Proceedings of Machine Learning Research 105:1–10, 2019 Conformal and Probabilistic Prediction and Applications

Split Knowledge Transfer in Learning Under Privileged Information Framework

Niharika Gauraha NIHARIKA.GAURAHA@FARMBIO.UU.SE
Department of Pharmaceutical Biosciences Uppsala University, Uppsala, Sweden
Fabian Söderdahl FABIAN.SODERDAHL@STATISTICON.SE
Statisticon AB, Uppsala, Sweden
Ola Spjuth OLA.SPJUTH@FARMBIO.UU.SE
Department of Pharmaceutical Biosciences Uppsala University, Uppsala, Sweden

Editor: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo and Evgueni Smirnov

Abstract

Learning Under Privileged Information (LUPI) enables the inclusion of additional (privileged) information when training machine learning models, data that is not available when making predictions. The methodology has been successfully applied to a diverse set of problems from various fields. SVM+ was the first realization of the LUPI paradigm which showed fast convergence but did not scale well. To address the scalability issue, knowledge transfer approaches were proposed to estimate privileged information from standard features in order to construct improved decision rules. Most available knowledge transfer methods use regression techniques and the same data for approximating the privileged features as for learning the transfer function. Inspired by the cross-validation approach, we propose to partition the training data into K folds and use each fold for learning a transfer function and the remaining folds for approximations of privileged features - we refer to this as split knowledge transfer. We evaluate the method using four different experimental setups comprising one synthetic and three real datasets. The results indicate that our approach leads to improved accuracy as compared to LUPI with standard knowledge transfer.

Keywords:
Knowledge Transfer, Machine Learning, LUPI, Privileged Information

1. Introduction

A classical supervised machine learning paradigm is: given a set of training examples in the form of Independent and Identically Distributed (IID) pairs

$$(x_1, y_1), ..., (x_l, y_l), \quad x_i \in \mathcal{X}, \quad y_i \in \mathcal{Y} \tag{1}$$

seek a function, in a given set of functions $f(x, \alpha), \alpha \in \Lambda$, that minimizes a loss function. Training examples are represented as features x_i and the same feature space is required for predicting future observations. In the Learning Using Privileged Information (LUPI) paradigm (Vapnik and Vashist, 2009), training examples instead come in the form of IID triplets

$$(x_1, x_1^*, y_1), ..., (x_l, x_l^*, y_l), \quad x_i \in \mathcal{X}, \quad x_i^* \in \mathcal{X}^*, \quad y_i \in \mathcal{Y} \tag{2}$$

where x^* denotes Privileged Information (PI). The objective is the same as in classical machine learning, with the extension that privileged information is available in the training stage.

The LUPI framework has been successfully applied to a diverse set of problems from various fields such as in computer vision (Sharmanska et al., 2013), image classification problems (Lambert et al., 2018) and in drug discovery (Gauraha et al., 2018). SVM+ was the first realization of the LUPI paradigm that has been shown to have limited scalability (Pechyony et al., 2010). To address the scalability problem, the LUPI framework was extended and various knowledge transfer approaches were proposed to transfer knowledge from the space of privileged information to the space where the decision rule is constructed, for example see Vapnik and Izmailov (2015), Vapnik and Izmailov (2016a) and Vapnik and Izmailov (2017). The privileged space represents information that is unavailable at prediction time, for example it could be expensive to generate, time consuming to capture, or restricted due to regulatory or privacy issues. In the knowledge transfer LUPI, these privileged features are commonly approximated using regression methods constructed on standard features, and the same data is used for learning the regression function and for approximating the privileged features used for learning the decision rules.

Inspired by the cross-validation approach, we propose to partition the training data into K folds and use each fold for learning the regression function and the remaining (K-1) folds for approximating the corresponding privileged features which is used for learning the decision rule. Then we use the synergy method (Vapnik and Izmailov, 2016b) to combine the results from K decision rules. We refer to this method as split knowledge transfer, and we compare it with standard knowledge transfer in LUPI, using synthetic and real datasets for classification problems.

The paper is organized in the following way. In section 2, we outline the background concepts and notations used throughout the paper. In Section 3 we introduce split knowledge transfer in LUPI. In section 4 we perform numerical analysis on a set of real data sets. In Section 5, we summarize our results and in Section 6 we conclude and discuss implications and future outlook.

2. Background and Notations

In this section, we fix notations and assumptions used throughout the paper and we provide a brief background on the LUPI framework.

2.1. Notations and Assumptions

In this paper we focus on binary classification problems. The object space is denoted by $\mathcal{X} \in \mathbb{R}^p$, where p is the number of standard features and the label space is denoted by $\mathcal{Y} \in \{-1, 1\}$. The privileged feature space is denoted by $\mathcal{X}^* \in \mathbb{R}^m$, where m is the number of privileged features. We assume that each training example consists of corresponding objects in decision space ($x_i \in \mathcal{X}$), privileged space ($x_i \in \mathcal{X}^*$) and its label ($y_i \in \mathcal{Y}$), and a training set consists of ℓ training examples, $\{(x_i, x_i^*, y_i)\}_{i=1}^{\ell}$. However, a test object consists of only an object in the decision space, $x \in \mathcal{X}$.

We denote the design matrix by $\mathbf{X}_{\ell \times p} = (x_1, \ldots, x_\ell)^T$ and the design matrix in the privileged space by $\mathbf{X}^*_{\ell \times m} = (x_1^*, \ldots, x_\ell^*)^T$. We use super-scripts to denote the columns of

$\mathbf{X}$, i.e. $\mathbf{X}^{(j)}$ denotes the j^{th} column, and sub-scripts to denote the rows of the matrix, i.e. $\mathbf{X}_{(i)}$ denotes the i^{th} row. Let $k \subset \{1, \ldots, \ell\}$, then $\mathbf{X}_{-k}$ denotes the design matrix $\mathbf{X}$ with all the rows omitted from the set k.

2.2. LUPI with Knowledge Transfer (KT-LUPI)

Privileged information, i.e. information that is available at time of training but unavailable at prediction time, is not covered in the traditional machine learning paradigm. Originally introduced in Vapnik and Vashist (2009), the LUPI framework allows learning algorithms to use privileged information. The earliest implementation SVM+ (Pechyony et al., 2010) was an extension of the Support Vector Machine (SVM) algorithm, and had limited scalability to larger sample sizes. Consequent work in Vapnik and Izmailov (2015), Vapnik and Izmailov (2016a) and Vapnik and Izmailov (2017) introduced knowledge transfer, addressing the scalability problem in the LUPI framework by transferring the privileged information from privileged feature space $\mathcal{X}^*$ to the object space $\mathcal{X}$. The knowledge transfer method allows the learning problem to be solved by using standard SVM solvers. Below we define LUPI with knowledge transfer (KT-LUPI).

Given ℓ IID triplets $(\mathbf{X}, \mathbf{X}^*, \mathbf{Y}) = \{(x_i, x_i^*, y_i)\}_{i=1}^{\ell}$ for each privileged feature, a regression function is learned by using p standard features in $\mathbf{X}$ as explanatory variables and privileged feature vectors $\mathbf{X}^{*(i)}$, for $i = 1, \ldots, m$, as response variables. In total m regression functions $\phi_j(x), j = 1, \ldots, m$ are learned. Various regression techniques can be used for approximating privileged features.

The design matrix is augmented with the predicted values from the m regressions, yielding a modfied dataset

$$\bar{\mathbf{X}} = \begin{bmatrix} x_1 & \phi_1(x_1) & \phi_2(x_1) & \ldots & \phi_m(x_1) \\ x_2 & \phi_1(x_2) & \phi_2(x_2) & \ldots & \phi_m(x_2) \\ \multicolumn{5}{c}{\dotfill} \\ x_\ell & \phi_1(x_\ell) & \phi_2(x_\ell) & \ldots & \phi_m(x_\ell) \end{bmatrix} \tag{3}$$

Finally we train an SVM on the modified dataset $(\mathbf{Y}, \bar{\mathbf{X}})$, and learn a decision rule F in $(m + p)$-dimension decision space. Given a new object in the decision space $x \in \mathcal{X}$, we compute the m-regression estimates $\hat{x}^*$ using m-regression functions learned previously, then inference is made in the $m + p$ dimension space using the decision rule F.

3. LUPI with Split Knowledge Transfer (SKT-LUPI)

As mentioned previously, in LUPI with knowledge transfer approach the same data is used to learn the m-regression functions $\Phi = \phi_j(x), j = 1, \ldots, m$, and to approximate the privileged features using standard features which can be given as

$$\Phi(\mathbf{X}) = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) & \ldots & \phi_m(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \ldots & \phi_m(x_2) \\ \multicolumn{4}{c}{\dotfill} \\ \phi_1(x_\ell) & \phi_2(x_\ell) & \ldots & \phi_m(x_\ell) \end{bmatrix} \tag{4}$$

Inspired by the cross-validation approach, we propose LUPI with Split Knowledge Transfer, SKT-LUPI, where the training set is split into K folds of equal (or almost equal) size; in our experiments we use $K = 5$ or $K = 10$. For each $k = 1, \dots, K$, we use the k^{th} fold $(\mathbf{X}_k, \mathbf{X}_k^*)$ for learning the m transfer function $\Phi_k = (\Phi_{k1}, \dots \Phi_{km})^T$ and the rest of the training set is augmented with its predicted values from m regressions resulting in a modified design matrix $\bar{\mathbf{X}}_k = (\mathbf{X}_{-k}\ \Phi_k(\mathbf{X}_{-k})$. Using this augmented design matrix $\bar{\mathbf{X}}_k$ and the corresponding labels, a decision rule, F_k is constructed. Similarly, K modified design matrices are computed for each fold using the corresponding regression functions, and K decision rules $F_1, \dots, F_k$ are constructed.

For a new test object $x \in \mathcal{X}$, we compute $\hat{x}_1^*, \dots, \hat{x}_K^*$ regression estimates using regression functions learned previously, then inference is made for each combination (x, x_k^*), for $k = 1, \dots, K$ in the $m + p$ dimension space using the corresponding decision rule F_k. Finally, its label is predicted by combining results obtained from K decision rules. We use the synergy method (Vapnik and Izmailov, 2016b) to combine the results from K decision rules. The estimated conditional probability, using Platt's scaling (Platt et al., 1999), of an SVM is a monotonically increasing function of its score. The average of monotone estimated conditional probabilities across various models are monotone, and has been shown to be more accurate than the individual scores in Vapnik and Izmailov (2016b). The SKT-LUPI method is summarized in Algorithm 1.

Algorithm 1 LUPI with split knowledge transfer (SKT-LUPI)

```
Input: training data: $\mathbf{X}$, privileged data: $\mathbf{X}^*$,
a training algorithm for a regression model: $\mathcal{A}$, number of folds: $K$
Output: regression functions $\Phi_k(x)$ and $F_k(x)$, for $k = 1, \dots, K$
Steps:
Partition the training set into K folds, $\{\mathbf{X}_k\}$ and $\{\mathbf{X}_k^*\}, k = 1, \dots, K$
for $k$ = 1...$K$ do
    for $j$ = 1...$m$ do
        Train regression function: $\phi_{kj} = \mathcal{A}(\mathbf{X}_{-k},\ \mathbf{X}_{-k}^{*(j)})$
    end
    $\Phi_k = (\phi_{k1}, \dots, \phi_{km})^T$
    Construct the augmented design matrix: $\bar{\mathbf{X}}_k = (\mathbf{X}_{-k}\quad \Phi_k(\mathbf{X}_{-k})$
    Learn the $k^{th}$ decision rule $F_k$ using the modified dataset $(\bar{\mathbf{X}}_k,\ \mathbf{Y}_{-k})$
end
return  $\Phi_k(x)$ and $F_k(x),\ k = 1, \dots, K$
```

4. Experiments

We compared SKT-LUPI with KT-LUPI on a simulated dataset, two classification datasets (Parkinsons and Ionoshpere) from the UCI machine learning repository (Lichman et al., 2013) and a classification dataset (kc2) from Shirabad and Menzies (2005). In the first experiment, we simulate the training and test dataset independently as suggested in Vapnik and Izmailov (2016a). Experiments 2, 3 and 4 are performed using the real world datasets, the specific properties of the corresponding training (80%) and test (20%) sets are given in Table 1. The selection of privileged features for datasets Ionosphere and kc2 were done as suggested in Izmailov et al. (2017) and for the dataset Parkinsons as suggested in Vapnik and Izmailov (2017).

In all the experiments, the SVM algorithm is used to create decision rule with RBF kernel. The SVM regularization parameter C and the RBF kernel parameter γ were selected using the 6-fold cross-validated error rate over a two-dimensional grid, where $log_2(C)$ ranged of from -5 to $+5$ with step 0.5, and $log_2(\gamma)$ ranged -6 to $+6$ with step 0.5. In the first three experiments the following four types of classification scenarios are considered:

1. SVM on standard features: an SVM algorithm is used to create a decision rule using an RBF kernel on standard features.

2. Knowledge transfer LUPI (KT-LUPI): knowledge transfer from privileged feature to the space of standard features is realized using multiple linear regression or kernel ridge regression with RBF kernel. After augmenting standard features with regressed values of privileged features, an SVM algorithm is used to create a decision rule with RBF kernel on the augmented decision space.

3. Split Knowledge transfer LUPI (SKT-LUPI): knowledge transfer from privileged feature to the space of standard features is realized using SKT-LUPI (Algorithm 1) with multiple linear regression or kernel ridge regression with RBF kernel. After augmenting standard features with regressed values of privileged features, an SVM algorithm is used to create a decision rule with RBF kernel on the augmented decision space.

4. SVM on all features (standard and privileged features): an SVM algorithm is used to create a decision rule with RBF kernel on all features.

Table 1: Description of the datasets used in the evaluation.

Dataset	Training	Test	Standard Features	Privileged Features
Parkinsons	156	39	12	10
Ionosphere	280	71	30	4
kc2	303	152	114	30

Experiment 1: Synthetic Dataset

We first consider the simple synthetic example as given in Vapnik and Izmailov (2016a), where two-dimensional random points $(x^{(1)}, x^{(2)})$ as standard features are generated as uniformly distributed in the square $[-1, 1] \times [-1, 1]$, a privileged feature is computed as $x^{(3)} = x^{(1)} + x^{(2)} + .01 * W$, where $W \sim Normal(0, 1)$, and the label is computed as $y = sign(x^{(1)} + x^{(2)})$. We considered all the four classification scenarios mentioned above, where knowledge transfer from privileged feature to the space of standard features is performed using multiple linear regression in KT-LUPI and SKT-LUPI. The average error rate for training sizes (25, 35, 45 55) and a test data of size 10000, over 20 runs are reported in Table 2.

Table 2: Comparison of error rates (in %) on the synthetic dataset between four types of classification scenarios considered: SVM on standard features, LUPI with knowledge transfer (KT-LUPI), LUPI with split knowledge transfer (SKT-LUPI) with 5-folds, and SVM on privileged features. Multiple linear regression is used for knowledge transfer.

Training Size	SVM on standard features	KT-LUPI	SKT-LUPI	SVM on PI
25	9.86	7.68	6.10	3.39
35	7.26	5.42	4.80	3.19
45	7.21	4.69	4.22	2.98
55	6.88	5.04	4.84	2.51

Table 3: Comparison of error rates (in %) on modified real datasets between four types of classification scenarios considered: SVM on standard features, LUPI with knowledge transfer (KT-LUPI), LUPI with split knowledge transfer (SKT-LUPI) with 10-folds, and SVM on all features (standard features and privileged features). Multiple linear regression is used for knowledge transfer.

Dataset	SVM on standard features	KT-LUPI	SKT-LUPI	SVM on all features
Parkinsons	10.25	8.58	7.05	6.53
Ionosphere	5.49	4.92	4.29	4.85
kc2	17.28	17.09	16.99	16.80

Experiment 2: Knowledge Transfer using Linear Transformation

In this experiment we then used the three real dataset as given in Table 1. All the four classification scenarios as in Experiment 1 are considered, here for the fourth case we considered all the features (standard+privileged), and multiple linear regression was used for knowledge transfer in KT-LUPI and SKT-LUPI. The average error rates over 20 runs are reported in Table 3.

Experiment 3: Knowledge Transfer using non-Linear Transformation

In this experiment we consider three datasets as given in Table 1. All the four classification scenarios as in Experiment 2 are considered, except that we used kernel ridge regression in KT-LUPI and SKT-LUPI for knowledge transfer. The average error rate over 20 runs are reported in Table 4.

Table 4: Comparison of error rates (in %) on modified real datasets between four types of classification scenarios considered: SVM on standard features, LUPI with knowledge transfer (KT-LUPI), LUPI with split knowledge transfer (SKT-LUPI) with 10-folds, and SVM on all features (standard features and privileged features). Kernel ridge regression is used for knowledge transfer.

Dataset	SVM on standard features	KT-LUPI	SKT-LUPI	SVM on all features
Parkinsons	10.25	8.97	7.82	6.53
Ionosphere	5.49	5.07	4.57	4.85
kc2	17.28	17.23	16.95	16.80

Experiment 4: Analysis of Variance on Error rate

This experiment is performed to analyze the variance on error rate for KT-LUPI and SKT-LUPI. The spread of error rates over 10 runs using the synthetic dataset (with 25 training examples), Parkinsons, kc2 and Ionoshpere datasets are plotted in Figure 1.

5. Results and Discussion

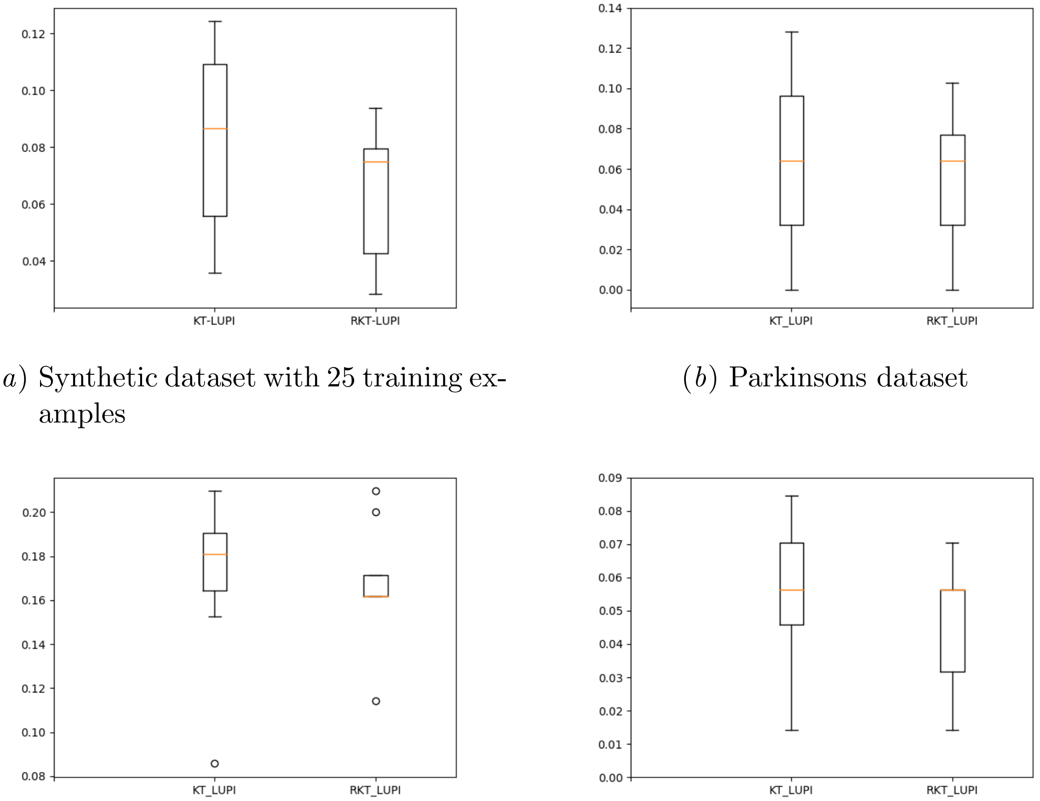
The aim of this paper was to introduce split knowledge transfer and to explore its performance in different settings. Our results indicate that SKT-LUPI yields lower error rates than KT-LUPI and SVM on standard features when evaluated on a synthetic dataset (see Experiment 1 and Table 2), and in Experiment 2 and 3 using three real world datasets for linear regression (Table 3) and non-linear regression (Table 4). We also observed that SKT-LUPI yields lower error rates than SVM on all features for Ionosphere dataset. Results from Experiment 4 show indications that SKT-LUPI has lower variance on error rates as compared to KT-LUPI.

SKT-LUPI is, because of the K folds, more computationally demanding than KT-LUPI, however the overhead is not large for the datasets in this study and the implementation can easily be parallelized if applied to larger datasets.

6. Conclusion and Future Scope

In this manuscript we introduced LUPI with split knowledge transfer (SKT-LUPI), where the training data is divided in K folds and where each fold iteratively is used for learning the transfer function and the remaining folds for approximations of privileged features. We demonstrated its advantages compared with standard KT-LUPI, with SKT-LUPI showing a lower error rate and reduced variance on synthetic and real data. The methodology is general and can be directly applied to LUPI settings with knowledge transfer.

Future plans include to extend SKT-LUPI for other machine learning frameworks such as artificial neural networks, and apply it to regression and unsupervised problems.



(a) Synthetic dataset with 25 training examples

(b) Parkinsons dataset

(c) kc2 dataset

(d) Ionosphere dataset

Figure 1: Results comparing the error rates over 10 runs for KT-LUPI and SKT-LUPI on a synthetic dataset (with 25 training examples) and the Parkinsons, kc2 and Ionoshpere datasets.

Acknowledgements

This project received financial support from the Swedish Foundation for Strategic Research (SSF) as part of the HASTE project under the call 'Big Data and Computational Science'. The computations were performed on resources provided by SNIC through Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) under project SNIC 2019/8 – 15.

References

Niharika Gauraha, Lars Carlsson, and Ola Spjuth. Conformal prediction in learning under privileged information paradigm with applications in drug discovery. In *Conformal and Probabilistic Prediction and Applications*, pages 147–156, 2018.

Rauf Izmailov, Blerta Lindqvist, and Peter Lin. Feature selection in learning using privileged information. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 957–963. IEEE, 2017.

John Lambert, Ozan Sener, and Silvio Savarese. Deep learning under privileged information using heteroscedastic dropout. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8886–8895, 2018.

Moshe Lichman et al. Uci machine learning repository, 2013.

Dmitry Pechyony, Rauf Izmailov, Akshay Vashist, and Vladimir Vapnik. Smo-style algorithms for learning using privileged information. In *DMIN*, pages 235–241, 2010.

John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

Viktoriia Sharmanska, Novi Quadrianto, and Christoph H Lampert. Learning to rank using privileged information. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 825–832, 2013.

J Sayyad Shirabad and Tim J Menzies. The promise repository of software engineering databases. *School of Information Technology and Engineering, University of Ottawa, Canada*, 24, 2005.

Vladimir Vapnik and Rauf Izmailov. Learning with intelligent teacher: Similarity control and knowledge transfer. In *International Symposium on Statistical Learning and Data Sciences*, pages 3–32. Springer, 2015.

Vladimir Vapnik and Rauf Izmailov. Learning with intelligent teacher. In Alexander Gammerman, Zhiyuan Luo, Jesús Vega, and Vladimir Vovk, editors, *Conformal and Probabilistic Prediction with Applications*, pages 3–19, Cham, 2016a. Springer International Publishing.

Vladimir Vapnik and Rauf Izmailov. Synergy of monotonic rules. *The Journal of Machine Learning Research*, 17(1):4722–4754, 2016b.

Vladimir Vapnik and Rauf Izmailov. Knowledge transfer in svm and neural networks. *Annals of Mathematics and Artificial Intelligence*, 81(1-2):3–19, 2017.

Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.