Projection-based PILP: computational learning theory with empirical results

Hiroaki Watanabe and Stephen Muggleton

Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK

Abstract. Evaluations of advantages of Probabilistic Inductive Logic Programming (PILP) against ILP have not been conducted from a computational learning theory point of view. We propose a PILP framework, projection-based PILP, in which many-to-one projection functions are used to produce a "*lossy*" compression dataset from an ILP dataset. We present sample complexity results including conditions when projection-based PILP needs less examples than PAC. We experimentally confirm the theoretical bounds for the projection-based PILP in the Blackjack domain using *Cellist*, a system which machine learns Probabilistic Logic Automata. In our experiments projection-based PILP shows lower predictive error than the theoretical bounds and achieves substantially lower predictive error than ILP. To the authors' knowledge this is the first paper describing both a computer learning theory and related empirical results on an advantage of PILP against ILP.

1 Introduction

Probabilistic Inductive Logic Programming (PILP) [4] demonstrates a way to extend ILP towards relational Machine Learning (ML) under uncertainty. From an ILP perspective, the following question is still pertinent: (*Q*)*Does the additional representational power of probabilistic logics (p-logics) make logic-based ML harder?* We investigate this question in this paper by developing a computational learning theory that characterises PILP.

Our PILP framework, projection-based PILP, is illustrated in Fig. 1 in which given positive/negative examples for learning a Blackjack player model are projected onto a "*lossy*" probabilistic example (p-example). The projection f maps a number on a playing card (A,..,K) onto a new card ($C1$,..,$C4$) as defined in the figure. The projection is "*lossy*" since we lose the information about the original labels (pos/neg) as it is *draw* after the projections. To handle this uncertainty, we attach an estimated probability label, 0.5, to the projected example to express *the degree of positiveness*. In this setting, f and estimation errors of the probability labels could affect the sample complexities of ML from the projected p-examples. The structure of this paper is as follows. In Chapter 2, we provide theoretical results on sample complexities in our PILP framework. We compare our theoretical results with empirical results in the Blackjack domain in Chapter 3. Discussions conclude this paper in Chapter 4.

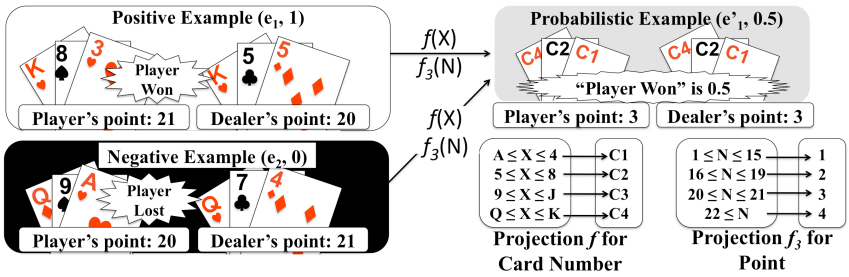

Fig. 1. "Lossy" Projection and Probabilistic Example

2 Projection-based PILP

Projection-based PILP can be achieved by a 2-steps approach: **[Step1]** *projecting given examples to p-examples* and **[Step2]** *learning hypotheses using the p-examples*. In **[Step1]**, Learner is expected to provide the following function.

Definition 1 (Projection Function f). *Assume both X and X' are non-empty sets. Let f be a surjective (or many-to-one) function from X to X'.*

An example of f can be found in Fig. 1. Let E be a set of N Boolean labelled given examples, $\langle (e_1, l_1), ..., (e_N, l_N) \rangle$ in which $l_i = 1$ for a positive example whereas $l_i = 0$ for a negative example. Each of the N examples is then projected via f onto m p-examples, $E' = \{(e'_1, \hat{p}_1), ..., (e'_m, \hat{p}_m)\}$, in which $\hat{p}_j$ is an estimated probability label for e'_j. In Fig. 1, both $(e_1, 1)$ and $(e_2, 0)$ are projected onto $(e'_1, 0.5)$. Let n_j be the number of Boolean labelled examples mapped onto e'_j. A lower bound of n_j for the estimation of a true probability p_j, $\hat{p}_j$, is obtained as follows. Our proof can be found in Appendix A.

Theorem 1. *For each $e'_j \in E'$, sample complexity for estimating p_j with error ϵ at confidence level $1 - \delta$ is $n_j > \frac{\pi(1-\delta)^2}{32\ \epsilon^2}$.*

For example, we obtain $n_j > 8.86$ when $\delta = 0.05$ and $\epsilon = 0.1$. In the case of $\delta = 0.05$ and $\epsilon = 0.05$, the lower bound is $n_j > 35.4$.

[Step1] is described in the following *Sequential Probability Label Estimation* algorithm. Intuitively, we continue both *samplings* and *projections* from E to E' until enough Boolean-labelled examples are mapped onto each p-examples. Note that the time bound T is required since it might take time to obtain n examples for all e'_j under some unrepresentative probability distribution over E. Regarding the step 4 in the following algorithm, the estimation error of e'_j is obtained as follow. Discussions about sequential sampling algorithms can be found in [7]. Our proof can be found in Appendix A.

Corollary 1 (Estimation Error ϵ_δ). *The estimation error of e'_j, $\epsilon_{\delta j}$, for a confidence level δ is $\epsilon_{\delta j} = \frac{\sqrt{2}\ erf^{-1}(1-\delta)\sqrt{\frac{1}{n_j - 1}\sum_{k=1}^{n_j}(l_k - \hat{p}_j)^2}}{\sqrt{n_j}}$ in which n_j is the number of Boolean labelled examples mapped onto e'_j and $erf(x)$ is the Gauss error function $erf(x) = \frac{2}{\pi}\int_0^x e^{-t^2} dt$.*

Sequential Probability Label Estimation

Input: A sequence of Boolean-labelled examples $e_i \in E, \langle (e_1, l_1), (e_2, l_2), ... \rangle$
Projection function f, confidence level δ, error level ϵ, time bound T

Output: Actual estimation error ϵ_δ
A set of m probabilistic examples $\{(e'_1, \hat{p}_1), ..., (e'_m, \hat{p}_m)\}$

1. Set $i = 1$, $j = 1$, $cnt_k = 0$, $t_k = 0$ $(1 \leq k \leq m)$, $X' = \emptyset$.
   Compute the sample complexity $n = \frac{\pi(1-\delta)^2}{32\ \epsilon^2}$
2. If $i \leq T$, take (e_i, l_i), otherwise go to Step 4.

3a. If $e'_j \notin X'$ such that $e'_j = f(e_i)$, set $cnt_j = 1$ and add (e'_j, cnt_j) to X'.
   Set $j = j + 1$. If $l_i = 1$, $t_j = t_j + 1$. If $cnt_j \leq n$ for all $(1 \leq j \leq m)$,
   set $i = i + 1$ and go to Step 2, otherwise go to Step 4.

3b. If $e'_j \in X'$, update cnt_j of $(e'_j, cnt_j) \in X'$ to $cnt_j = cnt_j + 1$. Set $j = j + 1$.
   If $l_i = 1$, $t_j = t_j + 1$. If $c_j \leq n$ for all $(1 \leq j \leq m)$, set $i = i + 1$ and
   go to Step 2, otherwise go to Step 4.

4. Compute the estimation error ϵ_j for each j $(1 \leq j \leq m)$.
5. Output the largest ϵ_j as ϵ_δ and $\{(e'_1, \hat{p}_1), ..., (e'_m, \hat{p}_m)\}$ such that $\hat{p}_j = t_j / cnt_j$.
6. Exit.

Next, **[Step1]** and **[Step2]** are linked by the estimation error ϵ_δ that is used as a constant in our complexity result in **[Step2]** as follows. Note that in our projection-based PILP, a hypothesis h returns a "degree of acceptance" $h(e'_j)$ in a probability value for e'_j.

Theorem 2 (Sample Complexity for Projection-based PILP). *Given p-examples with error ϵ_δ, for any ϵ_δ, ε', and δ' such that $0 \leq \varepsilon' + 2\epsilon_\delta \leq 1$, $0 \leq \delta \leq 1$ and $0 \leq \delta' \leq 1$, let m be the number of the p-examples sufficient for any ϵ_δ consistent learner to successfully learn any target concept in the hypothesis space H with true error ε' in confidence $(1-\delta')$. Then m is bounded as $m \geq \frac{ln|H| + ln\frac{1}{\delta'}}{\varepsilon' + 2\epsilon_\delta}$.*

Our proof is in Appendix A. We compare this upper bound with a lower bound on sample complexity of PAC learning to *clarify the conditions in which projection-based PILP needs less examples than PAC.* The lower sample bound of PAC is reported in [1] as follows. Consider any concept class C such that VC-dimension [2] $VC(C)$, any learner L, and any $0 < \varepsilon < 1/2$, and $0 < \delta < 1/100$. Then there exists a distribution D and target concept in C such that if L observes fewer examples than $max\left[\frac{1}{\varepsilon}\ ln\left(\frac{1}{\delta}\right),\ \frac{VC(C)-1}{32\varepsilon}\right]$ then with probability at least δ, L outputs a hypothesis h having $error_D(h) > \varepsilon$. If H contains C and $|H|$ is finite, the above formula can be written [2] as $max\left[\frac{1}{\varepsilon}\ ln\left(\frac{1}{\delta}\right),\ \frac{log_2|H|-1}{32\varepsilon}\right]$.

Theorem 3. *Consider any PAC learner L_{PAC} with a hypothesis space $|H|$, any $0 < \varepsilon < 1/2$, $0 < \delta < 1/100$, and a distribution over examples D_{PAC}. Then there exists a distribution D_{PAC}, target concept in H, such that if a projection-based learner L can design projection function that results the ε-exhausted hypothesis space H' such that $\frac{|H|}{|H'|} > \frac{2}{\delta'}$ when $\frac{1}{\varepsilon}\ ln\left(\frac{1}{\delta}\right) < \frac{log_2|H|-1}{32\varepsilon}$ or $|H'| < \left(\frac{1}{\delta'}\right)^{\frac{2\epsilon_\delta}{\varepsilon}}$ when $\frac{1}{\varepsilon}\ ln\left(\frac{1}{\delta}\right) \geq \frac{log_2|H|-1}{32\varepsilon}$ then with probability at least δ', L outputs a hypothesis $h \in H'$ having $error_D(h) < \varepsilon + 2\epsilon_\delta$ with less examples than any PAC learner.*

Proof is in Appendix A. For example, (a) if the projected hypothesis space H' is 20 times smaller than the original hypothesis space H, (b) $|H'|$ can achieve $\delta' = 0.1$, and (c) the size of the original hypothesis is $2^{74} < |H|$, projection-based PILP has an advantage in terms of the number of examples for $\epsilon_\delta = \varepsilon = \delta = 0.1$.

3 Experiments in Blackjack Domain

Material: We experimentally compare ILP and projection-based PILP on the Blackjack domain. Blackjack is a card game between the *player* and the *dealer*. We adopt the standard face-up game rule. We assume that the *player* and the *dealer* have the following strategies to play: the *player* deals only when the sum is less than 16 whereas the *dealer* is less than 19. Based on this strategies, we implemented a Blackjack simulator that estimate (a) the probability of the player's win is 51.3% and (b)the average number of cards drawn in a positive example is 5.97. With these numbers, we estimate that there could exist 162175 ($= 52 \times 51 \times 50 \times 49 \times 48 \times 47 \times 0.513$) positive (*player's won*) examples.

We explore three different representations. First representation uses the original number of cards and the score of the hands. In the second and the third representations, a *number* printed on a card is projected by f defined in Fig. 1 and the *points* are projected by f_2 and f_3 respectively as follows. Projection f_2 maps from the points N to $\lfloor N \rfloor$ whereas f_3 maps (a)$1 \leq N \leq 15$ to 1, (b)$16 \leq N \leq 19$ to 2, (c)$20 \leq N \leq 21$ to 3, and (d)$22 \leq N$ to 4.

Regarding the creation of p-examples, the 52 playing cards are randomly shuffled and a sequence of plays is generated based on the strategies for each game. The generated sequence is stored in the multi-set E_0 as a non-projected example. We re-shuffled the 52 cards and generate sequences of play until we obtain 10000 examples in E_0. Then E_0 is taken by `Sequential Probability Label Estimation` algorithm. E_1 is created by combining f and f_1 whereas E_2 is by f and f_2. We separated E_0, E_1, and E_2 into (a) training data and (b) test data. Using the training data, we generated 5 sets of training examples with 5 different sizes (10, 20, 30, 40, and 50). The size of the test data is 100 in our experiment.

Method: We learn the *player*'s strategy from the observations of their plays as a Probabilistic Logic Automaton (PLA) using a Machine Learning System *Cellist* [6]. *Cellist* supports two-steps model construction: structure learning of the PLA followed by EM-based parameter estimation. Our structure learning algorithm consists of (a) state mergings for topology learning of Automata and (b) Plotkin's lgg [3] motivated most general specialisation of existentially quantified conjunctions of literals. Regarding inference in PLAs, the probability of the given input sequence being accepted by a PLA model is computed by the Forward Algorithm that is a logical modification of HMM's forward inference algorithm [5]. Given a p-example, the gap between the acceptance-probability of the given example and the probability label attached to the p-example is treated as a *predictive error*.

Results: Our empirical results are shown in Fig.2 and Fig.3. Fig.2 shows that the ILP-based approach results hypotheses in large errors. As we see in Fig.2 and

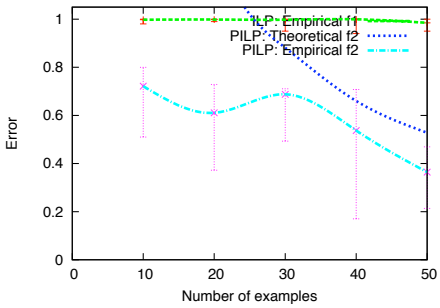

Fig. 2. ILP vs f_1-based PILP

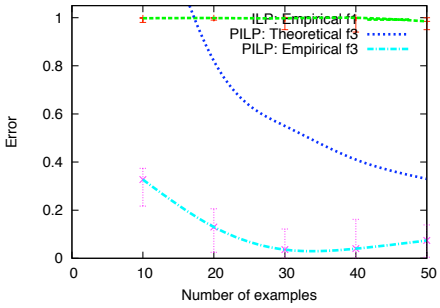

Fig. 3. ILP vs f_2-based PILP

demerit

Fig.3, projection-based PILP shows lower error compared with the ILP-based approach although the projection f_2 resulted in hypotheses with lower error than the hypotheses via projection f_1 for all the sample sizes.

4 Discussions and Conclusions

Theorem 3 suggests the answer for (Q)*Does the additional representational power of probabilistic logics make logic-based ML harder?* is "*No, not always in terms of the number of examples*" in PILP. In our approach, f causes ϵ_δ for each p-example, however, ML in the projected knowledge representations overcomes this demerit in the Blackjack domain. One possible explanation in PLA is as follows. The game point in the non-projected representation increases 1 point each whereas f_1 and f_2 result in *coarser* representations. In Plotkin's lgg, "*finer*" logical ground terms are more likely to be replaced by first-order variables which could cause over-fittings.

Regarding the comparison between f_1 and f_2, the projection f_2 encodes more information about the strategies and rules in the form of the thresholds 16, 19, and 21. Since f_2 shows better predictive accuracy, a *quality* of projection functions could affect the result of the learning. Our projection function is flexible enough to encode more complex functions unless it is a many-to-one function. We believe that the projection-based PILP has potential for applying PILP in large numerical datasets with relations effectively.

Appendix A: Proof of Theorem 1: Central Limit Theorem states $\lim_{n_j\to\infty} Pr[\frac{X-n_jp}{\sqrt{n_jp(1-p)}} \leq z] = \Phi(z)$ where $\Phi(z)$ is the cumulative distribution function of the standard normal distribution of N(0,1). This leads $Pr[\hat{p} > p+\epsilon] = Pr[\hat{p} < p-\epsilon] \approx 1-\Phi(\frac{\epsilon\sqrt{n_j}}{p(1-p)})$ since $\Phi(-z) = 1-\Phi(z)$ and $\Phi(-\frac{\epsilon\sqrt{n_j}}{p(1-p)}) = Pr(\frac{X-n_jp}{\sqrt{n_jp(1-p)}} \geq \frac{\epsilon\sqrt{n_j}}{\sqrt{p(1-p)}}) = Pr(X/n_j - p \geq \epsilon)$. Because of $Pr[\hat{p} > p+\epsilon] + Pr[\hat{p} < p-\epsilon] < \delta$, we obtain $\Phi(\frac{\epsilon\sqrt{n_j}}{p(1-p)}) > 1-\frac{1}{2}\delta$. From this formula, we obtain $n_j > \frac{\pi(1-\delta)^2}{2\epsilon^2}\{p(1-p)\}^2$ since (1) $\Phi(z)$ can be expressed as: $\Phi(z) = \frac{1}{2}[1+\mathtt{erf}(\frac{z}{\sqrt{2}})]$ where $\mathtt{erf}$ is a special function called the error function: $\mathtt{erf}(z) = \frac{1}{\sqrt{\pi}}\sum_{n_j=0}^{\infty}\frac{(-1)^{n_j}\ z^{2n_j+1}}{n_j!(2n_j+1)}$ and (2) the Maclaurin series of $\mathtt{erf}^{-1}(z)$ is $\mathtt{erf}^{-1}(z) = \sqrt{\pi}(\frac{1}{2}z + \frac{1}{24}\pi z^3 + ...)$. Finally $n_j > \frac{\pi(1-\delta)^2}{2\epsilon^2}\{p(1-p)\}^2 > \frac{\pi(1-\delta)^2}{2\epsilon^2}$. □

Proof of Corollary 1: For $n_j > 30$, the Central Limit Theorem guarantees the error $\epsilon_{\delta j}$ forms a Normal distribution and $\epsilon_{\delta j}$ can be calculated by using the estimated

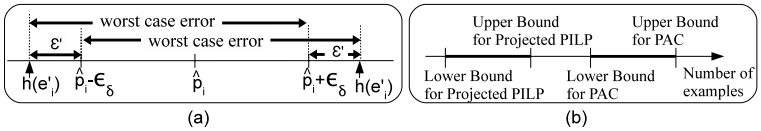

Fig. 4. (a)Worst Case Error, (b)Relation between lower bound of PAC and upper bound of projection-based PILP

variance of population, $\hat{\sigma} = s/\sqrt{n_j}$ where s^2 is the sample variance. $\epsilon_{\delta j}$ is defined as a function of δ, $\epsilon_\delta = z\hat{\sigma}$, where z is a function of δ such that $z = \sqrt{2}\ erf^{-1}(1-\delta)$. Then for estimation of e'_j, the sample variance s^2 can be written as $s_j^2 = \frac{1}{n_j - 1}\sum_{k=1}^{n_j}(l_k - \hat{p}_j)^2$. This leads the corollary. □

Proof of Theorem 2: Let us introduce a ϵ_δ *consistent hypothesis* h such that $\hat{p}_j - \epsilon_\delta \leq h(e'_j) \leq \hat{p}_j + \epsilon_\delta$ holds for every $(e'_j, \hat{p}_j) \in E'$. Then an inconsistent hypothesis h has a worst-case error, $\varepsilon' + 2\epsilon_\delta$ as shown in Fig. 4(a). Let $h_1, ..., h_k$ be all the hypotheses with errors greater than ε'. The probability that this hypothesis will be ϵ_δ consistent with m independently drawn p-examples is at most $\{1 - (\varepsilon' + 2\epsilon_\delta)\}^m$. The probability that at least one of k will be ϵ_δ consistent with all m probabilistic training examples is at most $k\{1 - (\varepsilon' + 2\epsilon_\delta)\}^m$. Since $k \leq |H|$, this is at most $|H|\{1 - (\varepsilon' + 2\epsilon_\delta)\}^m$. Finally, we use a general inequality stating: $(1 - x) \leq e^{-x}$ if $0 \leq x \leq 1$. For $0 \leq \varepsilon' + 2\epsilon_\delta \leq 1$, $|H|\{1 - (\varepsilon' + 2\epsilon_\delta)\}^m \leq |H|e^{-(\varepsilon' + 2\epsilon_\delta)m}$. By considering the upper bound δ' of this error, we get $|H|e^{-(\varepsilon' + 2\epsilon_\delta)m} \leq \delta'$. This leads $m \geq \frac{ln|H| + ln\frac{1}{\delta'}}{\varepsilon' + 2\epsilon_\delta}$. □

Proof of Theorem 3: We consider conditions when our PILP has smaller sample complexity than PAC. Fig. 4 (b) shows the case in which a number of required examples for our PILP is always smaller than a number of examples for PAC. This figure shows the relation between the upper sample complexity of our PILP in Theorem 2 should be smaller than the lower bound of sample complexity of PAC shown in [1]. Regarding the first case in which $\frac{log_2|H| - 1}{32\varepsilon} > \frac{ln|H'| + ln\frac{1}{\delta'}}{\varepsilon + 2\epsilon_\delta}$, this leads $\frac{ln(|H|/2)}{ln(|H'|/\delta')} > \frac{32\varepsilon ln2}{\varepsilon + 2\epsilon_\delta}$. Since $\frac{32\varepsilon ln2}{\varepsilon + 2\epsilon_\delta} > 1$ for $0 < \varepsilon < 1$ and $0 < \epsilon_\delta < 1$, we obtain $\frac{ln(|H|/2)}{ln(|H'|/\delta')} > 1$. This leads $|H|/2 > |H'|/\delta'$. Therefore we get $\frac{|H|}{|H'|} > \frac{2}{\delta'}$. Let us consider the second case in which $\frac{1}{\varepsilon}\ ln\left(\frac{1}{\delta}\right)$ is greater than (1). $\frac{1}{\varepsilon}\ ln\left(\frac{1}{\delta}\right) > \frac{ln|H'| + ln\frac{1}{\delta'}}{\varepsilon + 2\epsilon_\delta}$ If we consider $\delta = \delta'$, we obtain $|H'| < \left(\frac{1}{\delta}\right)^{\frac{2\epsilon_\delta}{\varepsilon}}$.

References

1. A Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Informution and computation*, 82:247–261, 1989.
2. T.M. Mitchell. Machine learning. *McGraw-Hill*, 1997.
3. G. Plotkin. A note on inductive genralization. *Machine Intelligence*, 5:153–163, 1970.
4. Luc De Raedt and Kristian Kersting. Probabilistic inductive logic programming. *ALT-2004*, LNCS 3244:19–36, 2004.
5. Stuart J. Russell and Peter Norvig. Artifical intelligence: A modern approach. *Prentice Hall*, 2nd edition, 2003.
6. Hiroaki Watanabe and Stephen Muggleton. Can ILP be applied to large datasets? In *LNCS 5989 (ILP 2009)*, pages 249–256, 2009.
7. Osamu Watanabe. Sequential sampling techniques for algorithmic learning theory. *Theoretical Computer Science*, 2348(1,2):3–14, 2005.