Aim Low: Correlation-based Feature Selection for Model-based Reinforcement Learning

Shitian Shen
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
sshen@ncsu.edu

Min Chi
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
mchi@ncsu.edu

ABSTRACT

We explored a series of feature selection methods for model-based Reinforcement Learning (RL). More specifically, we explored four common correlation metrics and based on them, we proposed the fifth one named Weighed Information Gain (WIG). While much existing correlation-based feature selection methods mostly explored high correlation by default, we explored two options: *High* vs. *Low*. The former selects the next feature that has the highest correlation measure with existing selected ones while the latter selects the one with the lowest correlations. The 10 correlation-based methods were compared against previous feature selection methods for model-based RL across several datasets collected from two vastly different intelligent tutoring systems. Our results showed that the 10 correlation-based methods significantly outperform all other methods across all datasets. Among the five correlation metrics, WIG performed best. Surprisingly, for each of correlation metrics, the low option significantly outperform its high correlation peer and thus it suggests that low correlation-based feature selection methods are more effective for model-based RL than high ones.

1. INTRODUCTION

Optimal decision making in complex interactive environments is challenging. In Intelligent Tutoring Systems (ITSs), for example, system's behaviors can be treated as a sequential decision process where at each step system selects an appropriate action from a set of alternatives. Each of these system decisions will affect the user's subsequent actions and performance. Its impact on outcomes cannot be observed immediately and the effectiveness of each decision is dependent upon the effectiveness of subsequent decisions. *Pedagogical strategies* are policies that are used to decide what system action to take next in the face of alternatives.

Reinforcement Learning (RL) is one of the best machine learning approaches for decision making in interactive environments. RL focuses on inducing optimal policies on what action(s) an agent should take in any context that would maximize the agent's cumulative reward. While various RL approaches have shown promising, existing RL approaches tend to perform poorly when the interactive environment is complex in that many factors can impact desired outcomes yet not fully understood. Our general approach is to start from a collection of potentially relevant features and to apply *feature selection* methods to narrow them down to a compact and effective state representation. Many feature selection methods such as Least-squares temporal difference (LSTD) with lasso regularization [11], Monte-Carlo tree search algorithm [5] have successfully applied for RL. However, most of then are designed for model-free RL and we used model-based RL (Section 3).

In this paper, we proposed a series of correlation based feature selection methods by exploring different correlation metrics. Correlation-based methods have been widely used in supervised learning, where we use input state feature space X to predict output label Y and previous approaches mainly select the subsets of X with the *highest* correlation with the output label Y [8, 21]. However, for RL there is no output label Y and thus, to apply correlation-based feature selection methods directly to RL, we explored two options: *High* and *Low*. The former is to select the next feature that is the **most correlated (High)** with the selected ones while the latter option is to select the **least correlated (Low)** one. Theoretically speaking, choosing the most correlated feature may be effective since the selected feature is more likely to be related to decision making, however it may not make more contribution than the current selected feature set does. On the other hand, choosing the least correlated feature may raise the diversity of selected feature set and enrich the state representation, however it takes a risk of selecting irrelevant or noisy features.

In short, we explored both high and low options for five correlation metrics and resulted in 10 correlation-based methods. We compared them against an ensemble method, the methods involved in [3] referred as **RLPreviousFS** for the rest of paper, and the random feature selection method across several datasets collected from two vastly different ITSs: one is a data-driven logic tutor named Deep Thought and the other is a natural language physics tutor named Cordillera.

2. RELATED WORK

In general, existing feature selection for RL can be classified into three categories [6]: Filter, Wrapper and Embedded. Filter approaches can be seen as a preprocessing procedure in that it usually employs a ranking function so that either a fixed number of features with the highest rank or a feature set above a preset threshold value will be selected from the high-dimensional state space. This process is independent from the subsequent model learning process. For RL, the ranking function is generally based on which state feature subset would directly influence the rewards. For example, Morimoto et al. applied *kernel dimension reduction* to evaluate the conditional independence among state features and those with the most impacts on the next-time-step rewards are selected [14]. Hirotaka and Masashi [7] proposed a filter-type approach by directly evaluating the independence between immediate reward and state-feature sequences using conditional mutual information. However, it is not clear how their approach can be applied when immediate reward is not directly observable and only delayed reward is present.

Wrapper approaches search feature space and generate several candidate feature subsets, evaluate each subset using a learning algorithm, and then select the subset with the best performance. For example, Gaudel and Sebag applied Monte-Carlo tree search algorithm to generate candidate feature subsets and then evaluate the goodness of feature subset using the predefined score function [5]. In addition, Keller, et al applied LSTD to approximate value function, selected a feature subset by implementing *Neighborhood Component Analysis* to decompose approximation error, which can be used to evaluate the goodness of the feature subset [9]. Similarly, in *LSPI-FFS* Li, Williams and Balakrishnan also applied LSTD to approximate value function using linear model. They updated the parameters of the linear model through gradient descent and selected a feature subset with largest magnitude of weight [13].

Embedded approaches for RL conduct feature selection and policy induction process simultaneously. Kolter and Ng applied LSTD with *Lasso* regularization to approximate value function as well as to select effective feature subset [11]. Bach explored the penalization of approximation function by using *Multiple Kernel learning* (MLKL)[2]. Wright, Loscalzo and Yu proposed *IFSE-NEAT*, the feature selection embedded in neuroevolutionary function, which approximates the value function, and features are selected based on their contributions to the evolution of topology of network[20].

In short, while much of prior research has done on feature selection for RL, most of them is for model-free RL. For Model-based RL, Chi et al. investigated 10 filter-based methods (**RLPreviousFS**) [3]. These methods were implemented to derive a set of various policies, where features are selected mainly based on the single feature performance and the covariance in training data. Their results showed there was no consistent winner among the ten feature selection methods and in some particular cases these methods performed no better than the random baseline method. Therefore, much research on feature selection for model-based RL is needed.

3. REINFORCEMENT LEARNING & MARKOV DECISION PROCESS

Generally speaking, RL can be divided into two categories: **model-free** and **model-based**. Model-free RL [4] typically uses samples to learn a value function, from which a policy is implicitly derived. Model-based RL, by contrast, first builds up a model from samples and then compute a policy based the model. Both approaches have their own strengths and weaknesses. Model-free methods are appropriate for domains where data collection is inexpensive and trivial. Model-based methods, on the other hand, are suitable when collecting data is expensive. Given the high cost of collecting training data in our task, we focused on model-based RL and used a Markov Decision Process (MDP) framework.

MDP is defined as a tuple $\langle S, A, T, R\rangle$. S denotes state space, which reflects the generalization of interactive environment; actions A are agent's possible behaviors; reward function R can be immediate or delayed feedback from environment respect to agent's behavior and $R^a_{SS'}$ denotes the reward of transiting from state S to state S' by taking action a; transition probabilities T are defined as $T = \{p(S_j|S_i, A_k)\}^{k=1,\cdots,m}_{i,j=1,\cdots,n}$, which is estimated from training corpus. More specifically, $T^a_{SS'} = p(S'|S, a)$ denotes the probability of transiting from state S to state S' by taking action a.

Once the tuple $\langle S, A, T, R\rangle$ is set, we transform the problem of inducing effective pedagogical strategies into computing an optimal policy in an MDP by dynamic programming approaches. More specifically, we calculate the value function $V^\pi(S)$ under a policy π though Bellman equation[17], which is defined as:

$$\begin{aligned} V^\pi(S) &= E_\pi(R_t|S_t = S) \\ &= \sum_a \pi(S, a) \sum_{S'} T^a_{SS'} \left[R^a_{SS'} + \gamma V^\pi(S')\right] \end{aligned}$$

where γ is a constant called discount factor. The optimal value function can be estimated by

$$V^*(S) = \max_\pi V^\pi(S)$$

Then we can derive the optimal policy corresponding to the optimal value function $V^*(S)$. Here we used the toolkit developed by Tetreault and Litman [18]. Besides inducing an optimal policy, Tetreault, & Litman's toolkit also calculate the Expected Cumulative Reward (ECR) for the induced policy. The ECR of a policy is derived from a side calculation in the policy iteration algorithm: the V-values of each state, the expected reward of starting from that state and finishing at one of the final states. More specifically, the ECR of a policy π can be calculated as follows:

$$ECR_\pi = \sum_{i=1}^{n} \frac{N_i}{N_1 + \cdots + N_n} \times V(s_i) \qquad (1)$$

Where $s_1, \cdots, s_n$ is the set of all starting states and $V(s_i)$ is the V-values for state s_i; N_i is the number of times that s_i appears as a start state in the model and it is normalized by dividing $\frac{N_i}{N_1+\cdots+N_n}$. In other words, the ECR of a policy π is calculated by summing over all the initial start states in the

space and weighting them by the frequency with which each state appears as a start state. The higher the ECR value of a policy, the better the policy is supposed to perform.

In our application, we defined our action set A and reward function R in Section 5. However the state space S is not well-defined, where each state is a vector representation composed of a fixed number of state features $F = \{F_1, F_2, ..., F_p\}$. Our approach is to apply various feature selection methods to narrow a wide set of feature space to a compact and effective subset that would model student learning process accurately.

4. METHODOLOGY

In this section, we first describe the five basic correlation metrics we used and then describe our general feature selection procedure. More specifically, we will describe our 10 correlation-based methods, the ensemble method, and finally briefly describe the RLPreviousFS methods.

4.1 Five Correlation Metrics

In order to quantize correlation among features, we used five correlation metrics. The first four are commonly used in supervised learning and here we will investigate whether they can be applied to RL. We proposed the fifth one, Weighted Information gain, by combining the four commonly used metrics and adapting them based on the characteristic our task and datasets. More specifically, we have:

1. Chi-squared (CHI)[22]: a statistical test used to identify whether the distribution of a categorical variable differ from the other one, which induces the independence between two variables. CHI is usually applied to evaluate the independence of two variables in mathematical statistics.

2. Information gain (IG)[12]: it measures the differ between the uncertainty of a variable Y and the uncertainty of Y given variable X as conditional information. It is calculated as:

$$IG(Y, X) = H(Y) - H(Y|X)$$

where $H()$ is called entropy function, measure uncertainty of a variable. IG evaluates the certainty of variable Y obtained from variable X, which can be treated as one type of correlation between X and Y. IG has the bias towards the variable with a large number of distinct values.

3. Symmetrical certainty (SU)[21]: it is defined as:

$$SU(Y, X) = \frac{H(Y) - H(Y|X)}{H(X) + H(Y)}$$

SU evaluates the correlation between two variables by normalizing IG. SU compensates the weakness of IG and it is a symmetrical measurement, which treats a pair of variables symmetrically.

4. Information gain ratio (IGR)[10]: it's the ratio of information gain to the intrinsic information, which is the entropy of conditional information. IGR can be represented as:

$$IGR(Y, X) = \frac{H(Y) - H(Y|X)}{H(X)}$$

Comparing with IG, IGR takes the uncertainty of conditional information into account with purpose of removing bias of selecting variable with many distinct values. However, IGR is not a symmetrical measurement ($IGR(X, Y) \neq IGR(Y, X)$).

5. Weighted Information gain (WIG): it is proposed as:

$$WIG(Y, X) = \frac{H(Y) - H(Y|X)}{(H(Y) + H(X))H(X)}$$

We propose WIG by combining IG, SU and IGR. Comparing with IGR, WIG normalized IG by considering the uncertainty of both variables X and Y and also compensate the weakness of IG. Comparing with SU, although WIG is not symmetrical measurement. Based on the above equation, WIG sets more weight for variable X. In our application, WIG is used for evaluating the correlation between current selected feature set Y with the new feature X .

For each of the five correlation metrics, we explored two options: High and Low, which resulted in 10 correlation-based methods named five High methods: CHI-high, IG-high, SU-high, IGR-high, WIG-high and five Low methods: CHI-low, IG-low, SU-low, IGR-low, and WIG-low. Our goal is to investigate which option is better: high vs. low and which of the five correlation metric performs the best.

4.2 Correlation-based Feature Selection

In this project, we followed a forward stepwise feature selection procedure in that: given current selected feature set, our correlation-based methods select the feature forwardly based on the five correlation metrics described above.

Algorithm 1 Correlation-based Feature Selection Algorithm

Require: Ω: Feature space; $\mathcal{D}$: Training data; $\mathcal{N}$: Maximum number of selected features
Ensure: $\mathcal{S}^*$: Optimal feature set
for f_i in Ω **do**
  $ECR_i \leftarrow$ CALCULATE-ECR($\mathcal{D}$, f_i)
end for
Add f^* with highest ECR to $\mathcal{S}^*$
while SIZE($\mathcal{S}^*$) $< \mathcal{N}$ **do**
  for f_i in $\Omega - \mathcal{S}^*$ **do**
    $C_i \leftarrow$ CALCULATE-CORRELATION($\mathcal{S}^*$, f_i, m)
  end for
  $\mathcal{F} \leftarrow$ SELECTTOP(C, 5, reverse) ▷ Select top 5 features based on correlation metrics
  for f_i in $\mathcal{F}$ **do**
    $ECR_i \leftarrow$ CALCULATE-ECR($\mathcal{D}$, $\mathcal{S}^* + f_i$)
  end for
  Replace $\mathcal{S}^*$ by $\mathcal{S}^* + f_i$ with higgest ECR
end while

Algorithm 1 shows the concrete process of our correlation-based feature selection procedure. It contains three major

parts: in the first part (lines 1–4), it constructs MDPs for each single feature, induces a single-feature policy and calculates it ECR. Then the feature with highest ECR is added into current optimal feature set. In the second part (lines 6–9), it evaluates the correlations between current optimal feature set $\mathcal{S}^*$ with other features $f_i \in \Omega - \mathcal{S}^*$, ranks the correlations, and then selects the top 5 highest ones for high correlations or the bottom 5 lowest ones for low correlations. They are selected to form a feature pool $\mathcal{F}$. In the third part (lines 10–13), several candidate feature sets are generated by combining current optimal feature set $\mathcal{S}^*$ with each feature $f_i \in \mathcal{F}$. Then ECR for each candidate feature set can be evaluated by applying *Calculate-ECR* function. Current optimal feature set $\mathcal{S}^*$ will be replaced by the candidate feature set with highest ECR. The algorithm will terminate until the size of optimal feature set reaches maximum number $\mathcal{N}$. The third part can be treated as the process of wrapper approach where several candidate feature sets are evaluated by the RL method. Therefore, our correlation-based methods are the combination of filter and wrapper approaches.

4.3 Ensemble Method

Our ensemble approach combines the 10 proposed correlation-based methods and 4 *RL-based* methods (Section 4.4), which are most effective methods among RLPreviousFS. Its procedure is similar to that of correlation-based method except the second part (lines 6–9). The ensemble approach integrates the features generated from each method and generates a relatively big feature pool $\mathcal{F}$. The maximum size of $\mathcal{F}$ is up to 70 but often smaller because of the overlapping feature sets. Note that it is still much larger than any of our 10 correlation-based methods which has 5 candidates for each step. After generating the feature pool, the ensemble method jumps to the third part (lines 10-13) of Algorithm 1. At each step, the ensemble method explores feature sets by adding the feature with maximum ECR.

4.4 RLPreviousFS

Chi et. al [3] grouped RLPreviousFS into three categories:1) four RL-based methods; 2) two PCA-based method, which selects features with the high correlation with principle components; 3) four PCA&RL-based methods, which use RL-based methods to select features from a candidate feature set which is generated from PCA-based method. All three categories can be seen as the filter approaches.

5. TRAINING DATASETS

5.1 Two Deep Thought Datasets

Deep Thought (DT)[15] is a data-driven ITS . It is a rule-based system where students need to select different rules to complete logic proof problems. In DT, we focused on a problem level decision named problem solving (PS) vs. Worked Example(WE). More specifically, when starting the next training problem, the tutor will make a simple decision: "should it ask student to solve the next problem **(PS)**, or should it provide an example to show the student how to solve the next problem **(WE)**".

Our training dataset includes a total of 303 undergraduate CS students who used DT as part of class assignment in Fall 2014 and Spring 2015. The average amount of time spent in the tutor was 416.60 minutes. To induce RL policies, a total of 134 features were extracted from the student-system log files. The reward function in DT dataset is calculated based on level score $LevelScore_i$ where $i \in [1, 6]$. Particularly, we designed two type of reward: immediate and delay reward. Immediate reward is defined as $R_i = LevelScore_i - LevelScore_{i-1}$ where $i \in [1, 6], R_1 = LevelScore_1$, it reflects the change of students' performance level by level. Delayed reward is represented as $R_{delay} = LevelScore_6 - LevelScore_1$, which determines the change of students' performance across all levels. For the convenience, we denote the two DT datasets with immediate reward as *DT-Immed* and that with delayed reward as *DT-Delay* respectively.

5.2 Six Cordillera Datasets

Cordillera [19] is a natural language tutoring system teaching college introductory physics. Different from DT tutor system, Cordillera requires students to input their answer by natural language free text. The data collection consists of the following stages: 1) background survey; 2) studying textbook and prerequisite materials, 3) taking a pretest; 3) training on Cordillera, 4) and taking a post test. Cordillera makes step-level decision: *Elicit/Tell* **(ET)**. The ET decision means "should the tutor system *elicit* the next problem-solving step for student, or should it *tell* student the instruction of next step directly".

Our training corpus involves 64 students. In Cordillera, there are five primary Knowledge Components (KCs): Definition of Kinetic Energy (KE), Gravitational Potential Energy (GPE), Spring Potential Energy (PE), Total Mechanical Energy (TME), and finally Conservation of Total Mechanical Energy (CTME). In STEM domains such as math and science, it is commonly assumed that the relevant knowledge is structured as a set of independent but co-occurring KCs. A KC is "a generalization of everyday terms like concept, principle, fact, or skill, and cognitive science terms like schema, production rule, misconception, or facet" [19]. For the purposes of ITSs, these are the atomic units of knowledge. It is assumed that a tutorial dialogue about one KC (e.g., kinetic energy) will have no impact on the student's understanding of any other KC (e.g, of gravity). This is an idealization, but it has served ITS developers well for many decades, and is a fundamental assumption of many cognitive models [1, 16]. Given the KCs' independence assumptions, we will apply RL to induce KC-specific pedagogical strategies for each of the five primary KCs individually. Moreover some steps in Cordillera have mixed KC, thus we also apply RL to induce pedagogical policies irregardless of the KCs involved (denoted by Across). In short, we have a total of **six** Cordillera KC datasets, one per KC for the five primary KCs and one KC-general for the Across policy. Each of the KC datasets contains 50 state features and to induce RL-rules, we used the delayed reward defined as student Normalized Learning Gains (NLGs): $NLG = \frac{Posttest - Pretest}{MaximumScore - Pretest}$. Here $MaximumScore$ is the maximum score a student can get and for both pretest and posttest, the maximum score is set to be 1.

6. EXPERIMENT & RESULT

To evaluate the effectiveness of induced policies, we set the maximum number of selected features to be 6 considering the size of our training datasets. In this section, we present

Table 1: The highest ECR Induced by Correlation-based Methods Across Eight Datasets

ITS	Data	CHI		IG		SU		IGR		WIG	
		High	Low	High	Low	High	Low	High	Low	High	Low
DT	Immed	55.89	129.82	53.87	95.81	53.87	95.81	53.87	95.81	59.04	**143.16***
	Delay	8.89	12.56	8.89	12.58	10.73	12.58	8.94	**15.43***	8.94	**15.43***
Cordillera	KE	5.86	6.75	5.86	6.75	5.86	6.75	5.57	**7.64***	5.57	7.62
	GPE	10.47	13.39	11.80	13.39	11.21	13.39	11.10	**17.23***	10.82	**17.23***
	SPE	12.67	17.17	12.67	14.88	12.67	**18.02***	10.83	**18.02***	10.27	**18.02***
	TME	7.34	7.96	7.57	9.42	7.47	9.42	6.98	**10.04***	6.40	**10.04***
	CTME	23.01	32.71	24.01	31.22	24.31	31.22	23.01	**33.24***	23.01	**33.24***
	Across	1.77	2.26	1.77	2.26	1.77	**2.57***	1.77	2.26	1.77	**2.57***

Note: The best ECR among 10 methods for each dataset is highlighted by *.

Table 2: Overall Evaluation Across Eight Datasets

	DT		Cordillera					
	Immed	Delayed	KE	GPE	SPE	TME	CTME	Across
Low Correlation	**143.16**	**15.43**	**7.64**	**17.23**	**18.02**	**10.04**	**33.24**	2.57
High Correlation	59.03	10.72	5.85	11.80	12.67	7.57	24.31	1.71
Ensemble	127.79	12.61	7.33	16.40	16.95	9.12	32.06	**2.68**
RLPreviousFS	60.28	12.56	6.17	14.41	11.90	7.15	24.60	2.03
Random	8.53	7.62	4.26	7.34	10.52	4.78	22.02	1.20

the experimental analysis of the correlation-based methods, the ensemble, the RLPreviousFS used in previous research, and random feature selection methods which is our baseline method.

6.1 Comparing correlation-based methods

In this section, we want to answer two questions:
1) which option is better for model-based RL: High vs. Low;
2) which of the five correlation metrics performs the best.

High VS Low. Table 1 shows the performance of the 10 correlation based methods across eight training datasets: two DT and six Cordillera datasets. The rows represent the eight datasets while columns represent the 10 correlation-based methods. Each cell in Table 1 shows the highest ECR of the policy generated from the corresponding correlation-based feature selection method on the corresponding dataset when the number of features varies from 1 to 6.

Table 1 shows that for each of five correlation metrics, the low correlation-based method significantly outperform its high correlation-based peer. For *DT-Immed* dataset, the ECR of WIG-low is 143.16, while ECR of WIG-High is only 59.04; the former is 140% higher than the latter. Similarly, the $ECRs$ of CHI-low and CHI-High are: 129.82 vs. 55.89 and the former is 132% higher than the latter. The similar results is true across all five correlation metrics and across all eight datasets.

Moreover, the out-performance of the Low option over the High option seems to be more prominent on DT datasets than Cordillera datasets. For DT data, the average percent increase for the low correlation methods over the high correlation methods is 75.35%, the maximum percent increase is 142.48% and the minimum percent increase is 17.24%. For Cordillera KC datasets, the average percent increase for the low correlation methods over the high ones is 35.15%, the maximum percent increase is 75.46% and the minimum percent increase is 8.45%. On average the low correlation methods outperform the high correlation peers by 45.2%.

To summarize, our results showed that the low correlation option is more suitable for the model-based RL than the high correlation option. It indicates that it is important to include a variety of features in the state representation for applying RL to induce pedagogical policies.

Five Correlation Metrics. In Table 1, for each of the eight datasets, we highlight the best ECR of the induced policies by *. Table 1 shows that the WIG is the consistent winner in that it has the best ECR for all datasets except for *KE*. On the KE dataset, WIG-Low performance is slightly lower than the best policy: 7.62 for WIG-Low vs. the highest 7.64 for IGR-Low. Following WIG, IGR is the second best in that it has the highest ECR for six out of eight datasets. Note that WIG and IGR together produced all the best policies across all eight datasets and they overlapped on *DT-Delay, GPE, SPE, TME, CTME*. Except for WIG and IGR, the remaining three metrics only induced 2 best policies and both are found by SU-Low. In short, our proposed WIG performed the best among the five correlation metrics followed by IGR.

6.2 Overall Evaluation

Table 2 shows the overall comparison among all feature selection methods. With the purpose of simplicity, for the five low-correlation methods, the five high-correlation methods and the RLPrevousFS methods, we select the best one from each category. Thus, Table 2 will compare the five categories of feature selection methods: the best of the five Low-

correlations, the best of the five High-correlations, the ensemble, the best of RLPreviousFS and the random method.

In Table 2, rows denote the five categories and columns show the eight datasets. Table 2 shows that as expected the random method performs the worst across all datasets. In addition, the best of the low correlation-based methods outperforms all other methods in all datasets except in the *Across* dataset, where the ensemble method performs slightly better that the best of the low correlation-based methods. On average, the best low correlation-based method increases over the best of RLPreviousFS by 43.87% and over the ensemble method by 9.05%. In addition, the ensemble method improves over the best of RLPreviousFS on average 36.46%. To summarize, we can rank the five categories of methods as Low correlation-based > Ensemble > High correlation-based, RLPreviousFS $\gg$ Random.

7. CONCLUSIONS & FUTURE WORK

In this paper, we proposed 10 correlation-based feature selection methods for model-based RL. Our result clearly showed that the low correlation-based methods are more effective than the ensemble, the high correlation-based, the RLPreviousFS, and the random method. Among the five correlation-based metrics, our proposed WIG performed the best. WIG found the best policies across all eight datasets except that on KE, its performance is only slightly lower than the best one which is found by IGR.

While in supervised learning features associated with highest correlation are generally selected, for model-based RL selecting the next feature with lowest correlation is more effective. Moreover, it is surprising to see that the ensemble method only performed the best on one out of eight datasets. Given that the motivation for applying the ensemble method is that it can take the advantages of each method with purpose of achieving better results. Therefore, one of our future work is to explore other ways to make our ensemble method more effective.

8. ACKNOWLEDGEMENTS

This research was supported by the NSF Grant 1432156 "Educational Data Mining for Individualized Instruction in STEM Learning Environments".

9. REFERENCES

[1] J. R. Anderson. *The architecture of cognition*. Cambridge, Mass. : Harvard University Press, 1983.

[2] F. R. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *NIPS*, pages 105–112, 2009.

[3] M. Chi, K. VanLehn, D. Litman, and P. Jordan. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 2011.

[4] N. D. Daw. Model-based reinforcement learning as cognitive search: neurocomputational theories. *Cognitive search: Evolution, algorithms and the brain*, pages 195–208, 2012.

[5] R. Gaudel and M. Sebag. Feature selection as a one-player game. In *ICML*, pages 359–366, 2010.

[6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[7] H. Hachiya and M. Sugiyama. Feature selection for reinforcement learning: Evaluating implicit state-reward dependency via conditional mutual information. In *Machine Learning and Knowledge Discovery in Databases*, pages 474–489. Springer, 2010.

[8] M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[9] P. W. Keller, S. Mannor, and D. Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 449–456. ACM, 2006.

[10] J. T. Kent. Information gain and a general measure of correlation. *Biometrika*, 70(1):163–173, 1983.

[11] J. Z. Kolter and A. Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 521–528. ACM, 2009.

[12] C. Lee and G. G. Lee. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information processing & management*, 42(1):155–165, 2006.

[13] L. Li, J. D. Williams, and S. Balakrishnan. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *INTERSPEECH*, pages 2475–2478, 2009.

[14] J. Morimoto, S.-H. Hyon, C. G. Atkeson, and G. Cheng. Low-dimensional feature extraction for humanoid locomotion using kernel dimension reduction. In *ICRA*, pages 2711–2716. IEEE, 2008.

[15] B. Mostafavi, G. Zhou, C. Lynch, M. Chi, and T. Barnes. Data-driven worked examples improve retention and completion in a logic tutor. In *Artificial Intelligence in Education*, pages 726–729. Springer, 2015.

[16] A. Newell. *Unified Theories of Cognition*. Harvard University Press; Reprint edition, 1994.

[17] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 1998.

[18] J. R. Tetreault and D. J. Litman. A reinforcement learning approach to evaluating state representations in spoken dialogue systems. *Speech Communication*, 50(8):683–696, 2008.

[19] K. VanLehn, P. W. Jordan, and D. J. Litman. Developing pedagogically effective tutorial dialogue tactics: experiments and a testbed. In *SLaTE*. Citeseer, 2007.

[20] R. Wright, S. Loscalzo, and L. Yu. Embedded incremental feature selection for reinforcement learning. Technical report, DTIC Document, 2012.

[21] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.

[22] M. F. Zibran. Chi-squared test of independence. *Department of Computer Science, University of Calgary, Alberta, Canada*, 2007.