# Length of Hospital Stay Prediction through Unorganised Turing Machines

Luigi Lella[1] and Ignazio Licata[2]
[1]Azienda Sanitaria Unica Regionale delle Marche, Ancona, Marche, Italy
[2]Institute for Scientific Methodology, Bagheria, Sicily, Italy

Keywords: Data Mining, Pattern Recognition and Machine Learning, Healthcare Management Systems.

Abstract: Length of hospital stay (LoS) prediction is one of the most important goals in Health Informatics, due to the fact that through this it is possible to optimize the management of health structure resources. In Italian local healthcare systems we are experimenting an health cost containment process and the minimization of care costs is considered an important objective to be achieved. For this reason we have tested several datamining models trained with hospital discharge data, capable to make accurate LoS predictions. In another work we have reached encouraging results by the use of unsupervised models which detect autonomously the subset of non-class attributes to be considered in these classification tasks. Here we are interested in studying also another intelligent data analysis model, the Turing unorganised A-type machine, that is capable to represent the acquired knowledge in a logic formalism. In other terms this solution can explain its predictions by the use of a set of self-acquired knowledge base rules.

## 1 INTRODUCTION

Length of hospital stay (LoS) prediction is considered an important strategic objective for the optimization of healthcare system resources (Wright et al., 2003, Gomez and Abasolo, 2009). As a matter of fact this kind of knowledge can lead to costs containment by the reduction of hospital stays and readmission rates (Chang et al., 2002, Robinson et al., 1966). This is considered a factor of vital importance in Italian States like Marche Region where the central maneuver of health costs containment has led to the overall reorganization of healthcare system processes and to a consistent reduction of hospital structures and beds. But this kind of prediction can have also important clinical outcomes, not just economic results. It has been proved that the knowledge of the potential discharge date can improve also long term care activities or discharge activities planning (Rowan et al., 2007).

Several solutions have been adopted to cope with LoS prediction. A first group is based on statistical algorithms such as t-test, one-way ANOVA and multifactor regression (Arab et al., 2010).

A second kind of methods is based on IA algorithms such as decision trees and artificial neural networks (ANN). ANN have produced important results in the context of postoperative phase of cardiac patients (Rowan et al., 2007) or in emergency rooms (Wrenn et al., 2005).

Indeed the best results have been achieved by the adoption of ensemble models (Jiang et al., 2010).

Learning techniques in general are based on a structural knowledge representation, both symbolic and subsymbolic. Subsymbolic models reach the best results in LoS prediction (Tu and Guerriere, 1992). These models can be further subdivided in classification algorithms (Jiang et al., 2010, Tu and Guerriere, 1992), association algorithms (Agrawal and Srikant, 1994), clustering algorithms (Kohonen, 1999, Van Hulle, 2012, Licata and Lella, 2007).

In classification learning a system is trained with a set of samples to provide a class output to new presented inputs. Unfortunately this approach is effective only when the correlation among the class and non-class attributes is clearly known beforehand.

In LoS context this prerequisite cannot be guaranteed. Sometimes the adoption of new therapies and diagnostic techniques can result in an increase of hospital stay. For this reason could be very difficult to determine beforehand a classified



Lella, L. and Licata, I.
Length of Hospital Stay Prediction through Unorganised Turing Machines.
DOI: 10.5220/0006577804020407
In *Proceedings of the 11th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2018) - Volume 5: HEALTHINF*, pages 402-407
ISBN: 978-989-758-281-3

set of samples, especially when there is a lack of guidelines or clinical pathways.

In association learning classes are not defined at all. The system just tries to detect interesting correlations among attributes. But these kind of systems don't cope very well with LoS classes prediction.

Finally clustering algorithms are "unsupervised", in other words there is not a set of classified examples which can be used in the training phase of the system. Just selecting the class attribute (i.e. the LoS class), the system is simply capable to extrapolate different clusters characterized by certain LoS values. In this way it can be easily argued that human expert knowledge is not needed.

SOM models are the clustering algorithms (Kohonen, 1999) which have been used in LoS prediction (Gorunescu et al., 2010), but in another work (Lella and Licata, 2017) we have successfully deployed an unsupervised algorithm which can operate in contexts, like the LoS one, where there is not a strong correlation among the class attribute and the other ones. The Growing Neural Gas (GNG) model by B. Fritzke (1994) that we have used, is able to detect the exact number of needed attributes to predict the class of hospital stay. We have achieved interesting prediction accuracy levels, but this subsymbolic model was not able to explain the result using a logic formalism.

In this preliminary work we are studying another unsupervised clustering algorithm which is based on the Turing A-type unorganised machine (Turing, 1948). The Turing's unorganised machine is generated "in a unsystematic and random way" from a set of two-input NAND gates. Turing chose a NAND gate because every other logical operations can be accomplished by a set of NAND units. A Turing A-type unorganised machine can be considered "a kind of Boolean neural network without a layered structure, due to the fact that recurrent connections are allowed with no constraints" (Teuscher and Sanchez, 2000). We used a genetic algorithm (GA) (Mitchell, 1996) to determine the best A-type network configuration. GAs are used to find high quality solutions in optimization and search problems by relying on bio-inspired operators of natural selection like mutation, crossover and selection.

After the evolution, i.e. the training phase, the best A-type network configuration is able to make LoS predictions, providing an explanation of the results through a logic formalism.

## 2 DATASET PREPROCESSING

We have processed the hospital discharge summary forms provided by our health structures. In particular we considered just a part of this dataset, which were the attributes being filled at the admission of the patients. The set of non-class attributes was: 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. The hospital stay period was codified in a discretized form as class attribute: one day hospital stay, two day hospital stay, three days hospital stay, below regional threshold stay, over regional threshold stay (5 days).

Weka platform (Witten et al., 2011) was 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 assumed that all the technologies and processes of care have been kept unchanged in 2013, and we processed all the hospital discharge summary forms of the year. The initial dataset, made up of 274962 instances of hospital stay, was reduced to 1374 instances in order to speed up the training phase of the tested models by the use of Weka "Resample" filter.

The chosen self-organizing networks (SOM, GNG and A-type network) were trained using the methodology suggested by Kohonen (1999). Each input vector was built by a concatenation of a context part representing the length of hospital stay of the instance and a symbol part consisting of the other attributes. The symbol part and the context part formed a vectorial sum of two orthogonal components such that the norm of the second part predominated over the norm of the former. Both the symbol part and the context part were encoded in a binary way. In particular discrete variables having relatively few values were encoded using a one-hot code system. For example the context part was codified by 5 bits, with just one of them capable to be in high (1) state. The main diagnosis and the main intervention attributes were instead coded in binary (base-2) representations. In this way each of the hospital discharge cases was codified by an array of 104 bits for the symbol part (the binary representation of the non-class attributes) and an array of 5 bits for the context part.

## 3 TRAINING AND TEST

The 66% of the resampled dataset was used as a training set, while the remaining 34% was used as test set. Both the symbol part and the context part of the training set was used for the self-organizing networks (SOM, GNG and A-type network), while just the context part of the test set was used to test the predictive accuracy of these models.
The first tested algorithm was the ZeroR (Witten et al., 2011) that is used in many cases as a benchmark. ZeroR predicts always the majority class in case of a nominal class attribute, and it is considered the simplest predictive algorithm.
The second tested algorithm was the OneR (Witten et al., 2011, Holte, 1993), standing for "one rule", that generates a decision tree defined by just one level. Each attribute value is assigned by a rule to the most frequent class attribute. At the end of the training phase just the rule with the lowest error rate is used to make the predictions in the test phase. This method has revealed a predictive power that is a little lower than the ones belonging to other decision tree models.
The third tested algorithm was the J48 (Witten et al., 2011), that is the eighth version of C4.5 (Quinlan, 1993) that is the last version distributed as free within this family of algorithms. J48 is based on a "divide and conquer" algorithm and its decision tree is recursively generated. At each training step the node having the highest information quantity is selected and a branch for each of its possible values is created. This process stops when all the instances belong to the same attribute class value.
The fourth tested algorithm was the SOM (Kohonen, 1999). A Self Organizing Map is a mapping of a higher-dimensional input space. A two-dimensional mapping was tested in this work. During the training phase different parts of the network can respond similarly to certain input patterns. The training is based on competitive learning, that is just one unit for each training input vector is selected as winner, the one whose weight vector is closer to the input.
The fifth tested model was the GNG (Fritzke, 1994) that is based on the Competitive Hebbian Learning (CHL) (Martinetz, 1993) and the Neural Gas (NG) (Martinetz and Shulten, 1991) algorithms. The former deploys an initial number of centers, i.e.the weight vectors of the units having the same dimension of the input space, and subsequently adds topological connections among the couples of closest centers to the presented inputs. The other algorithm adapts the k nearest centers, with k decreasing from a large initial value to a small final value. In this way the network topology is generated incrementally by CHL, with a locally varying dimensionality. The NG algorithm is used to move the centers of the nearest unit and its topological neighbours to the input signal by fractions $\varepsilon_v$ and $\varepsilon_n$ respectively of the total distance.
At last we chose an A-type model consisting of 24 NAND gates.
The first three algorithms were tested with Weka default parameters.
The output of ZeroR, OneR, J48 algorithms provided by Weka Explorer are represented in figures 1,2,3. As expected J48 seems to perform better than the other two.

```
Test mode:split 66.0% train, remainder test

=== Classifier model (full training set) ===

ZeroR predicts class value: 4

Time taken to build model: 0 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         198               42.3983 %
Incorrectly Classified Instances       269               57.6017 %
Kappa statistic                          0
Mean absolute error                      0.2856
Root mean squared error                  0.3785
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              467

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0         0          0         0         0          0.5      1
                 0         0          0         0         0          0.5      2
                 0         0          0         0         0          0.5      3
                 1         1          0.424     1         0.595      0.5      4
                 0         0          0         0         0          0.5      5
Weighted Avg.    0.424     0.424      0.18      0.424     0.252      0.5

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
   0   0   0 124   0 |   a = 1
   0   0   0  59   0 |   b = 2
   0   0   0  51   0 |   c = 3
   0   0   0 198   0 |   d = 4
   0   0   0  35   0 |   e = 5
```

Figure 1: ZeroR prediction accuracy.

```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         214               45.8244 %
Incorrectly Classified Instances       253               54.1756 %
Kappa statistic                          0.2479
Mean absolute error                      0.2167
Root mean squared error                  0.4655
Relative absolute error                 75.871  %
Root relative squared error            122.9832 %
Total Number of Instances              467

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.831     0.464      0.393     0.831     0.534      0.684    1
                 0.102     0.061      0.194     0.102     0.133      0.52     2
                 0.235     0.05       0.364     0.235     0.286      0.592    3
                 0.414     0.152      0.667     0.414     0.511      0.631    4
                 0.314     0.016      0.611     0.314     0.415      0.649    5
Weighted Avg.    0.458     0.202      0.497     0.458     0.437      0.628

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 103   8   1  11   1 |   a = 1
  34   6   8  10   1 |   b = 2
  22   4  12  11   2 |   c = 3
  91  12  10  82   3 |   d = 4
  12   1   2   9  11 |   e = 5
```

Figure 2: OneR prediction accuracy.

```
=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         266               56.9593 %
Incorrectly Classified Instances       201               43.0407 %
Kappa statistic                          0.2992
Mean absolute error                      0.2367
Root mean squared error                  0.3463
Relative absolute error                 82.8719 %
Root relative squared error             91.4866 %
Total Number of Instances              467

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.532     0.026      0.88      0.532     0.663      0.769    1
                 0         0          0         0         0          0.555    2
                 0         0          0         0         0          0.57     3
                 0.975     0.677      0.515     0.975     0.674      0.655    4
                 0.2       0.023      0.412     0.2       0.269      0.607    5
Weighted Avg.    0.57      0.296      0.483     0.57      0.482      0.66

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
  66   0   0  54   4 |   a = 1
   3   0   0  55   1 |   b = 2
   2   0   0  47   2 |   c = 3
   2   0   0 193   3 |   d = 4
   2   0   0  26   7 |   e = 5
```

Figure 3: J48 prediction accuracy.

SOM and GNG models have been developed by two Java implementations. The resampled dataset was pre-processed as explained in section 2, obtaining a 109-bits training set and a 109-bits test set. In the test set we replaced the 5 bits representing the context part by a zero padding.

A 12x12 SOM was trained for 500 epochs with the following parameters: $\sigma_{start}$ =1, $\sigma_{start}$ =0.1, $\varepsilon_{start}$ =0.5, $\varepsilon_{end}$ =0.005.

The GNG model was tested with the following parameters:$\lambda$=100, $\varepsilon_v$=0.2, $\varepsilon_n$=0.006, $\alpha$=0.5, $\alpha_{max}$= 50, $\delta$=0.995. The training was stopped when the main square error, i.e. the main of the local square error related to each unit (expected distortion error), dropped below the threshold of E=1.

The prediction accuracy of 96,3597% of the GNG model was considerably higher than the 87,5912% of the SOM algorithm and the 56,9593% of the J48 algorithm.

Finally the A-type unorganised Turing machine was tested by a Java implementation. The output of the network was provided by just 5 units, to give a one-hot answer. Each of the two inputs of the logic gates was represented by the output of another NAND gate or an input unit, that is one of the bits used to codify the non-class attributes of the hospital admission form. The overall network was made up of 128 units, that is the sum of the 104 input units (non-class attributes) and the 24 NAND gates.

Each of these units were codified by a 7-bit vector, which is able to represent 128 units. The resulting chromosome of the GA algorithm, modelling a certain network configuration, was made up of an array of 7x2x24=336 bits.

We evolved a population of 7000 chromosomes with a mutation rate of 0.015.

We employed a tournament selection method (Miller and Goldberg, 1995). Tournament selection involves running several "tournaments" among a few individuals, i.e. the chromosomes, chosen at random from the population. The winner of each tournament, that is the one with the best fitness rate, is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected. We chose a tournament size of 30 individuals. Crossover was implemented in the single crossover point version.

We also employed the elitism (Baluja and Caruana 1995), meaning that at the end of each generation the most performing individual was preserved by the effects of mutation and crossover operators.

The fitness of the network was defined as the number of correctly classified cases.

The evolution was stopped until we have reached a prediction accuracy similar to the J48 one.

The evolution of the chromosomes population just needed an average of 30 generations after that the system was also able to justify its answers.

We took into consideration just the inputs of the activated output unit to decode the answer.

Only few attributes were taken into consideration by the system to give an answer as represented in figure 4.

```
The expected hospital stay of patient #1239
is higher than the threshold value because :
 main diagnosis,

...

The expected period of stay of patient #1277
is 1 day because :
 not a patient in the first day hospital
 cycle for the specific diagnosis,
 patient in the next cycle of day hospital
 for the same diagnosis,

...

The expected period of stay of patient #1372
is between 4 days and the threshold value
because :
 ordinary hospitalization,
The expected period of stay of patient #1373
is 1 day because :
 patient in the first day hospital cycle for
the specific diagnosis,

Correctly Classified Instances: 56.20438 %
```

Figure 4: A-type prediction accuracy.

The right answers provided by the system were subsequently validated by a team of human experts chosen within the ASUR medical staff.

## 4 CONCLUSIONS

We actually know that there are not universal datamining techniques or methodologies to deal with every kind of problem or task. For length of hospital stay prediction we think that only unsupervised models can achieve the best results, because there is a lack precise guidelines and best practices capable to infer exactly the period of staying of patients, especially in those contexts characterized by rapid changes in technologies and organizational settings. In other words the knowledge of human experts in these cases cannot be exploited to define an accurate LoS prediction system.

For this reason in our research we have focused on unsupervised machine learning algorithms, in particular clustering algorithms and self-organizing networks.

We have obtained encouraging results through the use of subsymbolic models like the Growing Neural Gas by B. Fritzke in a previous research work, but now we are trying to develop more “intelligent” data analysers which are also capable to give a human-understandable explanation of their predictions. A response produced according to a logic formalism could indeed support decision makers in their health resources and services management activities.

That is why we have chosen an A-type unorganised Turing machine to process the admission forms of hospital patients. The structure itself of the model could be used like a kind of “dynamic” guideline to be taken into consideration by a group of human experts in order to optimally organize the healthcare activities performed on patients.

The knowledge acquired by an unorganised Turing machine through its pattern of NAND gates connections could also be used to produce an explanation of the reasons that led the system to its LoS predictions as we have demonstrated in this preliminary work.

We stopped the training just after having reached the prediction accuracy of the most performant decision tree algorithm represented by the J48. Also this model could be used to build a knowledge representation to approach the LoS prediction problem. But its tree-like structure probably is too simple to generate the complex set of rules to be used in these kind of decision processes.

We think that these first results can be further improved adopting another unorganised Turing machine model, that is the B-type one (Turing, 1948). Also a B-type may contain any number of NAND gates connected in any pattern. Turing just added the further condition that each unit-to-unit connection must pass through a modifier device. The modifier state can be set in “pass mode”, in which the output of a NAND gate passes through it unchanged, or in “interrupt mode”, in which the signal is always 1, no matter what the output of the NAND gate is (Copeland and Proudfoot, 1996). The presence of the modifiers can enable what Turing described as “appropriate interference, mimicking education”.

We are going to design and test a two-phase training, similar to the one proposed by Teuscher and Sanchez (2000), with a first “evolutive” phase where the best network configuration is selected, and a “learning” phase where the switches of NAND gates are enabled and properly configured to optimize the prediction accuracy rate.

## ACKNOWLEDGEMENTS

Special thanks go to Eng. Antonio Di Giorgio for his support in Weka datamining processes.

## REFERENCES

Agrawal R., Srikant R., 1994. *Fast Algorithms for Mining Association Rules*. Proc. Of the 20th VLDB Conference, Santiago, Chile, 1994.

Arab M., Zarei A., Rahimi A., Rezaiean F., Akbari F., 2010. *Analysis of factors affecting length of stay in public hospitals in Lorestan Province*, Iran, Hakim Res, Vol. 12, No.4, 2010, pp.27-32.

Baluja S., Caruana R., 1995. *Removing the genetics from the standard genetic algorithm* ICML.

Chang K.C., Tseng M.C., Weng H.H., Lin Y.H., Liou C.W., Tan T.Y., 2002. *Prediction of length of stay of first-ever ischemic stroke*, Stroke, Vol. 33, No.11, 2002 pp.2670-4.

Copeland B.J., Proudfoot D., 1996, *Alan Turing’s forgotten ideas in computer science*. Sci.Am. n.280, pp. 76-81.

Fritzke B., 1994. *A Growing Neural Gas Network Learns Topologies*. Part of: Advances in Neural Information Processing Systems 7, NIPS, 1994.

Gomez V., Abasolo J.E., 2009. *Using data mining to describe long hospital stays*, Paradigma, Vol. 3, No.1, 2009, pp.1-10.

Gorunescu F., El-Darzi E., Belciug S., Gorunescu M., 2010. *Patient grouping optimization using hybrid Self-Organizing Map and Gaussian Mixture Model for length of stay-based clustering system*, Intelligent Systems (IS), 2010 5th International Conference.

Holte R.C., 1993. *Very simple classification rules perform well on most commonly used datasets*, Machine Learning, 1993.

Jiang X., Qu X., Davis L., 2010. *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.

Kohonen T., 1999. *The Self Organizing Map*, Proc. Of the IEEE, vol.78, No.9, 1999.

Lella L., Licata I., 2017. *Prediction of Length of Hospital Stay using a Growing Neural Gas Model*, in Proceedings of the 8th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2017), pp. 175-178

Licata I., Lella L., 2007. *Evolutionary Neural Gas (ENG): A model of self-organizing network from input categorization*, EJTP, Vol.4, No.14, 2007.

Martinetz T.M., 1993. *Competitive Hebbian learning rule forms perfectly topology preserving maps*. In ICANN'93: International Conference on Artificial Neural Networks, pp. 427-434. Amsterdam. Springer, 1993.

Martinetz T.M., Schulten K.J., 1991. *A neural gas network learns topologies*. In T. Kohonen, K. Kakisara, O. Simula, and J. Kangas, Editors, Artificial Neural Networks, pp. 397-402. North-Holland. Amsterdam, 1991.

Miller B., Goldberg D., 1995. *Genetic Algorithms, Tournament Selection, and the Effects of Noise*, Complex Systems. 9, pp. 193–212.

Mitchell M., 1996. *An Introduction to Genetic Algorithms*, Cambridge, MA: MIT Press, 1996.

Quinlan J. R., 1993. *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

Robinson G.H., Davis L.E., Leifer R.P., 1966. *Prediction of hospital length of stay*, Health Serv Res Vol.1, No.3, 1966 pp.287-300.

Rowan M., Ryan T., Hegarty F., O'Hare N., 2007. *The use of artificial neural networks to stratify the length of stay of cardiac patients based on preoperative and initial postoperative factor*,. Artif Intell Med, Vol. 40, No.3, 2007 pp.211-21.

Teuscher C., Sanchez E., 2000. *A Revival of Turing's Forgotten Connectionist Ideas: Exploring Unorganized Machines*. In Proceedings of the 6th Neural Computation and Psychology Workshop, NCPW6, University of Lige, 2000.

Tu J.V., Guerriere M.R., 1992. *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.

Turing A., 1948. *Intelligent Machinery*, in Collected Works of A.M.Turing:Mechanical Intelligence. Edited by D.C.Ince.Elsevier Science Publishers, 1992.

Van Hulle M.M., 2012. *Self Organizing Maps*, Handbook of Natural Computing, pp. 585-622, 2012.

Witten I.H., Frank E., Hall M.A., 2011. *Data Mining Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, 2011.

Wrenn J., Jones I., Lanaghan K., Congdon C.B., Aronsky D., 2005. *Estimating patient's length of stay in the Emergency Department with an artificial neural network*, AMIA Annu Symp Proc pp. 2005-1155, 2005.

Wright S.P., Verouhis D., Gamble G., Swedberg K., Sharpe N., Doughty R.N., 2003. *Factors influencing the length of hospital stay of patients with heart failure*, Eur. J Heart Fail, Vol. 5, No.2, 2003, pp. 201-9.