

Elicitation of probabilistic logic rules from medical records: A preliminary study in learning policies for the management of critically ill children

Michael A. Skinner^{1,2}, Lakshmi Raman², Neel Shah², Abdelaziz Farhat², and
Sriraam Natarajan¹

¹ University of Texas at Dallas, Dallas TX

² University of Texas Southwestern Medical Center, Dallas TX

{michael.skinner, sriraam.natarajan}@utdallas.edu,
{lakshmi.raman, neel.shah, abdelaziz.farhat}@utsouthwestern.edu

Abstract. The increased use of electronic health records has made possible the automated extraction of medical policies from patient records to aid in the development of clinical decision support systems. We adapted a Probabilistic Logic framework to learn probabilistic rules from clinical hospital records for the management of physiologic parameters of children with severe cardiac or respiratory failure who were managed with extracorporeal membrane oxygenation. In this preliminary study, the results were promising. In particular, the algorithm returned logic rules for medical actions that are consistent with medical reasoning. To our knowledge, this is the first report of the use of probabilistic logic techniques to automatically extract a medical policy from medical records.

Keywords: Probabilistic Logic learning · medical policy.

1 Introduction

The ability to automatically learn physician actions from electronic health records (EHR) could contribute to improved health care in a number of ways. For example, we could automatically discover optimal policies³ for managing particular diseases. Moreover, an optimal policy, once discovered, could be compared to a patient’s actual clinical course; if there is a deviation, physicians could be provided with suggestions for care. Finally, the ability to extract medical policies from EHRs would enable predictions of patient prognosis and outcomes.

In this work-in-progress, we investigate use of an inductive probabilistic learner to elicit weighted (probabilistic) clauses mapping the values of a set of physiologic parameters to physician actions in critically ill patients with respiratory or cardiac failure. We extracted the information from clinical trajectories

³ We follow the conventional reinforcement learning definition of a policy as a mapping from states to actions

documented in the EHR. The goal of this work is not to propose a new probabilistic logic learning framework or to compare different frameworks but to explore the use of such frameworks in this challenging medical task.

The clinical challenge – discovering a medical policy: Unwanted variation in medical care, recognized for over forty years, remains a challenge to health care providers in nearly every specialty[17, 4, 6]. Differences in care are typically observed between geographic regions, and the particular practice in an area often correlates with available resources. For example in one study, investigators found a high correlation between the availability of cardiac catheterization within a locality and use of angioplasty for managing cardiac disease[2]. It is surprising that these challenges persist, even as there has been a multiplication of published expert guideline documents for many medical conditions whose recommendations are based on well-performed prospective clinical trials[16].

To decrease variability of care, and to converge medical management around policies conforming to expert guidelines, clinical decision support systems (CDS) have been devised to render advice to clinicians as they care for patients[13]. Such systems were initially very limited, highly dependent on manual curation, and their scope was limited to a very few medical conditions[8]. However, the increased use of EHRs has stimulated the development of automated CDS systems holding promise for providing advice to health care providers in real time.

Required in an automated CDS system is the ability to monitor some aspects of the patient’s clinical state, as well as the physician actions[8]. Moreover, the system must possess some notion of optimal care; when clinicians deviate from the preferred management, or if unexpected events occur warranting a change in care, alerts or reminders are provided from the system. Whereas early systems used hard-coded rules to encode clinicians’ knowledge about the optimal policy, there is growing interest in automatically extracting optimal care patterns by mining the EHR[12].

Reinforcement learning (RL) is the most commonly reported technique to extract clinical policies from medical records[5, 15]. Somewhat surprisingly, to our knowledge, other policy learning schemes such as imitation learning have not been reported in the medical realm. Most of the reported RL models use deep neural networks, requiring many patient records for training and a proposition-alization/embedding technique that could lead to loss of information. Moreover, these models may be difficult to interpret, complicating the identification of best medical actions for a given patient state.

A key issue when learning medical policies is that it is non-trivial to construct a vector-based representation for EHR information. There can be multiple measurements performed over varying time-scales, multiple treatments of different conditions at the same time-step and differing numbers of observations per subject. Thus, if standard machine learning methods are used to represent these complicated data, critical information may be lost.

A more natural representation that allows for modeling these data is first-order logic. Hence, motivated by the fact that physicians generally make treatment decisions based on physical, laboratory and radiologic findings in a sys-

tematic manner through a series of (often implicit) "if-then" decisions, we investigated the usefulness of learning medical polices as sets of probabilistic clauses learned in a probabilistic logic framework[3, 14].

2 Problem Description - ECMO patients

Extracorporeal membrane oxygenation (ECMO) is a method of supporting patients with severe respiratory or cardiac failure. The technique requires placement of large cannulas in the neck or in the heart, and externally circulating the patient's blood through a system that oxygenates the blood and removes carbon dioxide. Reserved for the most critically ill of patients, mortality can be very high and even among survivors there are frequent treatment complications[7].

This study used de-identified medical data abstracted from EHRs for 141 children treated at the Children's Medical Center of Dallas who survived their period of ECMO. The study was performed in accordance with an exemption granted by the University of Texas Southwestern Institutional Review Board (IRB). The time on ECMO ranged from 6 to 985 hours, averaging 174 hours. For each hour of ECMO bypass, and for from 1 to 24 hours prior to cannulation (15 hours, on average), 40 physiologic and laboratory parameters were recorded. Not every parameter was measured each hour; for example, those exclusively associated with ongoing bypass (such as pump flow) were only recorded while the child was actually undergoing ECMO support.

We chose seven physiologic parameters thought to be the most useful for managing the respiratory and hemodynamic status of patients. These are tabulated, with the units of measurement, in Table 1. Parameter values were each discretized into five bins; the demarcations were based on meaningful physiologic categories. Thus, for example, the range of Mean arterial pressure (MAP) values was 50,60,70,80,80.1 (mm Hg.). If the $MAP \leq 50$, then the bin was labeled 50, if $MAP > 80$, then the assigned bin was 80.1, and if $50 < MAP \leq 60$, the bin assignment was 60, and so forth. Of course, the bin values and/or units are different for each parameter.

Table 1. Study parameters.

Parameter	Units
Mean arterial pressure	mm Hg.
Heart rate	beats/min
Respiratory rate	breaths/min
pH	none
pO2	mm Hg.
Pressure volume sensor	cm H2O
Measured flow	ml/kg-min

3 Probabilistic logic models for learning medical policies in ECMO patients

Given: EHR data from a set of patients and a set of actions listed in Table 1.

To Do: Learn (parameterized) policies for specifying the appropriate medical actions to alter physiologic parameters.

In other words, our aim is to learn from the data when physicians should initiate therapy to alter the parameters listed in Table 1.

We are inspired by prior work on learning policies using Probabilistic logic models (PLMs) [10] where (parameterized weighted logical) clauses were learned from observed trajectories. Broadly known as "imitation learning", the key idea is to learn a distribution over actions such that the policies are as close to the observed user policy as possible. This particular setting is quite useful in cases where the reward function is difficult to specify in advance. Imitation learning algorithms directly optimize the learned policy from trajectories instead of the expected cumulative discounted reward (as in reinforcement learning); in many cases this is easier, since when we have observed trajectories, we can avoid exploration. We consider learning from observations and learn a PLM from data.

Our PLM learning method is based on learning a set of logical regression trees (TILDE) [1] in a stage-wise manner. This learning method uses an underlying Inductive Logic Programming (ILP) learner [9] to induce a set of logical clauses and then fits the weights (parameters) of these clauses. We employ the machinery of gradient-boosting [11] where differences between observed and predicted probabilities are computed as gradients for the training examples and TILDE trees are learned at each step to fit these gradients. For more details, we refer to our previous work [11].

Recall that an ILP algorithm accepts a set of facts, sets of positive and negative examples of the concepts to learn, returning logic programs defining the learned concepts. To learn the concepts listed in Table 2, we include as facts the values of each parameter for each subject for each hour; for instance, "map(subj1,100,70)" represents that subject1 at time step 100 hours had a mean arterial blood pressure between 60-70 mmHg. As noted earlier, not every parameter was measured each hour. The examples were derived from these facts. If on the consideration of two consecutive measurements, there was a significant change in the parameter value (defined as a change of at least two bins in the discretized values), then we generated a positive example. For example, if in addition to the fact

Table 2. Policy actions.

Action
Increase mean arterial pressure
Increase/decrease respiratory rate
Decrease heart rate
Increase/decrease pH
Increase/decrease pO2
Increase/decrease pressure volume sensor
Increase/decrease measured flow

listed above, there was "map(subj1,101,80.1)", indicating a significant increase in blood pressure after hour 100, the positive example "mapincr(subj1,100)" would be generated. Otherwise, we synthesized a (false) negative example.

4 Results

In this preliminary experiment, we set the parameters of the boosted learning algorithm so that each concept was approximated by a set of 20 relational logic trees. In these trees, each node consists of a logic clause whose possible truth values are represented by the edges. Leaves of the tree are labelled with the weight (and the value subjected to the *sigmoid* function in parentheses) corresponding to logic rules constructed by following from the root to the leaf. A representative probabilistic logic tree is presented in Figure 1.

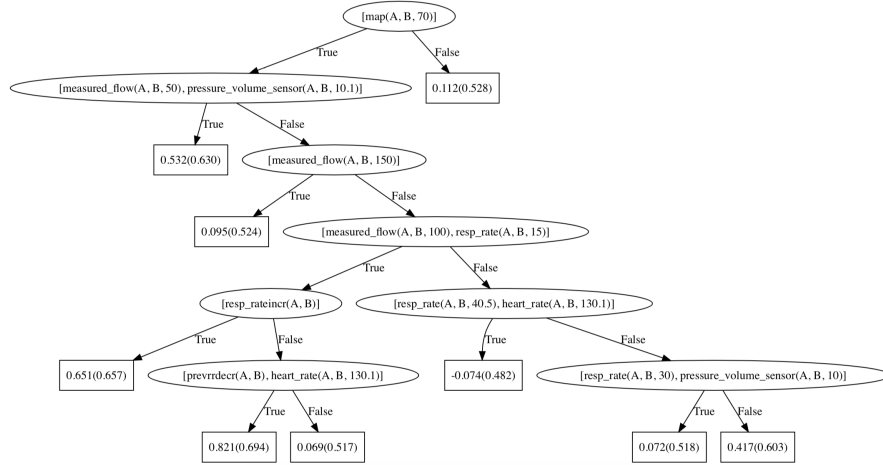


Fig. 1. Representative probabilistic logic tree for the action to increase mean arterial pressure (map_incr). The prevrrdecr(A,B) refers to a decrease in the respiratory rate in the previous hour, expressed by the logic rule $\exists C \mid B = C + 1 \wedge \text{resp_ratedecr}(A, C)$.

We extracted weighted probabilistic logic rules from this tree; the generated clauses are listed in Table 3.

Table 3. Weighted logic rules for mean arterial pressure increase generated from a representative boosted tree. Variable A represents the subject. Variables B and C represent the time.

Rule No.	Weight	Logic rule
1	0.112	$\text{mapincr}(A, B) \Leftarrow \neg \text{map}(A, B, 60 - 70)$
2	0.532	$\text{mapincr}(A, B) \Leftarrow \text{map}(A, B, 60 - 70) \wedge \text{measured_flow}(A, B, 20 - 50) \wedge \text{pressure_volume_sensor}(A, B, > 10)$
3	0.095	$\text{mapincr}(A, B) \Leftarrow \text{map}(A, B, 60 - 70) \wedge \neg[\text{measured_flow}(A, B, 20 - 50) \wedge \text{pressure_volume_sensor}(A, B, > 10)] \wedge \text{measured_flow}(A, B, 100 - 150)$
4	0.651	$\text{mapincr}(A, B) \Leftarrow \text{map}(A, B, 60 - 70) \wedge \neg[\text{measured_flow}(A, B, 20 - 50) \wedge \text{pressure_volume_sensor}(A, B, > 10)] \wedge \neg \text{measured_flow}(A, B, 100 - 150) \wedge \text{measured_flow}(A, B, 50 - 100) \wedge \text{resp_rate}(A, B, \leq 15) \wedge \text{resp_rateincr}(A, B)$
5	0.821	$\text{mapincr}(A, B) \Leftarrow \text{map}(A, B, 60 - 70) \wedge \neg[\text{measured_flow}(A, B, 20 - 50) \wedge \text{pressure_volume_sensor}(A, B, > 10)] \wedge \neg \text{measured_flow}(A, B, 100 - 150) \wedge \text{measured_flow}(A, B, 50 - 100) \wedge \text{resp_rate}(A, B, \leq 15) \wedge \neg \text{resp_rateincr}(A, B) \wedge \text{heart_rate}(A, B, > 130) \wedge [\exists C \mid B = C + 1 \wedge \text{resp_ratedecr}(A, C)]$
6	0.069	$\text{mapincr}(A, B) \Leftarrow \text{map}(A, B, 60 - 70) \wedge \neg[\text{measured_flow}(A, B, 20 - 50) \wedge \text{pressure_volume_sensor}(A, B, > 10)] \wedge \neg \text{measured_flow}(A, B, 100 - 150) \wedge \text{measured_flow}(A, B, 50 - 100) \wedge \text{resp_rate}(A, B, \leq 15) \wedge \neg \text{resp_rateincr}(A, B) \wedge \neg[\text{heart_rate}(A, B, > 130) \wedge [\exists C \mid B = C + 1 \wedge \text{resp_ratedecr}(A, C)]]$
7	-0.074	$\text{mapincr}(A, B) \Leftarrow \text{map}(A, B, 60 - 70) \wedge \neg[\text{measured_flow}(A, B, 20 - 50) \wedge \text{pressure_volume_sensor}(A, B, > 10)] \wedge \neg \text{measured_flow}(A, B, 100 - 150) \wedge \neg[\text{measured_flow}(A, B, 50 - 100) \wedge \text{resp_rate}(A, B, \leq 15)] \wedge \text{resp_rate}(A, B, > 40) \wedge \text{heart_rate}(A, B, > 130)$
8	0.072	$\text{mapincr}(A, B) \Leftarrow \text{map}(A, B, 60 - 70) \wedge \neg[\text{measured_flow}(A, B, 20 - 50) \wedge \text{pressure_volume_sensor}(A, B, > 10)] \wedge \neg \text{measured_flow}(A, B, 100 - 150) \wedge \neg[\text{measured_flow}(A, B, 50 - 100) \wedge \text{resp_rate}(A, B, \leq 15)] \wedge \neg[\text{resp_rate}(A, B, > 40) \wedge \text{heart_rate}(A, B, > 130)] \wedge \text{resp_rate}(A, B, 20 - 30) \wedge \text{pressure_volume_sensor}(A, B, 0 - 10)$
9	0.417	$\text{mapincr}(A, B) \Leftarrow \text{map}(A, B, 60 - 70) \wedge \neg[\text{measured_flow}(A, B, 20 - 50) \wedge \text{pressure_volume_sensor}(A, B, > 10)] \wedge \neg \text{measured_flow}(A, B, 100 - 150) \wedge \neg[\text{measured_flow}(A, B, 50 - 100) \wedge \text{resp_rate}(A, B, \leq 15)] \wedge \neg[\text{resp_rate}(A, B, > 40) \wedge \text{heart_rate}(A, B, > 130)] \wedge \neg[\text{resp_rate}(A, B, 20 - 30) \wedge \text{pressure_volume_sensor}(A, B, 0 - 10)]$

Owing to the fact that a probabilistic logical concept is represented by a sum of the 20 weighted trees, it is difficult to interpret the rules generated by our model. However, by comparing some of the weighted rules within a tree, we can elicit findings consistent with known clinical practice. When we look at Rule 1, we see that the weight of the action to increase the mean arterial blood pressure (map) is 0.112 when the map is not between 60-70 mm hg. (which is roughly the normal range). However, comparing to Rule 2, we see that the weight increases to 0.532 when the map is in the normal range, and if the pump flow is relatively low (20-50 ml/kg-min), and when the pump preload pressure is relatively high (> 10 cm H₂O). The increased weight on this clause, compared to Rule 1, suggests that in circumstances where the map is normal, but if the pump flow is low, physicians may elect to initiate treatment to raise the blood pressure. This is a reasonable treatment maneuver.

We see another example when comparing Rules 5 and 6. The difference between these rules is in the conjunction of the last two clauses:

`heart_rate(A, B, > 130) ∧ [∃C | B = C + 1 ∧ resp_ratedecr(A, C)].`

This clause is present in Rule 5 but negated in Rule 6. The markedly elevated heart_rate (> 130 beats/minute) and a recent decrease in respiratory rate are clinical signs of disease severity. The higher weight on Rule 5 (when these findings are present) indicates that the presence of these findings will result in a higher probability of the physician moving to increase the map, which is clinically very reasonable.

As we add clauses to the rules or negate them, moving down the tree, it is generally true that the changing weights make clinical sense. That is, a physician is able to explain why the rule was created. However, it also appears that some clauses seem peripheral to the task of deciding whether to increase the arterial pressure, and would not necessarily be used in the clinical setting. Without question, our automatic system is able to generate longer, more complicated probabilistic logic rules and use more (perhaps obscure, and perhaps important) clinical facts than could a human. Whether such rules will be clinically relevant and useful in a functioning clinical decision support system is a question requiring further research.

5 Discussion

We used a probabilistic logic framework to elicit policies for medical management of children undergoing ECMO. Our preliminary results provide some hope that the method can be used to provide interpretable strategies to physicians managing complicated patients that might be useful in automatic clinical decision support systems. To the best of our knowledge, this is the first study using EHR data to learn probabilistic rules governing the management of patients in the hospital. It can be observed from the rules presented that we include existential variables (observations recognized previously); simple encodings into a propositional framework will not suffice for such problems. Instead, a logical framework is necessary. Also, note the weights/regression values on the leaves.

This demonstrates the need for a “soft” framework and supports our choice of PLMs as a natural choice for such modeling tasks.

There are at least a few shortcomings to our study from a clinical perspective. First, it must be acknowledged that we did not have direct access to the physician actions, and rather derived them from the measured physiologic parameters. This complicates our analysis, owing to the fact that we are unable to distinguish when altered physiologic findings are related to medical care or to the course of the underlying disease. It is reasonable to surmise that when we have available the actual physician orders, we will have a cleaner, less noisy, set of data, perhaps allowing greater success in eliciting the medical policies. A second shortcoming is that in this study we selected only a small subset of the recorded parameters- ones thought to be most physiologically significant in the medical decision-making process. We might obtain better results if we broaden the set of parameters. Moreover, in our probabilistic boosted logic model experiment, we discovered policies encoded in multiple regression trees; without question, one could question whether the trees learned in the boosting algorithm are readily interpretable to physicians managing patients. This is a broader issue; if weighted logic models such as MLNs/PSL etc., are interpretable, then so are these boosted rules. However, we acknowledge that weighted logic may not be as interpretable to domain experts, and there is a need to explore models that are more explainable and interpretable.

Finally, our technique may provide new insight into which physician actions contribute to variation in clinical outcome for children undergoing ECMO support. For example, the most common risks of ECMO include bleeding (related to the necessary anticoagulation of the patient) and neurologic injury- either an intracranial hemorrhage or an ischemic event. What is not known is whether neurologic injury risk can be altered by different management schemes. We surmise that when we evaluate policies for patients partitioned by outcome class (that is, with or without neurologic event), aspects of the policies may be elicited that increase complication risk. We leave this to future work. But if such a finding were confirmed in a clinical study, ECMO outcomes could be improved.

References

1. Blockeel, H., De Raedt, L.: Top-down induction of first-order logical decision trees. *Artificial intelligence* **101**(1-2), 285–297 (1998)
2. Brownlee, S.: *Overtreated: Why too much medicine is making us sicker and poorer*. Bloomsbury Publishing USA (2010)
3. De Raedt, L., Kersting, K.: Probabilistic inductive logic programming. In: *Probabilistic Inductive Logic Programming*, pp. 1–27. Springer (2008)
4. Karimi, M., Sullivan, J.M., Lerer, T., Hronek, C.: National trends and variability in blood utilization in paediatric cardiac surgery. *Interactive cardiovascular and thoracic surgery* **24**(6), 938–943 (2017)
5. Komorowski, M., Celi, L.A., Badawi, O., Gordon, A.C., Faisal, A.A.: The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine* **24**(11), 1716 (2018)

6. Lilot, M., Ehrenfeld, J., Lee, C., Harrington, B., Cannesson, M., Rinehart, J.: Variability in practice and factors predictive of total crystalloid administration during abdominal surgery: retrospective two-centre analysis. *British journal of anaesthesia* **114**(5), 767–776 (2015)
7. Lin, J.C.: Extracorporeal membrane oxygenation for severe pediatric respiratory failure. *Respiratory care* **62**(6), 732–750 (2017)
8. Middleton, B., Sittig, D., Wright, A.: Clinical decision support: a 25 year retrospective and a 25 year vision. *Yearbook of medical informatics* **25**(S 01), S103–S116 (2016)
9. Muggleton, S.: Inductive logic programming. No. 38, Morgan Kaufmann (1992)
10. Natarajan, S., Joshi, S., Tadepalli, P., Kersting, K., Shavlik, J.: Imitation learning in relational domains: A functional-gradient boosting approach. In: *Twenty-Second International Joint Conference on Artificial Intelligence* (2011)
11. Natarajan, S., Kersting, K., Khot, T., Shavlik, J.: *Boosted statistical relational learners: From benchmarks to data-driven medicine*. Springer (2015)
12. Ohno-Machado, L.: Using health information technology for clinical decision support and predictive analytics. *Journal of the American Medical Informatics Association* **24**(1), 1–1 (2016)
13. Osheroff, J.A., Teich, J.M., Middleton, B., Steen, E.B., Wright, A., Detmer, D.E.: A roadmap for national action on clinical decision support. *Journal of the American medical informatics association* **14**(2), 141–145 (2007)
14. Raedt, L.D., Kersting, K., Natarajan, S., Poole, D.: *Statistical relational artificial intelligence: Logic, probability, and computation*. *Synthesis Lectures on Artificial Intelligence and Machine Learning* **10**(2), 1–189 (2016)
15. Raghu, A., Komorowski, M., Celi, L.A., Szolovits, P., Ghassemi, M.: Continuous state-space models for optimal sepsis treatment—a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422* (2017)
16. Weisz, G., Cambrosio, A., Keating, P., Knaapen, L., Schlich, T., Tournay, V.J.: The emergence of clinical practice guidelines. *The Milbank Quarterly* **85**(4), 691–727 (2007)
17. Westert, G.P., Groenewoud, S., Wennberg, J.E., Gerard, C., DaSilva, P., Atsma, F., Goodman, D.C.: Medical practice variation: public reporting a first necessary step to spark change. *International Journal for Quality in Health Care* **30**(9), 731–735 (2018)