Encapsulated models for reasoning and decision support

Ivana Čače [a] John-Jules Ch. Meyer [a, b] C.R.C. Pieterman [c] G.D. Valk [c]

[a] *Alan Turing Institute Almere, Louis Armstrongweg 84, 1311 RL Almere*
[b] *Intelligent Systems Group, ICS, Utrecht University, Princetonplein 5, 3584 CC Utrecht*
[c] *University Medical Center Utrecht, Endocrinology, Huispostnummer L.00.407, 3508 GA Utrecht*

Abstract

We show and discuss the benefits of incorporating a domain model as part of a classifier: it guards against over-fitting, helps ensure the classifier is appropriate for the problem at hand and adds to classification explainability.

1 Background

In this extended abstract [1] we outline how to incorporate knowledge from the medical domain in a classifier in a way that improves the transparency of classification, and makes the classification less dependent on both the particular data-set used for training and on peculiarities of the classification algorithm. Domain knowledge is incorporated as an isolated part of the classifier, based on the idea of *encapsulation* from the object oriented (OO) programming paradigm. Encapsulation is the practice of bundling methods and variables that 'belong together' in one object, for example because they pertain to a specific task or data. It makes software easier to develop, test and maintain.

We use a case study of Primary Hyper Parathyroidism (PHPT). We compare a decision tree incorporating a domain model with a tree built directly from the same data [1]. The classification problem here is to find those patients that should be scheduled for removing the parathyroid glands. A small data-set of 117 patients and a number of variables was available to us. Patient data included categorical variables like information on genetic mutations and family membership, which unfortunately consisted of too many values that could not be meaningfully clustered. The data also contained many missing values. As a result the original tree building algorithm used only two predictor variables: Calcium levels and parathyroid hormone (PTH) levels. We used the same two variables to allow a comparison between the two decision trees.

2 Approach

2.1 Domain Knowledge

When presented with a patient with suspected PHPT, the clinician will look for deviations in calcium and PTH levels, in addition to assessing other possible predictors. Two different medical models apply to calcium and PTH levels as predictors of PHPT: the normal ranges and the sigmoid relationship between calcium and PTH (see Fig. 1). PHPT patients show the same sigmoid curve, only displaced upwards and to the right.

[1] The full paper will be published in the proceedings of INFORMATIK 2013

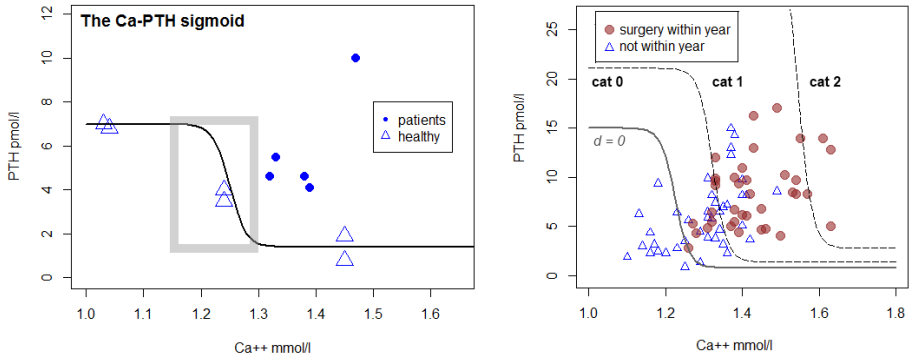

Figure 1: The calcium-PTH sigmoid graph, normal ranges in gray rectangle(l.), categories of d (r.).

2.2 Building a Decision Tree Incorporating the Domain Model

Based on domain knowledge we defined a sigmoid function which describes the relationship between calcium and PTH, and the severity of the PHPT expressed as d (deviation). The new measure d serves as input for the next classification step. From a computational point of view it reduces the data by one dimension, and it constrains the classifier to decision boundaries that correspond to the medical model. Semantically it constitutes a reasoning step: based on raw patient findings, a measure of deviation from normal findings is assigned to the case.

The parameters for the sigmoid function were fitted using the available data and a bootstrapping procedure; then d was discretized according to the Minimum Description Length principle to maximize its predictive value w.r.t. time to surgery. The decision tree was built using the Moku algorithm [1].

2.3 Results

Höhle [1] report an accuracy of 91% for the training cases and 95% for the test cases, while the tree that incorporates domain knowledge had a comparable 90% accuracy on both test and train cases. The original tree did appear to show some over-fitting: in some cases highly elevated PTH was contra-indicative of surgery.

2.4 Discussion and Conclusion

There is no single best classification technique that works best on all data-sets, considering domain knowledge ensures the classifier is appropriate for the problem. Incorporating domain knowledge also guards against over-fitting, this is particulary relevant if little train-data is available. We identified possible over-fitting in the decison tree that did not incorporate domain knowledge. In addition, an algorithm that incorporates concepts users are familiar with, makes the reasoning process more transparent which can aid the acceptance of the algorithm as a tool for decision support. The domain model is separate from the classification algorithm. This encapsulation should allow domain experts to work on the algorithm for calculating d independently from the rest of the classifier. The use of encapsulated models can be extended to incorporate more than one intermediate reasoning step.

References

[1] D. Höhle, C. R. C. Pieterman, G. D. Valk, A. R. Hermus, H. P. F. Koppeschaar, John-Jules Ch. Meyer, and R. P. J. de Lange. Classifying the decision to perform surgery in men1 cancer patients using decision trees. *Computer-Based Medical Systems, IEEE Symposium on*, pages 1–6, 2011.