Adaptive predictive-questionnaire

by approximate dynamic programming

Frédéric Logé [1,2] Erwan Le Pennec [1,3] Habiboulaye Amadou-Boubacar [2]

[1] CMAP, Polytechnique, Institut Polytechnique de Paris, Palaiseau, FR

[2] Air Liquide R&D, Jouy-en-Josas, FR

[3] XPop, Inria Saclay, FR

September 9th, 2020
UCAI'20: Workshop on User-Centered Artificial Intelligence

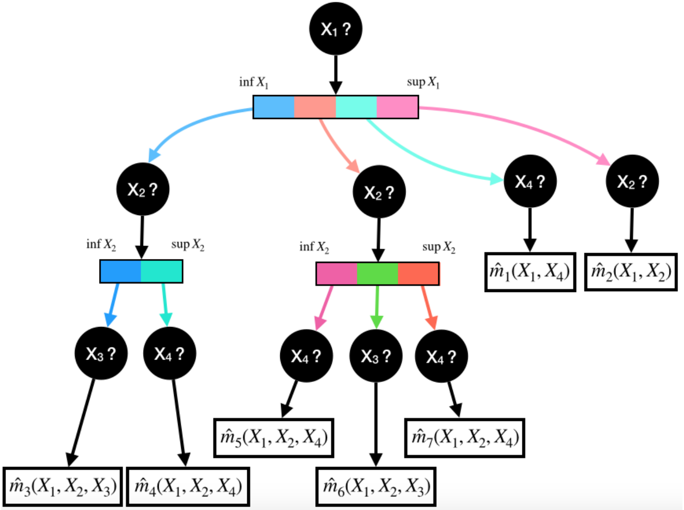
Talk outline

1. Introduction
2. Methodology
3. Benchmarks
4. Results, Planning
5. Results, Reinforcement Learning
6. Discussion

Introduction

Consider couple (X, Y) where Y is a target of interest and X are features which are related to the target, such that one may create a "predictor" of $Y|X$ sufficiently precise. This is the classic supervised learning setting.

Writing p the dimension of X, we wish to find an algorithm which sequentially retrieves q elements of X, $q \ll p$. The algorithm will be constructed such that, based on the partially-observed X, written $\tilde{X}$, the performance of our predictor will be the best possible.



X₁ ?
$\inf X_1$
$\sup X_1$
X₂ ?
X₂ ?
X₄ ?
X₂ ?
$\inf X_2$
$\sup X_2$
$\inf X_2$
$\sup X_2$
$\hat{m}_1(X_1, X_4)$
$\hat{m}_2(X_1, X_2)$
X₃ ?
X₄ ?
X₄ ?
X₃ ?
X₄ ?
$\hat{m}_5(X_1, X_2, X_4)$
$\hat{m}_7(X_1, X_2, X_4)$
$\hat{m}_3(X_1, X_2, X_3)$
$\hat{m}_4(X_1, X_2, X_4)$
$\hat{m}_6(X_1, X_2, X_3)$

Some motivating examples

- Patient follow-up e.g. home healthcare questionnaire
- Prospective calls e.g. fast telemarketing
- Cold start issue with new customer: ask without boring to determine customer cluster
- Balancing acquisition cost with available information
- Storing less data to make as good predictions

- Knowledge-based approaches e.g. in e-learning Nokelainen et al. (2001), in healthcare Dunlop (2019)
- Association rules for fast Myers-Briggs test in Mwamikazi et al. (2014)
- Reinforcement Learning and matrix factorization for the 20 questions game in Chen et al. (2018)
- Model-Based Reinforcement Learning in a huge state and action space healthcare problem in Besson et al. (2018)

- From training set, find best fixed subset of variables in terms of prediction performance
- Decision trees can be seen as a constrained version of the algorithm we aim to reach (top-down optim, simple split function, following splits forget known values)

Methodology

Consider $Y \in \mathcal{Y}$, $\dim(\mathcal{Y}) \geq 1$, our variable of interest and $X \in \mathcal{X}$, $\dim(\mathcal{X}) = p$, the variable vector which can be used to predict Y and can be collected via survey element-by-element. We write the sequence $(\tilde{X}_0, A_0, \ldots, A_{q-1}, \tilde{X}_q)$ the sequence of partial-information and questions asked. $\mathcal{A} = \{1, \ldots, p\}$ is the set of questions and $\tilde{\mathcal{X}}$ the space of $\tilde{X}$.

Objective: design π^*, assigning best next question:

$$\forall \tilde{x} \in \tilde{\mathcal{X}} \qquad \pi^*(\tilde{x}) \triangleq \arg\max_{\pi \in \Pi} \mathbb{E}_{\pi,(X,Y)}[score(\tilde{X}_q, Y) | \tilde{X} = \tilde{x}] \tag{1}$$

where the function *score* measures how accurately we can predict Y based on $\tilde{X}_q$, which we take as:

$$\forall (\tilde{x}, y) \in \tilde{\mathcal{X}} \times \mathcal{Y} \qquad score(\tilde{x}, y) \triangleq -\mathbf{R}(\hat{m}(\tilde{x}), y) \tag{2}$$

where $\hat{m}$ is a prediction function of the target based on partial information and $\mathbf{R}$ is an individual risk measure.

Note: Π is the set of functions mapping $\tilde{\mathcal{X}}$ to $\mathcal{A}$.

The questionnaire process can be modeled by a Markov Decision Process (MDP, Puterman (2014)) $\mathcal{M} = (\tilde{\mathcal{X}}, \mathcal{A}, T, R)$ where $\tilde{\mathcal{X}}$ is the state space, $\mathcal{A}$ is the action space.

T is the transition function mapping $\tilde{\mathcal{X}} \times \mathcal{A} \times \tilde{\mathcal{X}}$ to $[0,1]$ defined as

$$T(\tilde{x}, j, \tilde{x}') = \mathbb{P}_X\big(X_j = \tilde{x}'_j \mid \bigcap_{\{l : \tilde{x}_l \neq \text{``unknown''}\}} \{X_l = \tilde{x}_l\}\big) \tag{3}$$

for all $(\tilde{x}, j, \tilde{x}') \in \tilde{\mathcal{X}} \times \mathcal{A} \times \tilde{\mathcal{X}}$ such that $\tilde{x}_j =$ "unknown", $\tilde{x}'_j \neq$ "unknown" and $\forall l \in \{1, \ldots, p\} \setminus \{j\}, \tilde{x}_l = \tilde{x}'_l$. In other cases, either the question has already been asked (1) or it is incompatible (0).

R is the reward function defined for any terminal $\tilde{x}$ as

$$R(\tilde{x}) \sim \mathbb{P}_{(X,Y)}[score(\tilde{x}, Y) | \tilde{X} = \tilde{x}] \tag{4}$$

and equals 0 otherwise.

The MDP starts with initial state $\tilde{X}_0 =$ "unknown"^{p}, we then ask question A_0, coordinate A_0 is revealed (state $\tilde{X}_1$), we then ask question A_1, reach state $\tilde{X}_2$, and so on and so forth, until a terminal state is reached. In our case, we will stop when q questions will have been asked.

In one of our experiments (Coronary Heart Disease dataset), we extend the setting to the case where initial information is available, allowing us to personalize the initial question. We also consider that to a given action, multiple features may be revealed. Other extensions are discussed in the last section.

We assume that we have at our disposal the set

$$\{(x^{(i)}, y^{(i)}), i \in \{1, \dots, n\}\}$$

consisting of n independent and identically distributed instances from variables (X, Y). From there we can fit prediction function $\hat{m}$, create the set of observed transitions and the set of rewards for terminal states, since we define function *score*. Based on those datasets we propose to learn π^* through the following state-action value functions:

$$\forall (\tilde{x}, a) \in \tilde{\mathcal{X}} \times \mathcal{A} \quad Q_\pi(\tilde{x}, a) = \mathbb{E}_{\pi,(X,Y)}[score(\tilde{X}_q, Y) | \tilde{X}_t = \tilde{x}, A_t = a]. \tag{5}$$

The problem being episodic (q steps) and hierarchical, we can use approximate dynamic programming Bertsekas and Tsitsiklis (1996) to learn the value functions in a backward fashion as presented in figure 1.

Based on the calibrated neural networks $\{\hat{f}_j, j \in \{0, \dots, q-1\}\}$ we would apply the Smart Questionnaire as presented in figure 2.

Methodology

Learning value functions

Algorithm 1: Learning value functions

input (X, Y) pairs : $\{(x^{(i)}, y^{(i)}), i \in \{1, \dots, n\}\}$

learn f_{q-1} as $\hat{f}_{q-1}$, where

$$\begin{aligned} f_{q-1} : \tilde{\mathcal{X}}_{\{layer==q-1\}} \times \mathcal{A} &\to \mathbb{R} \\ (\tilde{x}, a) &\mapsto \mathbb{E}_{(X,Y)}[score(\tilde{X}_q, Y) | \tilde{X}_{q-1} = \tilde{x}, A_{q-1} = a] \end{aligned}$$

for $j \in \{q-1, q-2, \dots, 1\}$ **do**

  learn f_{j-1} as $\hat{f}_{j-1}$, where

$$\begin{aligned} f_{j-1} : \tilde{\mathcal{X}}_{\{layer==j-1\}} \times \mathcal{A} &\to \mathbb{R} \\ (\tilde{x}, a) &\mapsto \mathbb{E}_{(X,Y)}[\max_{a'} \hat{f}_j(\tilde{X}_j, a') | \tilde{X}_{j-1} = \tilde{x}, A_{j-1} = a] \end{aligned}$$

end

return set of networks : $\{\hat{f}_j, j \in \{0, \dots, q-1\}\}$

Methodology

Smart questionnaire

Algorithm 2: Smart Questionnaire algorithm

```
Interacting user knows (x, y)
initialize x̃ ← "unknown"^p
for j ∈ {1, ..., q} do
    select next action/question to ask
    a_j ← arg max_{a∈A} f̂_{j−1}(x̃, a)
    retrieve corresponding element
    x̃_{a_j} ← x_{a_j}
end
return prediction of y : m̂(x̃)
```

In the online setting, we need to handle exploration. Consider we are at iteration t, layer j.

One could consider using a softmax operator over the Q-values estimates:

$$A_j|\tilde{X}_{j-1} = \tilde{x} \sim \frac{\exp\{\beta \hat{f}_{j-1,t}(\tilde{x}, a)\}}{\sum_{a' \in \mathcal{A}} \exp\{\beta \hat{f}_{j-1,t}(\tilde{x}, a')\}} \tag{6}$$

or the upper-confidence bound (UCB) approach:

$$A_j|\tilde{X}_{j-1} = \tilde{x} = \arg\max_{a \in \mathcal{A}} \hat{f}_{j-1,t}(\tilde{x}, a) + \alpha \sqrt{\frac{\log(t)}{n(\tilde{x}, a)}} \tag{7}$$

where n is a pseudo-count function, for example, it could be:

$n(\tilde{x}, a) = \#\{$at step t where elements of $\tilde{x}$ are known as well as the $a^{th}\}$.

Benchmarks

Three toy models were considered in order to test our approach.

In each case we simulated in total 6000 samples, 67% of which are used for training and the rest for testing.

As predictor functions, we used random forests with 100 trees, as implemented in the `R` package `randomForest`. For model #1 we considered the inaccuracy score function and for models #2 and #3 we used the squared prediction error.

For models #2 and #3, we will write ε a standard Gaussian noise generated independently of features X.

A set of rules with binary features.

We consider $p = 8$ mutually independent binary features

$$X_j \sim \mathcal{B}(0.5) \quad \forall j \in \{1, \dots, p\}.$$

Let $E(X)$ denote the union of arbitrarily chosen events:

$$\begin{aligned} E(X) \overset{\Delta}{=} & \{X_1 = X_2 = X_8 = 0\} \cup \{X_6 = 0, X_2 = X_3 = 1\} \\ & \cup \{X_8 = X_1 = X_3 = 1\} \cup \{X_4 = X_5 = X_6 = 0\} \\ & \cup \{X_3 = X_4 = X_2 = 1\} \cup \{X_4 = X_8 = X_1 = 1\} \\ & \cup \{X_3 = X_5 = X_7 = 0\}. \end{aligned}$$

We define $Y|X \overset{\Delta}{=} \mathbb{1}\{\bar{E}(X)\}$.

Benchmarks, Toy models

Model #2

A set of rules with binary and continuous features.

We consider $p = 6$ mutually independent features

$$\forall j \in \{1, \dots, p-1\} \quad X_j \sim \mathcal{B}(0.5), \quad X_p \sim \mathcal{U}[0,1].$$

From there

$$Y|X = \mathbb{1}\{E_1(X) \cap \bar{E}_2(X)\} + 2\mathbb{1}\{E_2(X)\} + 0.2\varepsilon$$

with

$$\begin{aligned} E_1(X) &\stackrel{\Delta}{=} \{X_1 = 0, \{X_2 = 0 \cup X_6 > .7\}\} \cup \{X_4 = X_5 = 0, X_6 > .4\} \\ &\quad \cup \{X_1 = X_3 = 0, X_6 > .8\}, \\ E_2(X) &\stackrel{\Delta}{=} \{X_1 = 1, \{X_3 = 1 \cup X_6 > .7\}\} \cup \{X_3 = X_5 = 1, X_6 > .6\}. \end{aligned}$$

A regression with continuous features.

We consider $p = 8$ mutually independent features

$$X_j \sim \mathcal{N}(0,1) \quad \forall j \in \{1, \ldots, p\}.$$

From there,

$$Y|X = (X_2 + X_3)\mathbb{1}\{X_1 < 0\} + (X_4 + X_5)\mathbb{1}\{X_1 \geq 0\} + \sqrt{2}\varepsilon.$$

Benchmarks,Datasets

Details

Three benchmark datasets where considered:

- the Boston Housing dataset Harrison Jr and Rubinfeld (1978),
- the more recent AMES dataset De Cock (2011) and
- the Coronary Heart Disease dataset FHS (2020); Aman Ajmera (2020).

The first two contain house prices and characteristics jointly. Because of the relatively low sample sizes we relied on linear regression models as prediction functions, rather than non-parametric models. For the Coronary Heart Disease problem, having relatively large sample size we used extreme gradient boosting with validation split for early stopping, relying on R package `xgboost`.

Boston Housing This dataset contains 506 observations (one per suburb) and 13 variables, amongst which: the median value of owner-occupied homes (the target), crime rate, average number of rooms, pupil-teacher ratio.

AMES This dataset consists of 2930 observations of house value (log-scaled) and 81 characteristics such as overall quality, year of construction, surface information. The set of features was brought down to the following ten variables: *OverallQual, GrLivArea, YearBuilt, GarageCars, TotalBsmtSF, GarageArea, X1stFlrSF, FullBath, YearRemodAdd, LotArea*.

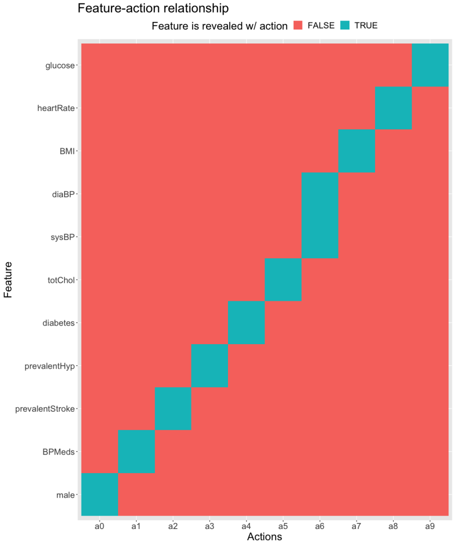
Coronary Heart Disease, CHD (i)

This dataset contains 4238 observations of patients: socio-demographic information (e.g. gender), medical information (e.g. diabetes), medical examination (e.g. glucose) and finally whether the patient developed Coronary Heart Disease during the following ten year period.

For this problem, we assumed gender as an already-known feature and one-to-one relationship between actions and features except for one: action 6 reveals diastolic and systolic blood pressure simultaneously.

Benchmarks,Datasets

Coronary Heart Disease, CHD (ii)

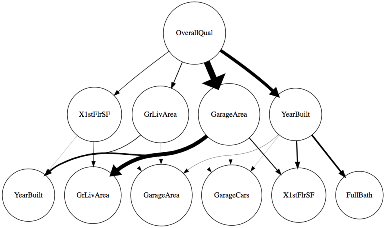


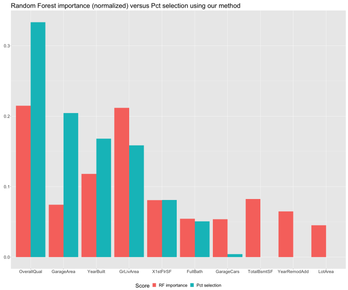
Results, Planning

For each problem, we replicate the train/test split at random 10 times, calibrate our Smart Questionnaire as well as three baselines on the training set and evaluate on the test set. We report the test set performance average and its standard deviation in the table below.

Benchmark	**Metric**	Bound	Oracle	SmartQ	Best q subset	CART depth $= q$
Toy models						
#1	AUC	1	1 (0)	0.87 (0.01)	0.75 (0.02)	0.81 (0.01)
#2	RMSE	0.2	0.3 (0.01)	0.37 (0.01)	0.46 (0.01)	0.41 (0.01)
#3	RMSE	$\sqrt{2}$	1.56 (0.02)	1.57 (0.03)	1.83 (0.03)	1.84 (0.02)
Datasets						
Boston	RMSE		4.99 (0.57)	4.92 (0.54)	5.33 (0.54)	5.16 (0.64)
AMES	RMSE		4.3 (0.22)	4.56 (0.14)	4.67 (0.19)	5.62 (0.15)
CHD	AUC		69.73 (1.92)	62.38 (3.2)	61.04 (4.35)	60 (3.41)

For a given Smart Questionnaire, we can plot the paths taken on the test set (left), with the thickness indicating the frequency. To be compared with random forest variable importance in supervised learning setting. This is informative as to when each variable is deemed important for prediction task.

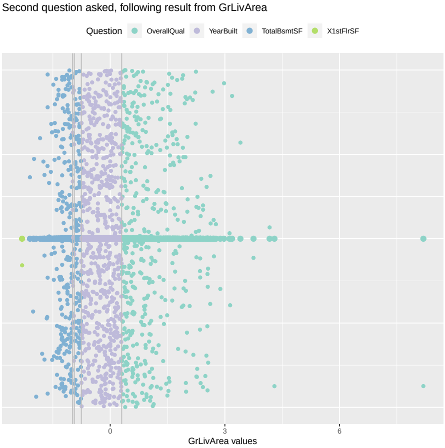




Results, Planning

Visualization, AMES dataset (i)

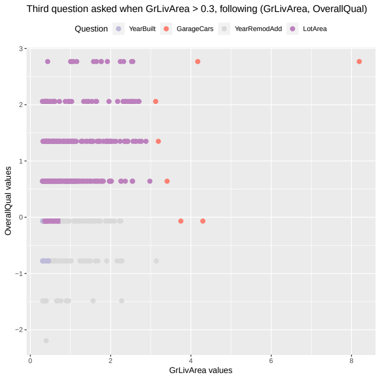
For AMES dataset, we look at the decisions made by the Smart Questionnaire on a test set, which starts by asking for *GrLivArea*.

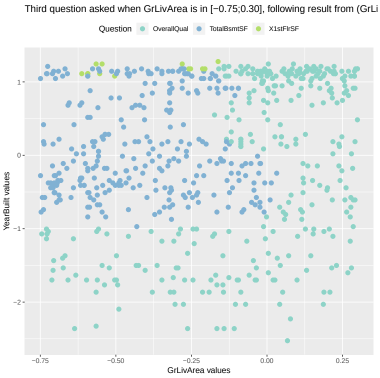


Results, Planning

Visualization, AMES dataset (ii)

We zoom in now on specific parts of the questionnaire.





Results, Reinforcement Learning

We considered toy model #3 presented in preceding section: consider $p = 8$ mutually independent random gaussians features

$$X_j \sim \mathcal{N}(0,1) \quad \forall j \in \{1, \ldots, p\}.$$

The target variable is defined by the simple linear model

$$Y = \begin{cases} X_1 + X_2 + \sigma\varepsilon & \text{if } X_3 < 0 \\ X_4 + X_5 + \sigma\varepsilon & \text{otherwise} \end{cases}$$

where $\varepsilon \sim \mathcal{N}(0,1)$ is the noise random variable, independent of X. Parameter σ, which controls the noise level in the model will be set to $\sigma^2 = 0.4$ in order to have a Sound-to-Noise ratio of 5. We consider the problem of gradually sampling data using the questionnaire i.e. gathering only $q = 3$ elements of feature vector X at a time.

Results, Reinforcement Learning

Process specifics

- initialize with 32 samples for each q-question combination
- at each iteration, we take sample $32 \cdot \binom{p}{q} = 1792$ following exploration policy
- half of the data is left for training $\hat{m}$ and the other half is for the neural networks
- at each iteration, we follow greedily the current questionnaire on a free-test dataset of 2000 samples, to track for final performance

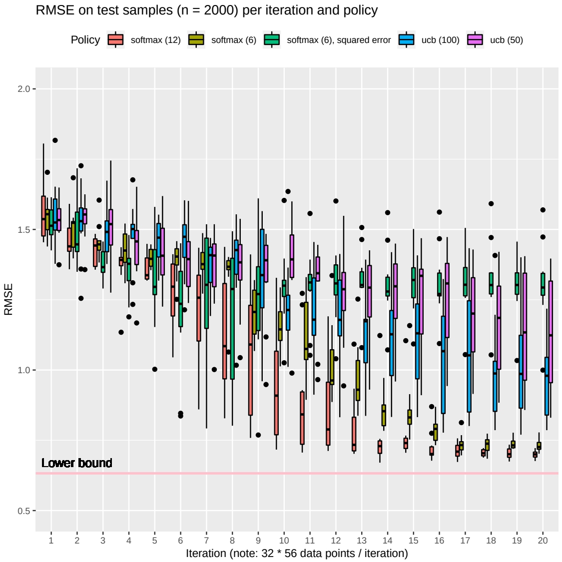
Results, Reinforcement Learning

Input specifics

- softmax and exploration policies are considered with different parameters
- we try two different score functions: the squared error and the square root absolute error
- algorithm random forest is used for $\hat{m}$
- networks fully-connected neural network, with four hidden layers of 24 neurons, input of size 24 (the first 16 representing the partially-known vector, the last 8 indicating which action is selected) and with output size 1
- experiments were repeated ten times to provide proper average and deviation estimates

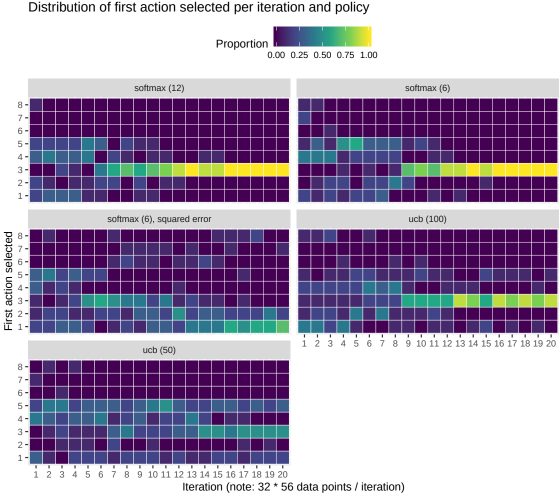
Results, Reinforcement Learning

Test sample performance (i)



Results, Reinforcement Learning

Test sample performance (ii)



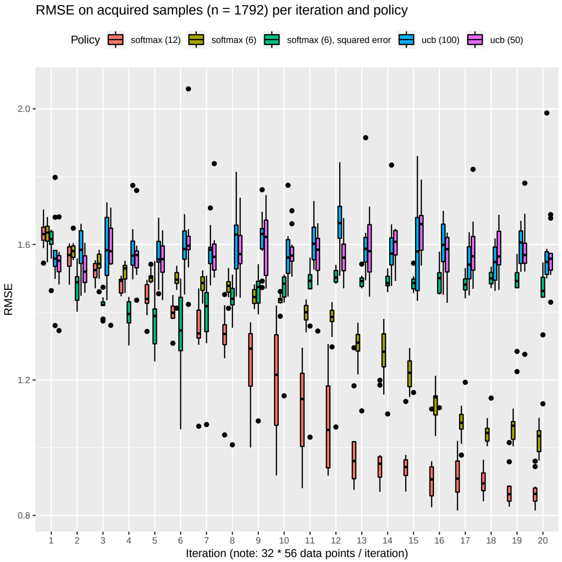
Results, Reinforcement Learning

Test performance summary

1. UCB methods take a longer time to reduce RMSE in contrast with softmax exploration,
2. UCB methods also present overall more variability than softmax experiment do,
3. The reward distribution matters, as the softmax policy relying on original squared errors is the only on which seems to stuck to a floor level, two times larger than the lower bound ($\sigma = \sqrt{0.4}$).

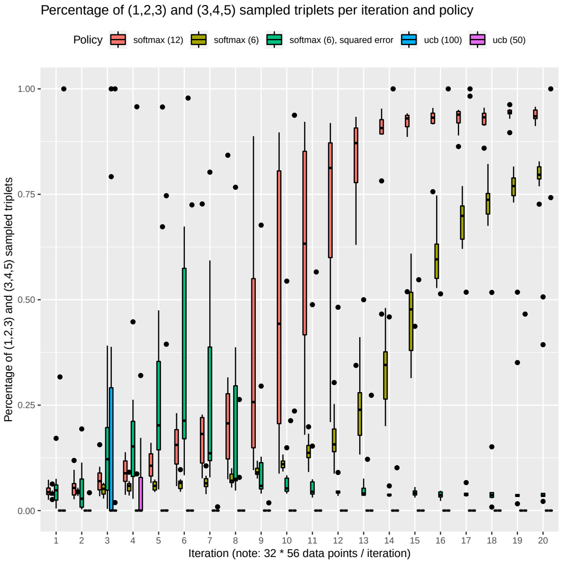
Results, Reinforcement Learning

Acquired sample performance (i)



Results, Reinforcement Learning

Acquired sample performance (ii)



Results, Reinforcement Learning

Acquired sample performance summary

1. Only the softmax policies focus their exploration phase on the actual important triplets, and consequently couples
2. For UCB exploration policies however, the pertinent triplets are very rarely sampled (there are some outliers i.e. good scenarios) and the pertinent couples are sampled in a very variable manner across our experiments: this is clearly not robust throughout experiments
3. UCB methods tend to focus on a few triplets, as in most cases the maximum percentage was largely higher than 50%, which simply constitutes less exploration

Results, Reinforcement Learning

Note

Something we thoughtfully omitted, is that the reward function changes throughout iterations, since it is defined based on our prediction $\hat{m}$ of m, which is updated as more data is actually available. We are fortunate enough, or in a sufficiently simple setting at least, for this approach to be working. Now, it seems reasonable that we would have some idea of the variance of our prediction, which opens the possibility to learn not the exact error but a lower upper bound on it (optimism-based approach).

The UCB strategy is clearly at a disadvantage compared to the softmax strategy from the very nature of the experiment: there are quite few times of exploration, hence the UCB criterion, which is deterministic by construction, explores much less than the stochastic policy.

Discussion

We built an adaptive predictive-questionnaire under constraint over the number of questions, motivated by the important balance between data acquisition and user experience.

We framed it as a Markov Decision Process and used approximate dynamic-programming, exploiting the nature of the problem.

We have shown on six instances that our approach outperforms

1. a decision tree submitted to the same budget constraint of questions
2. classic models based on a fixed (and pertinent) subset of questions.

We have also shown that we can deal with data initially available to personalize the whole questionnaire, and it seems we can also deal with such problem in the online setting.

Discussion

Further research

As a continuation of this work, we have a few ideas for further research.

- Scaling with (p, q): an intelligent exploration strategy might be needed
- Stopping criterion: waiting until accuracy is sufficiently high
- Performance criterion: stochastic action-features discovery, with associated costs
- Truthful responses and HCI: is the data reliable, especially as we sample it in a non-random manner ? safe (reinforcement learning) approach and POMDPs

References

Aman Ajmera (2020). Framingham Heart study dataset. https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset. Accessed: 2020-05-01.

Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Athena Scientific.

Besson, R., Pennec, E. L., Allassonniere, S., Stirnemann, J., Spaggiari, E., and Neuraz, A. (2018). A model-based reinforcement learning approach for a rare disease diagnostic task. *arXiv preprint arXiv:1811.10112*.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

Chen, Y., Chen, B., Duan, X., Lou, J.-G., Wang, Y., Zhu, W., and Cao, Y. (2018). Learning-to-ask: Knowledge acquisition via 20 questions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1216–1225.

De Cock, D. (2011). Ames, Iowa: Alternative to the Boston housing data as an end of semester regression project. *Journal of Statistics Education*, 19(3).

Dunlop, M. D. (2019). Ontology-Driven, Adaptive, Medical Questionnaires for Patients with Mild Learning Disabilities. In *Artificial Intelligence XXXVI: 39th SGAI International Conference on Artificial Intelligence, AI 2019, Cambridge, UK, December 17–19, 2019, Proceedings*, page 107. Springer.

FHS (2020). Framingham Heart Study, Three Generations of Dedication. https://framinghamheartstudy.org. Accessed: 2020-05-01.

Harrison Jr, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134.

Magelssen, M., Supphellen, M., Nortvedt, P., and Materstvedt, L. J. (2016). Attitudes towards assisted dying are influenced by question wording and order: a survey experiment. *BMC medical ethics*, 17(1):24.

Mwamikazi, E., Fournier-Viger, P., Moghrabi, C., Barhoumi, A., and Baudouin, R. (2014). An adaptive questionnaire for automatic identification of learning styles. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 399–409. Springer.

Nokelainen, P., Niemivirta, M., Kurhila, J., Miettinen, M., Silander, T., and Tirri, H. (2001). Implementation of an adaptive questionnaire. In *Proceedings of the ED-MEDIA Conference*, pages 1412–1413.

Provost, F., Melville, P., and Saar-Tsechansky, M. (2007). Data acquisition and cost-effective predictive modeling: targeting offers for electronic commerce. In *Proceedings of the ninth international conference on Electronic commerce*, pages 389–398.

Puterman, M. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.