On Efficiency of Learning Under Privileged Information

Ilia Nouretdinov I.R.NOURETDINOV@RHUL.AC.UK
Centre of Machine Learning, Royal Holloway University of London, Egham Hill, Egham, TW20OEX, United Kingdom

Editor: Ulf Johansson, Henrik Boström, Khuong An Nguyen, Zhiyuan Luo and Lars Carlsson

Abstract

The paradigm of Learning Under Privileged Information (LUPI) was used in various practical applications, including its combination with Conformal Prediction (CP) framework. In this note, we discuss possible sources and limitations of its efficiency. We try to argue that accuracy improvement coming from using privileged information is not occasional. For this goal, we consider some minimalistic models of LUPI where the contribution of the privileged information appears in its noise-free essence. Then, we discuss connection of LUPI paradigm and CP framework in relation with the models.

Keywords: Conformal prediction, privileged information, efficiency.

1. Introduction

In the standard machine learning approach, an instance consists of a feature vector and a label. There exists a training set of examples, where both of them are available, and new examples where only the feature vectors are known and the labels have to be predicted.

Learning Under Privileged Information (LUPI) paradigm of machine learning was initially presented by Vapnik and Vashist (1). In this view, an instance is divided into three parts instead of two. An object consists of a feature vector, a vector of privileged information, and a label. The role of the feature vector is the same as in the standard paradigm: it is available for all training and all the new examples. Unlike that, privileged information (PI) is available only for training examples. The same can be said about the availability of labeled examples. However, for the new examples, the prediction of the privileged features is not required. In another interpretation, Privileged Information means having missing values in some features of *testing* examples, not training ones.

The prediction task in LUPI paradigm is the same as in the usual one: to predict the labels for the new examples. Therefore, their results are comparable to each other. PI is additional and its usage is optional. As a baseline, it is always possible just to ignore all PI and to make the prediction in a usual way by one of the machine learning methods.

It may happen that ignorance of additional information may lead to a loss of efficiency (accuracy) of the predictions. Although many published applications demonstrate a positive contribution of PI to accuracy, using LUPI paradigm may be useless or even harmful if PI is noisy or redundant. This creates a difficulty in the analysis of real data results because any real additional feature contains elements of both real information and noise. Therefore, in this work, we take a step back from practical applications to some minimalist artificial models. This is needed to study the effect of PI clarified from noise and side circumstances. The most important point is to state the dependency on the training set size as the most principal dimension of the problem.

x	x^*	y	$P\{(x, x^*, y)\}$
0	1	odd	0.4
0	2	even	0.2
0	3	odd	0.2
0	4	even	0.2

Table 1: Distribution (minimalist example).

We will also pay attention to the question of a connection between LUPI paradigm and CP framework. The practical work (3) has shown the possibility of applications, and we try to explain this effect in connection to artificial modeling.

2. Modelling of LUPI

2.1. General setting

In this work, we follow notation from (1) x, x^*, y for the feature vector, the privileged information, and the label, respectively.

We assume that the data follows i.i.d. (power) assumption that all the triples (x_i, x_i^*, y_i) are generated independently by the same distribution P, and on n-th step y_{n+1} has to be predicted after training on the base of the full triples (x_i, x_i^*, y_i) for $i = 1, \dots, n$ and the new example's feature vector x_{n+1}.

2.2. A minimalist model

For our first example, we consider the following LUPI data generating model with the distribution P on triples as shown in Tab. 1.

This model is 'minimalist', we try to make a non-trivial LUPI model as simple as possible.

It can be said the goal of machine learning is approximation to the true density $P\{y|x\}$ for any example x. As far as the training data grows, training examples with x_i far from x_{n+1} become less important for estimation of $P\{y_{n+1}|x_{n+1}\}$. Therefore, in our first example, we assume that x is a constant value. It can be interpreted as ignoring all the training examples which are not close enough to x_{n+1}.

Next, the simplest machine learning prediction task is binary classification, therefore we assume that the range of y is two values.

We also need an example of privileged information. We assume that for each value in the range of y, there are two different possible values of x^*. So, there are four combinations (PI,label) pairs (x^*, y). If the numeration of the pairs (x^*, y) matches the numeration of x^*, then y can be omitted when talking about the pairs.

We assume that $P\{x^*, y|x\}$ is non-zero for any x^*. This models the realistic situation when x lies in the area where none of the classes dominates strongly, all of them are possible.

Remind that, regardless whether PI is used or not, the final machine learning task is to answer the question: "is y_{n+1} even or odd"? The best answer is "odd" because the true probability of this event is $P\{y = odd|x\} = 0.6$ that can not be improved. In any case this answer has a larger chance to be correct than "even". So, we will measure the accuracy of a prediction algorithm by its chance of giving the output "odd" after training.

As the prediction algorithm, we apply the simplest "all-neighbours" rule explained below. It makes the prediction by majority of votes, in case of tie makes the choice randomly.

- **Without privileged information:**

  training data: $(y_1, \ldots, y_n)$;

  decision rule: choose between "even", "odd" by majority of votes;

  breaking ties (same number for "even" and for "odd"): fairly randomised;

  examples:

  (even,odd,even)→even;

  (even,odd,odd)→odd;

  (even,odd,odd,even)→ $\frac{1}{2}$ odd, $\frac{1}{2}$ even.

- **With privileged information:**

  training data: $(x_1^*, \ldots, x_n^*)$;

  decision rule: choose between 1,2,3,4 by the highest number of votes, then convert to "odd" or "even";

  breaking ties (more than one winner): equally randomised between the winners;

  examples:

  (1,2,1,3,4)→ 1 → odd;

  (1,2,1,2,4)→ $\frac{1}{2}$ 1, $\frac{1}{2}$ 2 → $\frac{1}{2}$ odd, $\frac{1}{2}$ even;

  (1,2,3,1,2,3,4)→ $\frac{1}{3}$ 1, $\frac{1}{3}$ 2, $\frac{1}{3}$ 3 → $\frac{2}{3}$ odd, $\frac{1}{3}$ even.

Definitely, the final answers will match in majority of cases. But there may be some exceptions such as: (1,1,1,2,2,4,4)=(odd,odd,odd,even,even,even,even). This is where using PI improves the quality by replacing the answer "even" with "1=odd".

Let us show how using PI increases the probability to guess the correct ("odd") answer. The overall dependence on the training size is shown in Tab. 3 and Fig. 1. Note that in this artificial example, the probabilities and the corresponding gains can be calculated exactly. The examples of how this is done are given in Tab. 2 for $n = 4, 5$, for other values they are analogous. Please that there is some difference between even and odd values of n due to combinatorial causes, this is why a tooth-like pattern is visible on the gain plot.

We can make the following observations.

1. Starting from $n = 10$, PI always makes a positive contribution.

2. Starting from $n = 40$, the contribution of Pi decreases although remains positive.

3. There is a permanent difference between odd and even values of n

$y_1 \dots y_n$	prob. (with permutations)	non-PI	PI	PI gain
1124	12*0.4*0.4*0.2*0.2=0.0768	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	odd	+1/2*0.0768
2213	12*0.4*0.2*0.2*0.2=0.0384	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	even	-1/2*0.0384
3324	12*0.2*0.2*0.2*0.2=0.0192	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	odd	+1/2*0.0192
4413	12*0.4*0.2*0.2*0.2=0.0384	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	even	-1/2*0.0384
Total				0.0096

$y_1 \dots y_n$	prob. (with permutations)	non-PI	PI	PI gain
11223	30*0.4*0.4*0.2*0.2*0.2=0.0384	odd	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	-1/2*0.0384
11224	30*0.4*0.4*0.2*0.2*0.2=0.0384	even	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	+1/2*0.0384
11442	30*0.4*0.4*0.2*0.2*0.2=0.0384	even	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	+1/2*0.0384
11443	30*0.4*0.4*0.2*0.2*0.2=0.0384	odd	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	-1/2*0.0384
22331	30*0.2*0.2*0.2*0.2*0.4=0.0192	odd	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	-1/2*0.0192
22334	30*0.2*0.2*0.2*0.2*0.2=0.0096	even	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	+1/2*0.0096
33441	30*0.2*0.2*0.2*0.2*0.4=0.0192	odd	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	-1/2*0.0192
33442	30*0.2*0.2*0.2*0.2*0.2=0.0096	even	$\frac{1}{2}$ odd, $\frac{1}{2}$ even	+1/2*0.0096
Total				-0.0096

Table 2: Calculation of the gain for $n = 4, 5$ (minimalist example). Only the data sequences with different non-PI and PI results are included, up to permutations.

Training set size	Expected gain of LUPI
1	0
2	0
3	-0.016
4	0.0096
5	-0.0096
6	0.00576
7	-0.00448
8	0.009856
9	-0.00103936
10	0.0111176
11	0.00314446
12	0.0139637
13	0.00627315
14	0.0158457
15	0.00946521
16	0.0181392
17	0.0120808
18	0.0198352
19	0.014547
20	0.0216421

Table 3: Dependence on the training set size (minimalist example).

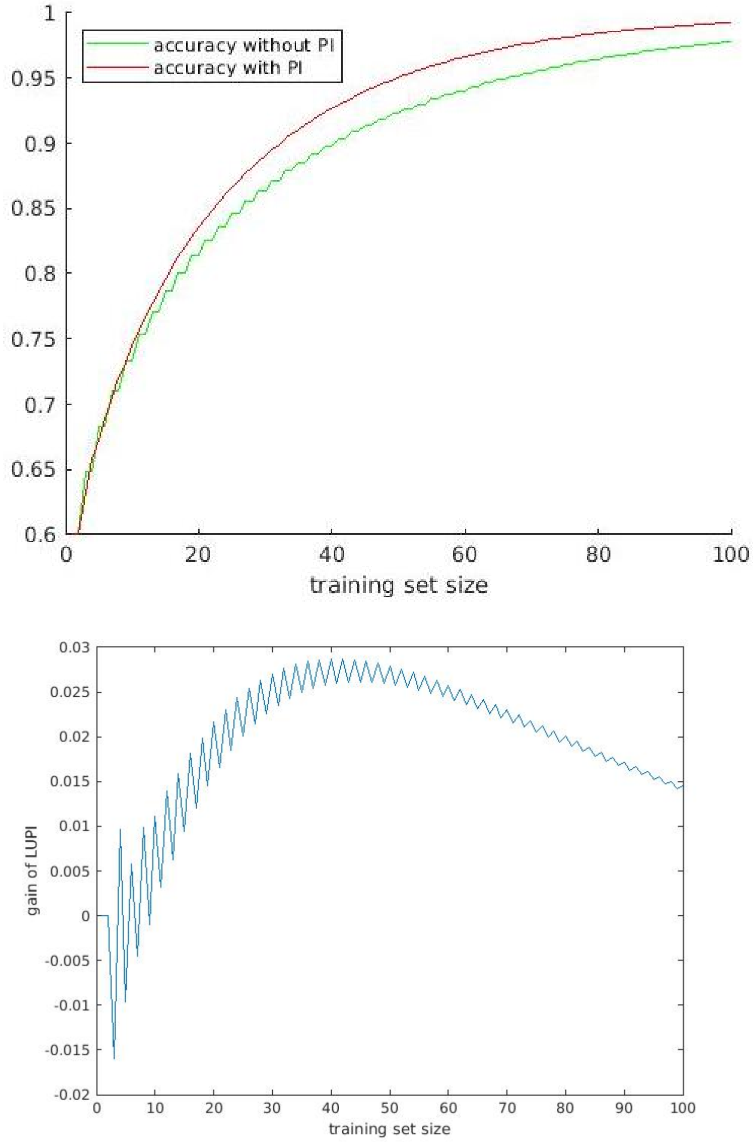

Figure 1: Dependence on the training set size (minimalist example): comparison and difference of the accuracy.

x	x^*	y	$P\{(x, x^*, y)\}$
0	1	odd	0.5
0	2	even	0.2
0	3	odd	0.1
0	4	even	0.2

Table 4: Distribution (imbalanced example).

Observation 3 is likely to be related to combinatorial effects within the concrete experimental setting. Observations 1–2 are more principal and can be preliminary explained. In the initial period ($n < 10$ in this example), PI does not help in classification because it split the data into very small classes, and this prevents any essential analysis. Later (starting from $n > 10$), PI becomes useful by giving the algorithm a 'hint' about the data structure. However, asymptotically the difference between learning with/without PI gradually becomes negligible because the learning algorithm discovers the correct data structure anyway as far as the training set becomes more and more representative. This is why the contribution is permanently decreasing after its peak is reached at $n = 40$.

To sum, we can make a hypothesis that involving PI is useless in the very early stage, not essential asymptotically, but it can help to accelerate the learning in the middle stage.

2.3. An imbalanced model

A modification of the previous example with probabilities shown in Tab.4. The dependence on the training size is shown in Tab. 5 and Fig. 2.

At the best point, the gain of using PI in accuracy reaches the level of 10%. This happens because the role of PI is concrete enough: it splits each of the classes into two clear sub-classes (clusters).

So, the preliminary conclusions are the following. PI is highly useful if it prevents an attempt to make a strong union between heterogeneous small classes which are labeled the same but do not have much in common with each other.

3. Connection with conformal prediction

3.1. Using LUPI with CP

The conformal Prediction (CP) framework for LUPI was earlier developed in (3) with application to the medical area (the symptoms as basic and privileged features). The remainder of its general scheme is shown in Alg. 1. Is to assign p-values to complex hypotheses about the pairs (x^*, y) unlike just y in the standard conformal prediction, and then calculation $p(y)$ as $\max_{x^*} p(x^*, y)$. So LUPI+CP combined approach made gain from the cases when $p(y)$ is large while for any complement x^*, $p(x^*, y)$ is small.

This has another support in the notion of *credibility* known from CP theory (2). The credibility is defined as $\max_y p(y)$, so it can be interpreted as using the original label in the role of x^*, at the same time leaving the label place y trivial. So, by analogy to LUPI, the calculation of credibility can be called Anomaly Detection Under Privileged Information (ADUPI).

Training set size	Expected gain of LUPI
1	0
2	0
3	0.008
4	0.0384
5	0.0264
6	0.05256
7	0.04928
8	0.0681856
9	0.0629686
10	0.0803262
11	0.0754191
12	0.0890781
13	0.0841907
14	0.0958017
15	0.0906497
16	0.100385
17	0.0951047
18	0.103262
19	0.0979449
20	0.104824

Table 5: Dependence on the training set size (imbalanced example).

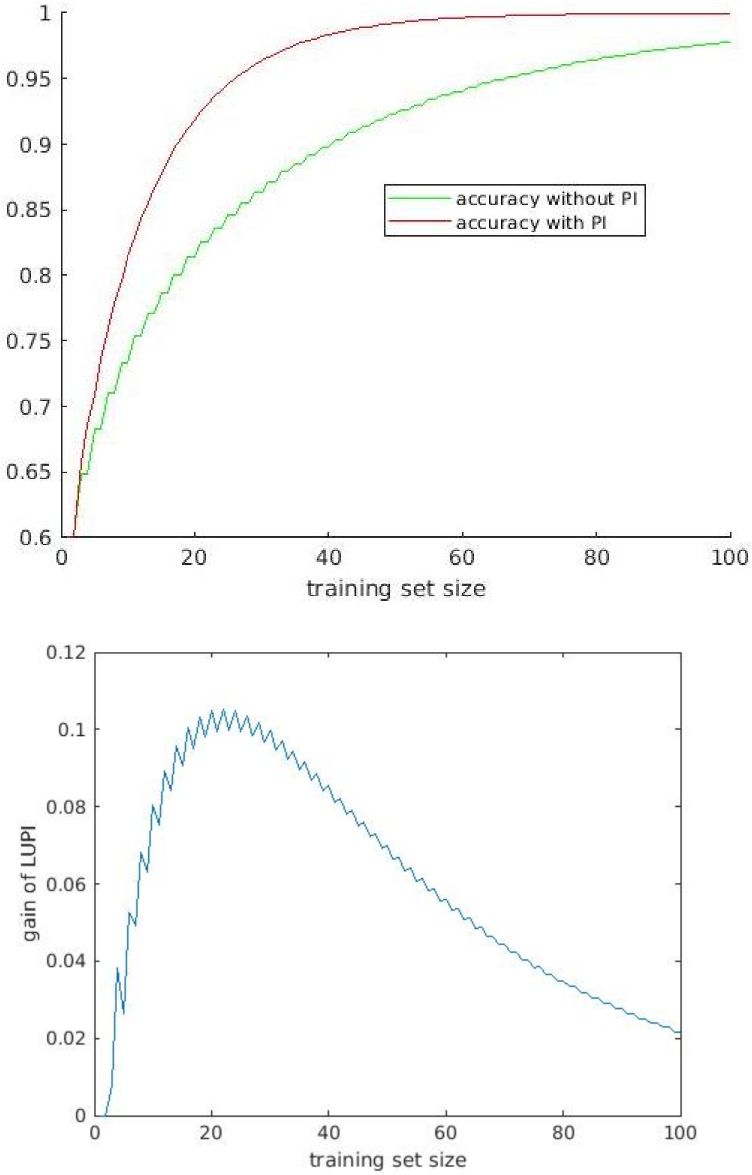

Figure 2: Dependence on the training set size (imbalanced example).

Algorithm 1 Conformal prediction scheme with/without PI

```
INPUT: training data (triples) $(x_1, x_1^*, y_1), \dots, (x_n, x_n^*, y_n) \in X \times X^* \times Y$
INPUT: testing example $x_{n+1}$
INPUT: conformity score function $A : (z, Z) \to [-\infty, +\infty]$ where $z$ is triple of type
$(x, x^*, y)$, $Z$ is a bag of triples.
INPUT: using PI? (yes/no)
if PI=no then
  set $X^* := \{0\}$ and all $x_i^* := 0$
end if
for $(x^*, y) \in X^* \times Y$ do
  $x_{n+1}^* := x^*$
  $y_{n+1} := y$
  for i:=1,. . . ,n+1 do
    $\alpha_i := A\Big((x_i, x_i^*, y_i), \{(x_j, x_j^*, y_j) | j = 1, \dots, j-1, j+1, \dots, n+1\}\Big)$
  end for
  generate $\theta$ with uniform distribution on [0,1]
  $p(x^*, y) = \frac{|\{i:\alpha_i<\alpha_{n+1}\}|+\theta|\{i:\alpha_i=\alpha_{n+1}\}|}{n+1}$
end for
```

3.2. Minimalist models

To have an understanding, let us also go through minimalist models, and them in terms of CP. In that minimalist examples, the conformity score is also defined simply as: 'the proportion of examples having the same extended label'. Here by the extended label we mean its extension (x^*, y) with privileged info if PI is included in training.

The results are shown on Fig. 3,4. Possible criteria of efficiency applicable to Conformal Prediction are discussed in (4). However, here for simplicity and comparability, we just measure the accuracy of the output chosen by the highest p-value. This way, in the non-PI setting we compare $p(even)$ to $p(odd)$ to get the answer 'odd or even?', while with PI the choice is done by comparing between $p(1), p(2), p(3), p(4)$. If the highest of them is $p(1)$ or $p(3)$, the output is 'odd', otherwise it is 'even'.

The figures are similar to the examples produced earlier beyond CP paradigm but in terms of conformal prediction, LUPI has a natural interpretation, in terms of hypothesis testing. A source of LUPI efficiency is that a complex hypothesis sometimes can be discarded more efficiently after splitting it into partial hypotheses.

3.3. A more complex example

As a more complex illustration, we present another toy example of LUPI approach, within CP framework. It is an example when each of two classes is a mixture of two normal distributions, and PI shows how to make un-mixing. The models are described in Tab. 6. They are analogous to two examples described earlier, the simple and the imbalanced ones.

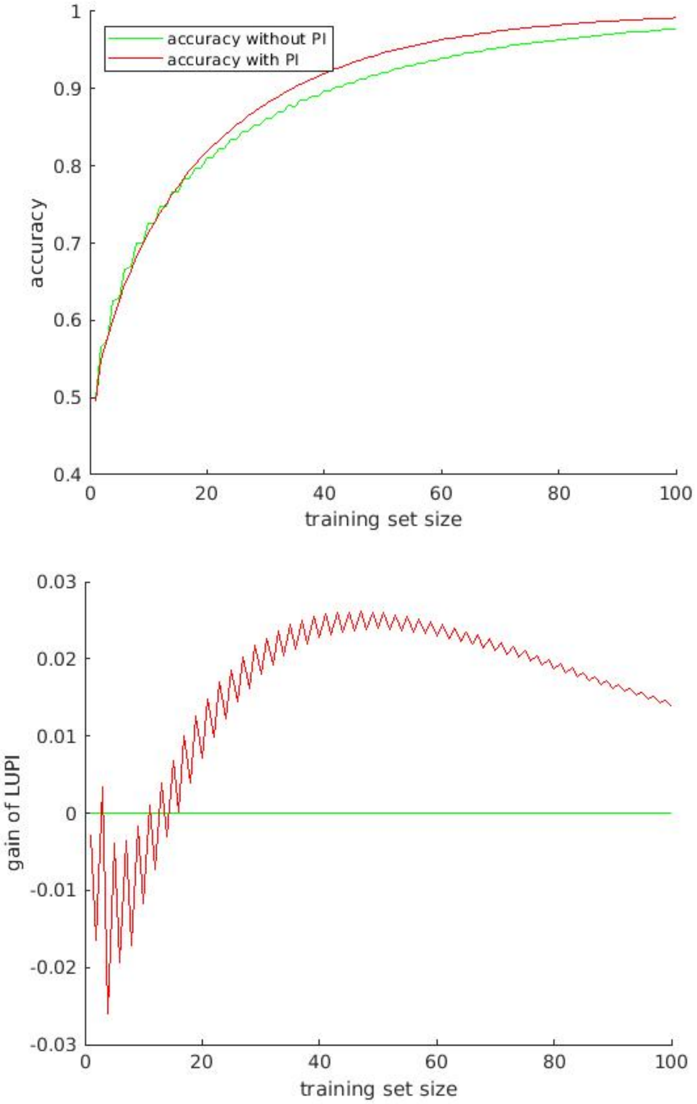

Figure 3: Dependence on the training set size (conformal version of the minimalist example).

$P\{x\|(x^*,y)\}$	x^*	y	$P\{(x^*,y)\}$
$N(1,1)$	1	odd	0.4
$N(2,1)$	2	even	0.2
$N(3,1)$	3	odd	0.2
$N(4,1)$	4	even	0.2

$P\{x\|(x^*,y)\}$	x^*	y	$P\{(x^*,y)\}$
$N(1,1)$	1	odd	0.5
$N(2,1)$	2	even	0.2
$N(3,1)$	3	odd	0.1
$N(4,1)$	4	even	0.2

Table 6: Distribution (conformal examples 1,2).

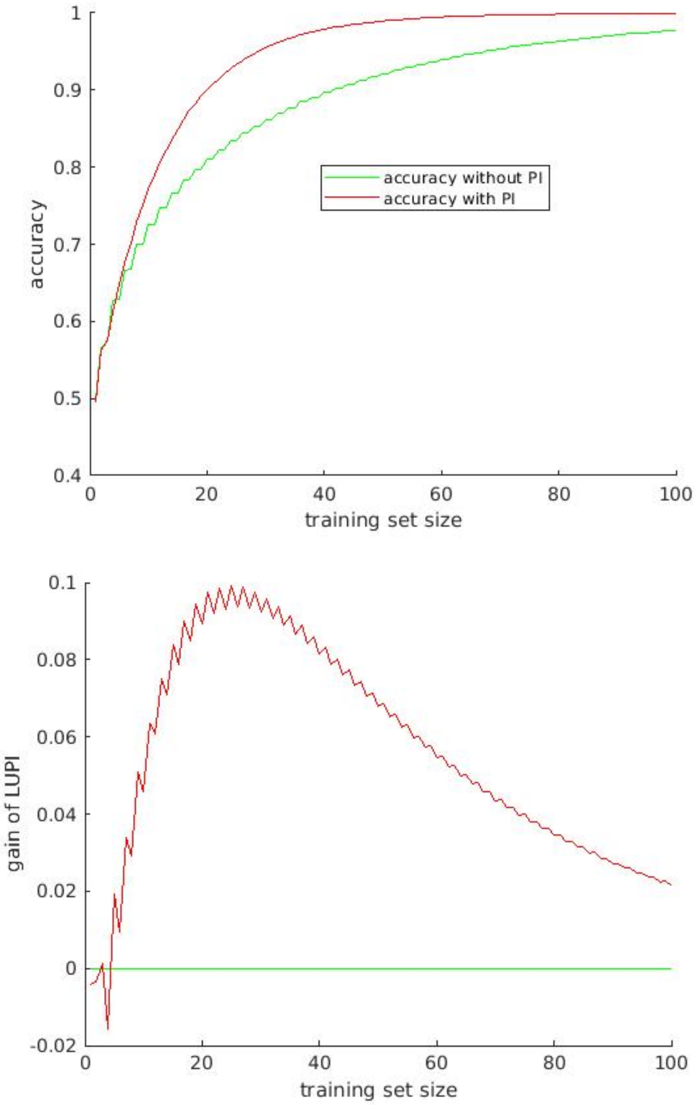

Figure 4: Dependence on the training set size (conformal version of the imbalanced example).

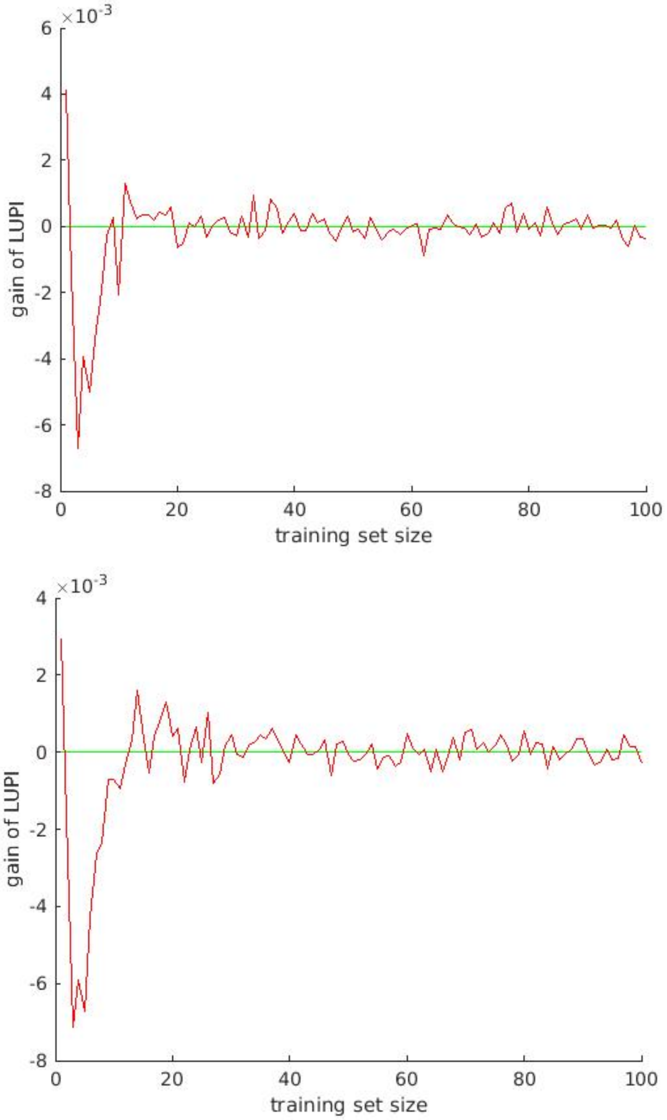

Figure 5: Dependence on the training set size (conformal examples 1,2).

As the conformity measure we use a simple one, often used in using CP framework for Nearest Neighbours basic method.

$$\frac{\textit{distance to the nearest neighbour with another label}}{\textit{distance to the nearest neighbour with the same label}}.$$

As before, the label means either y or (x^*, y) (the extended label), if PI is ignored or used respectively.

In this task, the expected gain is estimated empirically, by making a random sample of 100,000 data sequences. The results for different sizes of training set are shown in Fig. 5.

Surely, the contribution is smaller in the figures and less stable than the purely deterministic and featureless minimalist models demonstrated before. This is explainable by the fact that the gain of PI is local, related only to the areas where the distributions overlap essentially, so that the difference between PI and non-PI calculation may appear.

However, it can be seen that the shape of Fig. 5 is similar to that figures. Again, we have an intermediate stage (n between 15 and 25) where acceleration of the learning can be visible. So the principal conclusions are supported by this model as well.

4. Conclusion

In this work, we analyzed the efficiency LUPI learning paradigm by modeling it in simplistic ways. We supported our initial idea that the efficiency should be studied dynamically, in its dependence on the size of the training set.

Our observations lead to the following hypothesis about the dynamics of the impact of PI can be roughly divided into three stages, in dependence on the training set size.

1. Very small size – the gain from using PI is negative, as PI overloads the learning algorithm.
2. Medium size – the gain is positive, as PI accelerates the learning.
3. Large size – the gain is still positive but tends to decrease, as learning without PI also becomes efficient.

This report is just the beginning of the work in progress, and these conclusions are preliminary and due to further check and analysis. As far as the models become more complex, more limitations on the applicability of LUPI will be found. Even at this level, we have to note: if the noise in PI is high, then the pattern may be weaker than this one. Especially, if PI is completely useless then there is hardly any chance for the second ('Medium size') to appear.

One important direction of future work is the validation with real-world data sets. Here we have support from previous experience such as a combination of LUPI paradigm with CP framework applied to medical diagnostic in (3) where the contribution of PI to the efficiency was positive in the given setting. However, it also may gain from a dynamic investigation for different sizes of the training set. In this work, we also left aside a comparison of conformal predictions in terms of their proper criteria of efficiency observed is (4). This should be done in future investigations as well.

Acknowledgements

I am grateful to Alex Gammerman and Vladimir Vovk for the motivation and discussions of this work.

References

[1] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. Neural Networks, 22(5-6):544–557, 2009.

[2] Vladimir Vovk, Alex Gammerman, and Glenn Shafer. 2005. Algorithmic Learning in a Random World. Springer-Verlag, Berlin, Heidelberg.

[3] M. Yang, I. Nouretdinov, Z. Luo. Learning by Conformal Predictors with Additional Information. In The 9th Artificial Intelligence Applications and Innovations Conference (AIAI): 2nd Workshop on Conformal Prediction and its Applications. 2013. p. 394–400. (IFIP Advances in Information and Communication Technology).

[4] Vladimir Vovk, Ilia Nouretdinov, Valentina Fedorova, Ivan Petej, Alex Gammerman. Criteria of efficiency for conformal prediction. https://doi.org/10.48550/arXiv.1603.04416