well as isolated nodes. Finally all the local error variables are decreased multiplying them with a constant δ. If the number of the presented input signals is a multiple of a parameter λ a new unit is inserted and connected to the two units characterized by the highest local error variable (computed as the squared distance between the input signal and the corresponding center).

3 Data-set Preprocessing Techniques

We have considered as input data-set the hospital discharge summary forms regularly provided by our structures. These data have been provided by physicians through their electronic health records. Within these forms we were interested more in a subset of attributes which are the ones being filled at the admission of the patients. In particular we considered the following set of non-class attributes: recovery regimen, admission discipline, admission division, provenance, recovery type, trauma, hospital day care reason, hospital day care recovery type, main diagnosis, main intervention, complications, sex, age, marital status, qualification. We have chosen the hospital stay codified in a discretized form as class attribute.

The recovery regimen can take two values which stand for day hospital and ordinary recovery. For the admission discipline and the admission division there are 99 allowable values. There are only 9 values expected for the provenance: recovery without general practitioner suggestion, recovery with general practitioner suggestion, recovery programmed, transfer from a public structure, transfer from an accredited private structure, transfer from a not accredited private structure, transfer from another department or recovery regimen within the same institute, emergency medical service and other provenances. The recovery type can take 6 different values: recovery programmed, urgent hospitalization, mandatory medical treatment, recovery programmed with pre-hospitalization, voluntary hospitalization for medical treatment. The last value is used for not ordinary recoveries and for newborns. Trauma attribute codifies accidents, injuries and poisonings through 9 possible values: workplace accident, home accident, road accident, violence of others, self-harm or suicide attempt, animal or insect bite, sports accident, other type of accident or poisoning. This field is filled just in case of ordinary recovery. The hospital day care reason can be one of the following: day hospital, day surgery, day therapy, day rehabilitation while the hospital day care recovery type is codified in 3 values: not specified, first cycle for the specified diagnosis, following cycles for the specified diagnosis. The main diagnosis follows the international ICD9-CM coding system. Also the main intervention is based on the ICD9-CM system, but it considers just the first four digits of the code. Complications can take three values: without complications, not specified complications, with complications. Eight different age classes are expected: 0 years old, 1-4, 5-14, 15-44 male, 15-44 female, 45-64, 65-74, over 74. Six different marital status have been considered: celibate or unmarried, married, single separated, divorced, widower or widow, not specified. Six different qualifications are provided: no qualifications, elementary school license, middle or vocational school license, degree of professional qualification, baccalaureate, bachelor's degree.

At last the class attribute is codified in five different classes: one day hospital stay, two day hospital stay, three days hospital stay, below regional threshold stay, over regional threshold stay. The actual regional threshold for the hospital stay has been fixed to 5 days.

Weka 3.6.11 platform [19] has been used to launch Zero-R, One-R and J48 algorithms which need a conversion of all the discretized values in a nominal form by the use of "NumericToNominal" filter.

We have made the assumption that technologies and processes of care have remained unchanged in 2013, and we have processed all the hospital discharge summary forms in the year.

The data-set consisted of 274962 instances of hospital stay. In order to speed up the training phase of the chosen model we selected a significant sample of instances by the use of Weka "Re-sample" filter. As represented in figure 1, the "Re-sample" filter returned 1374 instances (corresponding to the 0.5% of the overall data-set) with the exact distribution of the original data-set.

To improve the learning process of the chosen self-organizing networks (SOM and GNG) we adopted the methodology suggested by Kohonen [10]. The representation input vector $\mathbf{x}$ was formed as a concatenation of a symbol part representing the hospital stay of the instance and a context part composed by the other attributes. The symbol part $\mathbf{x_s}$ and the context part $\mathbf{x_c=[x_{c1},\ldots,x_{c15}]}$ formed a vectorial sum of two orthogonal components such that the norm of the second part predominated over the norm of the former:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_s \\ \boldsymbol{x}_{c1} \\ \cdots \\ \boldsymbol{x}_{c15} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_s \\ 0 \\ \cdots \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \boldsymbol{x}_{c1} \\ \cdots \\ \boldsymbol{x}_{c15} \end{bmatrix}$$

In this way the symbols became encoded into a topological order (connection among neural units) reflecting their logical similarities.

Both the symbol part and the context part were encoded in a binary format. Discrete variables having relatively few values were encoded using a one-hot code system. The main diagnosis and the main intervention attribute values were transformed in binary (base-2) representations.

In the training phase both symbol and context part of input vectors were presented to the GNG model, while in the test phase just the context part was presented in order to predict the symbol part corresponding to the class attribute (LoS). Every time a test input vector was presented to the trained model, only a single unit of the self-organizing network “fired” (the most activated one). The predicted value, among all the possible ones of the class attribute, was the one closest to the symbol part of the center (weight vector) associated to the winning node.

4 Results

The re-sampled data-set was subdivided in a 66% (n=907 cases) part used as training set where the input vectors where used for SOM and GNG models with both the symbol part and the context part and a 34% (n=467 cases) part used as test set to test the predictive accuracy of the model.

The first three algorithms have been tested with the Weka default parameters and a 10-fold cross validation.

The output of ZeroR, OneR, J48 algorithms provided by Weka Explorer are represented in figures 1,2,3. Unexpectedly OneR performed better than the other two.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         574               41.7758 %
Incorrectly Classified Instances       800               58.2242 %
Kappa statistic                          0
Mean absolute error                      0.2881
Root mean squared error                  0.3795
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances             1374

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0         0         0         0          0.495    1
                 0         0         0         0         0          0.494    2
                 0         0         0         0         0          0.492    3
                 1         1         0.418     1         0.589      0.498    4
                 0         0         0         0         0          0.491    5
Weighted Avg.    0.418     0.418     0.175     0.418     0.246      0.495

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
   0   0   0 363   0 |   a = 1
   0   0   0 182   0 |   b = 2
   0   0   0 157   0 |   c = 3
   0   0   0 574   0 |   d = 4
   0   0   0  98   0 |   e = 5
```

Fig. 1. ZeroR prediction accuracy

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         887               64.556  %
Incorrectly Classified Instances       487               35.444  %
Kappa statistic                          0.5134
Mean absolute error                      0.1418
Root mean squared error                  0.3765
Relative absolute error                 49.2059 %
Root relative squared error             99.2245 %
Total Number of Instances             1374

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.835     0.236     0.559     0.835     0.67       0.799    1
                 0.374     0.06      0.489     0.374     0.424      0.657    2
                 0.561     0.053     0.575     0.561     0.568      0.754    3
                 0.643     0.1       0.822     0.643     0.721      0.771    4
                 0.602     0.025     0.648     0.602     0.624      0.788    5
Weighted Avg.    0.646     0.12      0.668     0.646     0.644      0.763

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 303  28   8  20   4 |   a = 1
  58  68  20  29   7 |   b = 2
  34  11  88  15   9 |   c = 3
 133  29  31 369  12 |   d = 4
  14   3   6  16  59 |   e = 5
```

Fig. 2. OneR prediction accuracy

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         838               60.9898 %
Incorrectly Classified Instances       536               39.0102 %
Kappa statistic                          0.3992
Mean absolute error                      0.206
Root mean squared error                  0.3304
Relative absolute error                 71.5034 %
Root relative squared error             87.0662 %
Total Number of Instances             1374

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.628     0.056      0.8       0.628     0.704      0.826    1
                 0.137     0.024      0.463     0.137     0.212      0.666    2
                 0.185     0.021      0.537     0.185     0.275      0.716    3
                 0.918     0.507      0.565     0.918     0.699      0.763    4
                 0.296     0.015      0.604     0.296     0.397      0.703    5
Weighted Avg.    0.61      0.234      0.613     0.61      0.566      0.757

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 228   4   5 120   6 |   a = 1
  22  25   7 121   7 |   b = 2
  10   9  29 108   1 |   c = 3
  17  15  10 527   5 |   d = 4
   8   1   3  57  29 |   e = 5
```

Fig. 3. J48 Prediction accuracy

For the SOM and GNG models we have developed two Java implementations of the algorithms. The re-sampled data-set was preprocessed as described in section 2 obtaining two sets of 123-bit vectors for the training set and the test set.

A 12x12 SOM was trained for 500 epochs with the following parameters: σ_{start} = 1, σ_{end} =0.1, ε_{start} =0.5, ε_{end} =0.005. In the test phase we obtained an accuracy of 87,59%.

Finally the GNG model was tested with the following parameters: λ=100, ε_v=0.2, ε_n=0.006, α=0.5, α_{max}=50, δ=0.995. The training continued until the main square error (that is the main of the local square error related to each unit, also called expected distortion error) dropped below the threshold of E=1 (corresponding to 207 epochs e.g. presentations of the training set).

We have reached an expected distortion error of 0.99 in the training phase with a network constituted by 950 units. In the test phase we obtained an accuracy of 96.36% which is considerably higher than the 64.56% accuracy of the OneR algorithm and the 87.59% of the SOM algorithm.

5 Discussion

The obtained results are indeed valuable for our local healthcare system allowing a good management of hospital beds. But we are interested in the extraction of the knowledge used by the model to predict so accurately the LoS.

Given the peculiar nature of GNG training algorithm we tried to use a clustering algorithm particularly suited to community-structured networks, that is networks where nodes are joined together in tightly knit groups connected by few edges [6]. We have used the JUNG API [13] for this kind of elaboration which was performed on a sub-net of the trained GNG network constituted by those units having the context part closer with the code of the over regional threshold stay. In other words we have selected the part of the trained network tied to the main criticality regarding the management of hospital beds.

For each cluster we extracted a set of attribute values considering the closest ones to the symbolic part of the center (or weight vector) of the nodes belonging to the cluster.

We have subsequently tagged the clusters by the use of the classic TF-IDF algorithm [3], considering all the extracted attributes.

The algorithm of Girvan and Norman has found eight main clusters and the TF-IDF algorithm assigned them seven tags which are related to the cases of hospitalization under general practitioner’s suggestion, suspicion of morbid condition in children, long stay hospitalization, obstetrics traumas, active muscoloskeletal exercises, children's cancer and other not well defined causes.

The elaborated criticalities have been validated by a group of human experts belonging to the management area of our organization. The first one is particularly interesting for the dimension of the cluster. The cases of hospitalization under general practitioner’s suggestion could represent a widespread phenomenon of defensive medicine, where general practitioners prescribe unnecessary and inappropriate visits to their patients.

This is only an attempt to extract valuable knowledge that surely require further research and a stricter scientific evidence. But the intent here is just to demonstrate how valuable knowledge could be extracted after the training phase with input data constituted by a symbol and a context part. Our final objective is to find a solution capable to give to our management sound and strong hints on healthcare system criticalities.

6 Conclusions

The processes of data mining and knowledge discovery don’t follow precise rules. There is not a model or a methodology capable to produce valuable results in every context of use. In the case of LoS prediction we have chosen a model which performs the so called “dimensionality reduction”. In other words it can find a low-dimensional space containing most of all input data.

This choice was driven by the assumption that there is not a clear correlation among clinical or anagraphic data and the LoS. The extraction of a significant set of examples associating patterns of non-class attributes to the LoS class sometimes can be a very problematic task to be performed, especially in all those cases where there is a lack of guidelines and clinical pathways, or where the innovation in technologies or clinical practice leads to an ever-changing correlation between clinical data and LoS.

In these cases the model has to self-organize his structure in an unsupervised manner in order to classify training data to the best possible. Growing Neural Gas has indeed the potentialities to adapt effectively to the input space, but it has to be correctly trained by the use of preprocessing techniques. Binary data in general are better assimilated by self-organizing networks, so we turned to the use of one-hot codes for nominal attributes with a limited set of values and to the use of binary (base-2) conversion in case of nominal attributes with a wide set of possible values.

Furthermore, we composed the input vector x as a concatenation of a symbol part representing the hospital stay of the instance and a context part composed by the other attributes taken from the hospital discharge summary forms regularly provided by our structures.

In this way, as suggested by Kohonen, symbols became encoded into a topological order (connections among neural units) reflecting their logical similarities.

The trained GNG performed better than other models (ZeroR, OneR, J48, SOM) reaching a prediction accuracy of 96.36%. This result proved the correctness of the choice of GNG model in LoS prediction tasks.

Finally we tried to extract the knowledge used by the model to predict hospital stays. As underlined before, symbols are encoded into a topological structure, meaning that the corresponding units (i.e. the units which are activated at their presentation) are connected to the units corresponding to other factors causing to the same LoS. The training algorithm itself is designed in a way that leads to the emergency of a community-structure. This consideration suggested us the opportunity to use a clustering algorithm suited for this kind of topological structures. Afterward by the use of the classic TF-IDF algorithm the identified clusters were tagged in order to extract the main factors (described by non-class attribute values) causing the overcoming of the regional LoS threshold.

Further experimentation is needed, but the first obtained results seem promising due to the fact that significant and verified knowledge has been extracted by the system.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. Proc. Of the 20th VLDB Conference, Santiago, Chile (1994).
2. Arab, M., Zarei, A., Rahimi, A., Rezaiean, F., Akbari, F.: Analysis of factors affecting length of stay in public hospitals in Lorestan Province, Iran. Hakim Res, vol. 12(4), pp.27-32 (2010).
3. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA (1999).
4. Chang, K.C., Tseng, M.C., Weng, H.H., Lin, Y.H., Liou, C.W., Tan, T.Y.: Prediction of length of stay of first-ever ischemic stroke. Stroke, vol. 33(11), pp.2670-4 (2002).
5. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. Part of: Advances in Neural Information Processing Systems 7, NIPS (1994).
6. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks, PNAS, vol.99(12), pp.7821-7826 (2002).
7. Gomez., V., Abasolo, J.E.: Using data mining to describe long hospital stays. Paradigma, vol. 3(1), pp.1-10 (2009).

8. Holte R.C.: Very simple classification rules perform well on most commonly used datasets. Machine Learning (1993)
9. Jiang, X., Qu, X., Davis, L.: Using data mining to analyze patient discharge data for an urban hospital. In: Proceedings of the 2010 International Conference on Data Mining, 2010 Jul 12-15; Las Vegas, NV., pp. 139-44 (2010).
10. Kohonen, T.: The Self Organizing Map. Proc. Of the IEEE, vol.78(9), (1999)
11. Martinetz, T.M.: Competitive Hebbian learning rule forms perfectly topology preserving maps. In ICANN'93: International Conference on Artificial Neural Networks, pp. 427-434. Amsterdam. Springer (1993)
12. Martinetz, T.M., Schulten, K.J.: A neural gas network learns topologies. In Kohonen, T., Kakisara, K., Simula O., and Kangas, J. Editors, Artificial Neural Networks, pp. 397-402. North-Holland. Amsterdam (1991)
13. O'Madadhain J., Fisher D., Smyth P., White S., Yan-Biao B.: Analysis and Visualization of Network Data using JUNG. J. of Stat. Software, pp.1-25 (2005)
14. Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.
15. Robinson, G.H., Davis, L.E., Leifer, R.P.: Prediction of hospital length of stay. Health Serv Res vol.1(3), pp.287-300 (1966).
16. Rowan, M., Ryan, T., Hegarty, F., O'Hare, N.: The use of artificial neural networks to stratify the length of stay of cardiac patients based on preoperative and initial postoperative factors. Artif Intell Med, vol. 40(3), pp.211-21 (2007).
17. Tu, J.V., Guerriere, M.R.: Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery. Proc Annu SympComput Appl Med Care, pp. 666-72 (1992).
18. Van Hulle, M.M.: Self Organizing Maps. Handbook of Natural Computing, pp. 585-622 (2012)
19. Witten, I.H., Frank, E., Hall, M.A.: Data Mining Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publishers (2011).
20. Wrenn, J., Jones, I., Lanaghan, K., Congdon, C.B., Aronsky, D.: Estimating patient's length of stay in the Emergency Department with an artificial neural network. AMIA Annu Symp Proc pp. 2005-1155 (2005).
21. Wright, S.P., Verouhis, D., Gamble, G., Swedberg, K., Sharpe, N., Doughty, R.N.: Factors influencing the length of hospital stay of patients with heart failure. Eur. J Heart Fail, vol. 5(2), pp. 201-9 (2003).
22. Xu, M., Wong, T.C., Chin, K.S.: A medical procedure-based patient grouping method for an emergency department. Applied Soft Computing, vol.14, pp.31-37 (2014)