Learning with Privileged Information: Decision-Trees and Boosting

Rahul Pasunuri Indiana University
Phillip Odom Indiana University
Tushar Khot Allen Institute of AI
Kristian Kersting TU Dortmund
Sriraam Natarajan Indiana University

Abstract

We consider the problem of learning with privileged information where the goal is to learn a classifier that uses features not available at test time to learn a better model at training time. While the earlier approaches under this formalism focused mainly on SVMs, we extend the setting to tree-based learners—decision trees and boosting for learning with privileged information. Our methods use privileged features to create additional labels for each example and use these privileged labels to guide the learning algorithms. We derive the theory and empirically validate the effectiveness of our learning approach both in the case of decision-trees and boosting. Our methods outperform the SVM based learner with privileged information.

1 Introduction

Machine learning methods that consider learning from sources beyond just a single set of labeled data have long been explored under several paradigms - learning with advice [Towell and Shavlik, 1994; Fung *et al.*, 2002; Maclin *et al.*, 2005; Kunapuli *et al.*, 2013], learning from preferences [Boutilier, 2002; Boutilier *et al.*, 2010; Drummond and Boutilier, 2014], learning from qualitative constraints [Altendorf *et al.*, 2005; Yang *et al.*, 2013], active learning [Settles, 2012], and transductive learning [Joachims, 1999] etc.

Recently, Vapnik and Vashist [Vapnik and Vashist, 2009] introduced the problem of learning from privileged information where more information in the form of features are provided during training. During testing, this information is not available and hence the classifier cannot use these features in the model but can use them to learn a more accurate model. Vapnik and Vashist learned an SVM in the privileged space and used this SVM to control the margin in the full training space. This approach was later adapted by Sharmanska et al. [Sharmanska *et al.*, 2013] where an interesting observation was made in that even random but informative privileged features helped improve the performance of the classifier.

While interesting, most of the original work and its adaptations were based on SVMs. In this work, we aim for learning more interpretable models, precisely decision-trees and their ensembles. To this effect, we develop the first set of tree based algorithms that can learn in the presence of privileged information. The key idea is to use the privileged features to group the examples into clusters such that examples within a cluster are more similar according to the privileged features. This is to say that the privileged features "prefer" that the examples be labeled accordingly when learning the final model. Consequently, these privileged labels are used along with the original labels to obtain a score that is a combination of these two labels given the training/test features.

The intuition is that while the privileged features provide some extra information, they are not being fully relied on when building the model. The resulting model, in essence, is a trade-off between the privileged information and the fully observed features - as is typically done in advice-based methods where the data and the expert knowledge are explicitly considered when learning the model. We evaluate the decision-tree algorithm and the ensemble learning method based on functional-gradient boosting on several benchmark data sets and a new imaging data set, that of predicting the shape of galaxies given astronomical features. Our results show that the learners generally benefit from privileged information, are never worse than just using the original data and the ensemble method typically outperform the SVM based classifier on most data sets.

We make a few key contributions in this work: First, as far as we are aware, this is the first work on learning tree based models in the presence of privileged information. Second, we present the algorithm for learning decision-trees that includes a principled way of trading-off between the privileged information and the observed data. Third, we consider the functional-gradient boosting extension of trees where we present the gradient updates and position the work in context to the existing work on SVMs. Finally, we perform exhaustive empirical evaluation that demonstrates the effectiveness of the proposed approaches on several benchmark test beds.

The rest of the paper is organized as follows: we next present the background and related work. Then, we consider the decision-trees with privileged information and present the learning algorithm. We next extend these ideas to learn ensembles of trees using gradient-boosting. Finally, we present extensive experiments before concluding by outlining areas of future work.

2 Background

We discuss the previous work on learning with privileged information followed by the related work in gradient-boosting.

2.1 Learning with Privileged Data

Human-machine interaction has been widely studied in artificial intelligence and machine learning through many different frameworks [Towell and Shavlik, 1994; Fung *et al.*, 2002; Maclin *et al.*, 2005; Boutilier, 2002]. One particular framework, learning with privileged information, is inspired by richer forms of interactions between human teachers and students [Vapnik and Vashist, 2009]. Particular (labeled) examples are given to the student along with explanations and intuitions that are able to speed up the comprehension of novel concepts. While privileged information is often used in classrooms and lecture halls, it is completely ignored by standard machine learning algorithms.

More formally, learning with privileged information assumes that more information is known about the training examples. However, as the expert is not available for testing, this additional information is not available at test time. Thus, training examples have the form $< \mathbf{x}_i^{\mathbf{CF}}, \mathbf{x}_i^{\mathbf{PF}}, y_i >$ while testing examples have the form $< \mathbf{x}_i^{\mathbf{CF}}, y_i >$. $\mathbf{CF}$ refers to the classifier features available during testing and $\mathbf{PF}$ to denote privileged features. Refer to section 3 for more details on notations.

Learning algorithms for privileged information have previously mainly focused on support vector machines (SVMs) [Vapnik and Vashist, 2009; Sharmanska *et al.*, 2013]. The original formulation—SVM+ [Vapnik and Vashist, 2009]—learned the difficulty of each training example. The key idea was to learn an SVM in the privileged space (using $[< \mathbf{x}_i^{\mathbf{PF}}, y_i >]$) and find the margin with respect to this SVM for each training example. Training examples closer to the margin are considered "more difficult" as they are closer to the decision boundary while examples farther from the margin are considered "less difficult". This information is similar to the intuition or explanation given by a human teacher. Vapnik and Vashist [Vapnik and Vashist, 2009] showed how this information can be used when learning an SVM over the standard features.

Since the introduction of the new learning paradigm and the corresponding SVM+ approach there is a growing body of work on it. Pechyony and Vapnik [Pechyony and Vapnik, 2010] started to develop a theoretical justification of the learning setting. Liang *et al.* [Liang *et al.*, 2009] established links between the SVM+ and multi-task learning. Recently, Hernández-Lobato et al. [Hernández-Lobato *et al.*, 2014] showed that privileged information can naturally be treated as noise in the latent function of a Gaussian Process classifier (GPC). That is, in contrast to the standard GPC setting, the latent function becomes a natural measure of confidence about the training data by modulating the slope of the GPC sigmoid likelihood function. Most closely related to our work, Chen *et al.* [Chen *et al.*, 2012] extend the setting to AdaBoost, and Lapin *et al.* [Lapin *et al.*, 2014] related privileged information to importance weighting within SVMs. Also, learning with privileged information has been proven beneficial in computer vision domains [Sharmanska *et al.*, 2013]. Decision tree learners, however, have not been considered yet. Moreover, instead of example weighting we establish a novel connection to advice-based machine learning, see Section 3 for explanation. We show that the prior knowledge expressible with privileged features can also be encoded by labels associated with every training example that in turn guide the learning algorithms. Moreover, boosting is realized by regularizing the log-likelihood via the KL divergence between the distribution using all features and the distribution using the features available at test time only.

2.2 Functional Gradient Boosting

Many probabilistic learning methods learn the conditional distribution $P(y_i|\mathbf{x}_i)$ using standard techniques such as gradient-descent performed on the log-likelihood w.r.t. parameters to find the best set of parameters that model the training data. Functional Gradient Boosting methods (GB) [Friedman, 2001; Dietterich *et al.*, 2008; Natarajan *et al.*, 2012; 2015] on the other hand represent the likelihood in a functional form (typically using the sigmoid function)

$$P(y_i|\mathbf{x}_i) = \frac{e^{\psi(y_i=\hat{y}_i|\mathbf{x}_i)}}{\sum_{y'} e^{\psi(y_i=y'|\mathbf{x}_i)}} \qquad (1)$$

where ψ is a regression function defined over the examples. Given this representation, GB methods obtain the gradient of the log-likelihood w.r.t. to $\psi(x_i)$ for each example, x_i as:

$$\Delta(y_i) = I(y_i = 1) - P(y_i = 1|\mathbf{x}_i), \qquad (2)$$

where *I* is an indicator function which returns 1 for positive examples and 0 for negative examples in a binary classification task. The GB approach starts with an initial regression function, ψ_0 to compute the probabilities of the training examples and thereby the gradients Δ_0. A regression function (typically a tree), h_0 is fit on the training examples with the gradients as the target regression values. This learned function is now added to ψ_0 and the process is repeated with $\psi_1 = \psi_0 + h_0$. Given that the stage-wise growth of trees resembles boosting, and the process involves computing gradients of functions, this method is called *Gradient Boosting*.

3 Learning Privileged Trees

Our problem is formally defined as:

Given: A set of training examples $[\langle y_i, \mathbf{x}_i^{\mathbf{CF}}, \mathbf{x}_i^{\mathbf{PF}} \rangle]$ and a set of test examples $[\langle y_i, \mathbf{x}_i^{\mathbf{CF}} \rangle]$, where

$$\mathbf{F} = \mathbf{CF} \cup \mathbf{PF} \quad \& \quad \mathbf{CF} \cap \mathbf{PF} = \emptyset$$

To do: Learn a classifier that employs only the classifier features $\mathbf{CF}$ for classifying the test data and can utilize the privileged features $\mathbf{PF}$ effectively in learning a better model.

$\mathbf{F}$ is the set of all features, $\mathbf{CF}$ is the set of features that are available at both training and test time (and we call them *classifier features*), $\mathbf{PF}$ are the privileged features that are accessible only during training and not during testing, y_i is the label of the i^{th} example and $\mathbf{x}_i$ are the features of that example. We use [] to denote sets. For example, the input to the algorithm is the set of all examples $[< y_i, \mathbf{x}_i^{\mathbf{CF}}, \mathbf{x}_i^{\mathbf{PF}} >]$. We consider two different approaches to leverage $\mathbf{PF}$: Advice-based and Margin-based. While our approaches are designed with tree-based classifiers, they can easily be extended to other clustering/classification techniques.

3.1 Advice-based

Following advice-based machine learning methods [Fung *et al.*, 2002; Kunapuli *et al.*, 2013; Towell and Shavlik, 1994], one can view privileged information as "advice" that guides the learning algorithm to a better model. The key idea in advice framework is that a human can provide preferences over labels given some feature combinations and this is explicitly weighted against data to learn a model [Odom *et al.*, 2015]. Similarly, we use the information from privileged features to specify "preferences" over labels for every example by grouping the examples according to a learned model, for example, a tree. Every example that reaches a particular leaf of the tree is assigned a *privileged label*. The intuition is that the examples that reach a particular leaf of a privileged tree (that is learned only with $\mathbf{PF}$) share the same characteristics (feature combinations) and hence are grouped together. Once these clusters are obtained, the examples inside a cluster are provided with the same privileged label. Note that while we employ trees to group the examples based on the privileged information, any clustering method can be used to group these examples. Now when learning the model using $\mathbf{CF}$, the scores based on the privileged labels are combined with the scores from the true labels. This will make it likely that examples within the same cluster due to the $\mathbf{PF}$ model be grouped together in the $\mathbf{CF}$ model as well.

We now describe our learning algorithm, *DTree+* in more detail. The first step is to assign privileged label to each example by learning a decision-tree using only $\mathbf{PF}$ and the true labels in the data (refer to Algorithm 1). We denote the privileged label as y_i^p for each x_i obtained using Algorithm 2.

We then learn a decision tree using only $\mathbf{CF}$ to build the model that will be used for prediction. To learn the decision tree, we score each split of the decision-tree using *entropy gain* (G) [MacKay, 2003]. For splitting in any node, we consider two different gains – G due to the true labels denoted as $G_{\mathbf{DL}}$ and G due to privileged labels denoted as $G_{\mathbf{PL}}$.

While, in principle, a single combination function could be used to combine these gains, the range of these values can be significantly different due to the different number of true labels (used by $G_{\mathbf{DL}}$) and privileged labels (used by $G_{\mathbf{PL}}$). Hence, we propose to use the following measure:

$$If \quad G_{\mathbf{DL}} < G_{\mathbf{PL}}\ then \quad Score1 = \frac{2G_{\mathbf{DL}} \times G_{\mathbf{PL}}}{G_{\mathbf{DL}} + G_{\mathbf{PL}}}$$
$$else \quad Score2 = \frac{G_{\mathbf{DL}}^2 + G_{\mathbf{PL}}^2}{G_{\mathbf{DL}} + G_{\mathbf{PL}}} \quad (3)$$

We want the gain from the labeled data to dominate the score as much as possible compared to the gain from the privileged labels. This combination function always ensures that the final score weighs the true labels more (this is shown in Figure 1). We also use a scaling factor α to scale the privileged gain $G_{\mathbf{PL}}$ up or down based on the quality of $\mathbf{PF}$. The choice of α is determined using line search. Finally, it is worth noting that the mean of $Score1$ and $Score2$ is also the mean of the two gains, i.e.,

$$Score1 + Score2 = G_{\mathbf{DL}} + G_{\mathbf{PL}}$$

Algorithm 1 D-Tree+

```
Input: $[< y_i, \mathbf{x}_i^{\mathbf{CF}}, \mathbf{x}_i^{\mathbf{PF}} >]$
Output: Tree T that uses only $\mathbf{CF}$.
Begin
$[y_i^p]$ = PTreeLearn($[< y_i, \mathbf{x}_i^{\mathbf{PF}} >]$)
Create $[< y_i^p, \mathbf{x}_i^{\mathbf{CF}} >]$ as privileged label data
while Tree Depth not reached do
  for all $f_j \in \mathbf{CF}$ do
    Compute $G_{\mathbf{DL}}$ using $y$ and $G_{\mathbf{PL}}$ using $y^p$ with the current set of examples
    Combine $G_{\mathbf{DL}}$ and $G_{\mathbf{PL}}$ using Eqn 3
  end for
  Choose $f = argmax_{f_k}(Score(f_k))$ for the current split and split examples
end while
return Tree
```

Hence, once each feature is scored using the gain, we simply choose the feature from $\mathbf{CF}$ with the maximum score for the current split, split the examples accordingly and continue the process till the tree is completely learned. Note that our tree construction is the standard decision tree learning algorithm with the modified scoring function.

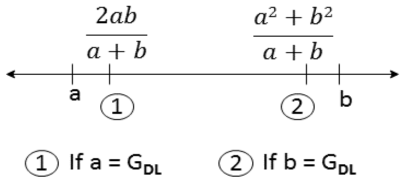


Figure 1: Combination of the classifier and privileged features. The goal is to keep the final score closer to the classifier gain as much as possible.

Algorithm 2 PTreeLearn

```
Input: $[< y_i, \mathbf{x}_i^{PF} >]$
Output: $[y_i^p]$ where $y_i^p$ is privileged label for $i^{th}$ example
PTree=DTLearn($< y_i, \mathbf{x}_i^{PF} >$)
for j=1 to numLeafs(PTree) do
  $\mathbf{E}_j$ - set of examples in leaf j
  Assign j as privileged label ($y_k^p$) $\forall e_k \in \mathbf{E}_j$
end for
return $[y_i^p]$
```

3.2 Margin-based boosting

While the previous approach used the privileged information tree to influence the final model learned over $\mathbf{CF}$, it did not leverage this learned model to further tune the privileged tree labels. By attempting to reduce the margin between the two models, we can potentially find a consistent labeling based on both the privileged and classifier features. We use gradient boosting (GB) to iteratively learn models using $\mathbf{PF}$ and $\mathbf{CF}$ features while simultaneously reducing the margin between two models.

In Vapnik's SVM+ model, he used the $\mathbf{PF}$ features to define an oracle function that can predict the slack on each example. In our probabilistic framework, we use the $\mathbf{PF}$ features to build an oracle model that can predict the true probability distribution of each example which is not captured by the discrete class labels. We learn a model that minimizes the error of the model over the training labels and the margin between the true distribution $P_D(y|\mathbf{x}^{\mathbf{PF}})$ and $P(y|\mathbf{x}^{\mathbf{CF}})$,

$$\min - \sum_i [-log(P(y_i|\mathbf{x}_i^{\mathbf{CF}})) + \alpha \cdot KL(P_D(y_i|\mathbf{x}_i^{\mathbf{PF}})||P(y_i|\mathbf{x}_i^{\mathbf{CF}}))]$$

where $-\sum_i log(P(y_i|\mathbf{x}_i^{\mathbf{CF}})$,the negative log-likelihood of the training data is used to model the error and KL denotes the KL divergence between P_D and P and is equal to $\sum_i P_D(i) log \frac{P_D(i)}{P(i)}$. We use α to model the trade-off between fitting to the labeled data versus fitting to the distribution learned over $\mathbf{PF}$. We can now use gradient boosting with respect to $\psi(\mathbf{x}^{\mathbf{CF}})$ to minimize this objective function. The first term of our objective function is the standard log-likelihood function which has the gradient:[1][2]

$$\frac{\partial \sum_i log(P(y_i|\mathbf{x}_i^{\mathbf{CF}})}{\partial \psi(\mathbf{x}_i^{\mathbf{CF}})} = I(y_i = 1) - P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}})$$

For the second term, we derive the gradients below:

$$\begin{aligned}
&\frac{\partial KL(P_D(y_i|\mathbf{x}_i^{\mathbf{PF}}), P(y_i|\mathbf{x}_i^{\mathbf{CF}}))}{\partial \psi(\mathbf{x}_i^{\mathbf{CF}})} \\
=&\frac{\partial \left(\Sigma_{y_i=\{0,1\}} P_D(y_i|\mathbf{x}_i^{\mathbf{PF}}) log(P_D(y_i|\mathbf{x}_i^{\mathbf{PF}})/P(y_i|\mathbf{x}_i^{\mathbf{CF}}, \psi))\right)}{\partial \psi(\mathbf{x}_i^{\mathbf{CF}})} \\
=&P_D(y_i = 0|\mathbf{x}_i^{\mathbf{PF}})P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}}) \\
&- P_D(y_i = 1|\mathbf{x}_i^{\mathbf{PF}})(1 - P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}})) \\
=&P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}}) - P_D(y_i = 1|\mathbf{x}_i^{\mathbf{PF}})
\end{aligned}$$

We combine the gradient terms to get the final gradient for each example:

$$\begin{aligned}
\Delta(\mathbf{x}_i^{\mathbf{CF}}) = &I(y_i = 1) - P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}}) \\
&- \alpha \cdot \left(P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}}) - P_D(y_i = 1|\mathbf{x}_i^{\mathbf{PF}})\right) \quad (4)
\end{aligned}$$

Intuitively, if the learned distribution has a higher probability of an example belonging to the positive class compared to the true distribution, $P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}}) - P_D(y_i = 1|\mathbf{x}_i^{\mathbf{PF}})$ would be positive and the gradient would be pushed lower. Hence the additional term would push the gradient (weighted by α) towards the true distribution as predicted by $\mathbf{PF}$.

The parameter, α controls the influence of the privileged data on the learned distribution. In the extreme case of $\alpha = 0$, $\mathbf{PF}$ are completely ignored and we end up with the standard functional gradient. Similar gradients can be computed for learning the true distribution using $\mathbf{PF}$ by switching P and P_D.

Algorithm 3 GB+: Boosting with privileged information

```
Input: $< y_i, \mathbf{x}_i^{\mathbf{CF}}, \mathbf{x}_i^{\mathbf{PF}} >$
Output: Model over classifier features, $P(y|\mathbf{x}^{\mathbf{CF}})$
$\psi_0^{\mathbf{PF}} = 0$
for m=1 to M do
    $< \mathbf{x}_i^{\mathbf{CF}}, \Delta(\mathbf{x}_i^{\mathbf{CF}}) >$ = Create examples using Eqn 4
    $h_m$ = FitRegressionTree($\mathbf{x}_i^{\mathbf{CF}}, \Delta(\mathbf{x}_i^{\mathbf{CF}})$)
    $\psi_m = \psi_{m-1} + h_m$
    $< \mathbf{x}_i^{\mathbf{PF}}, \Delta_D(\mathbf{x}_i^{\mathbf{PF}}) >$ = Create examples using Eqn 5
    $h_m^{\mathbf{PF}}$ = FitRegressionTree($\mathbf{x}_i^{\mathbf{PF}}, \Delta_D(\mathbf{x}_i^{\mathbf{PF}})$)
    $\psi_m^{\mathbf{PF}} = \psi_{m-1}^{\mathbf{PF}} + h_m^{\mathbf{PF}}$
end for
return sigmoid($\psi_M$)
```

$$\begin{aligned}
\Delta_D(\mathbf{x}_i^{\mathbf{PF}}) = &I(y_i = 1) - P_D(y_i = 1|\mathbf{x}_i^{\mathbf{PF}}) \\
&- \alpha \cdot \left(P_D(y_i = 1|\mathbf{x}_i^{\mathbf{PF}}) - P(y_i = 1|\mathbf{x}_i^{\mathbf{CF}})\right) \quad (5)
\end{aligned}$$

Given these gradients, we can now describe our approach called GB+ to perform gradient boosting using privileged information. Similar to standard GB, we iteratively learn regression functions (trees in our case) to fit to these gradients. We perform co-ordinate gradient descent, i.e. we alternate between taking a gradient step along ψ and along ψ_D. Algorithmically, we learn one regression tree using the gradients based on $\mathbf{CF}$, compute the gradients for $\mathbf{PF}$, learn a tree for the $\mathbf{PF}$ and repeat this M times, as shown in Algorithm 3.

Summary: While the specific updates and methodology of the two algorithms differ, the intuition in both is the same in that the privileged information is used to guide the learning algorithm by preferring some examples to be grouped together. We now present our experimental evaluation.

4 Experiments

We aim to answer the following questions:

Q1: How effective is privileged information in decision trees?

Q2: Can iteratively updating the privileged model improve boosting decision tree algorithms?

Q3: Are the algorithms robust to low-quality privileged information?

We present empirical evaluations of our proposed approaches for decision tree learning with privileged information (DT+), and for gradient boosted privileged decision tree learning (GB+). Across all domains, we report accuracy of the learned model on the test set. All our experimental results are obtained by 5-fold cross-validation.

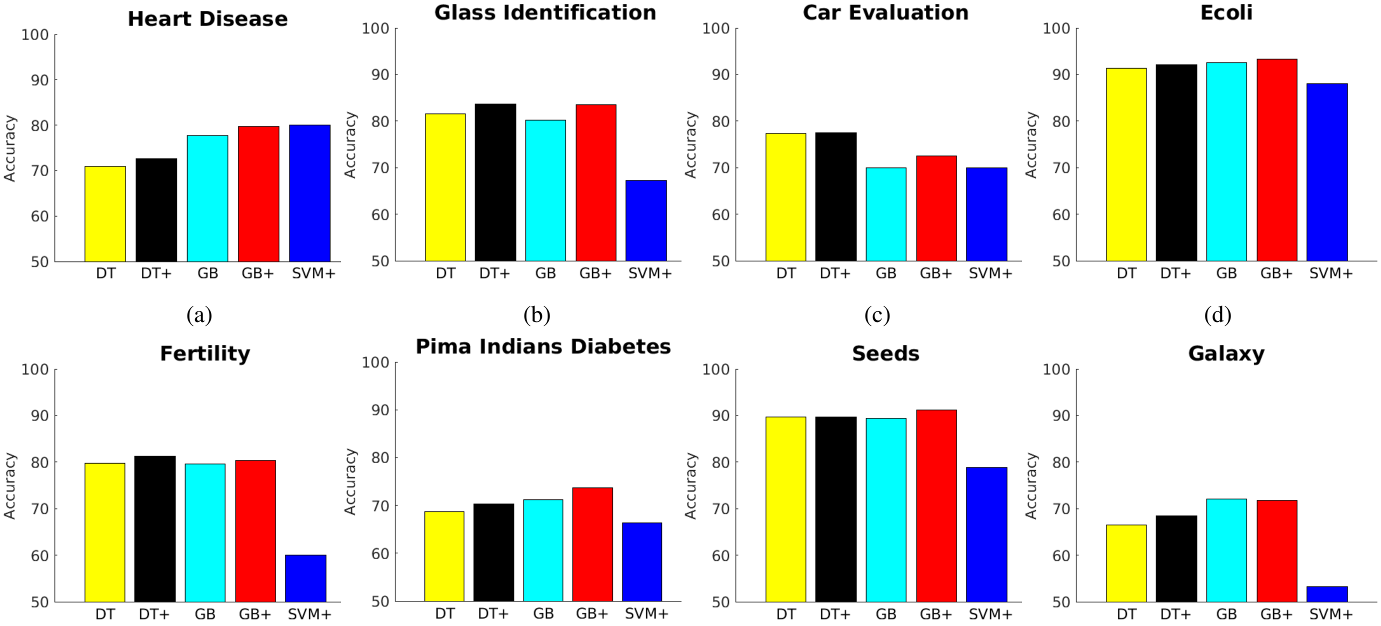
In order to answer these questions, we compare our algorithm against several different baselines. To show the effectiveness of DT+, we compare against standard decision trees (DT). Likewise, we show a similar result for GB+ by comparing against function gradient-boosting (GB). Finally, we also compare with SVM+ which also uses privileged information.

[1] We use $\psi(\mathbf{x}_i^{\mathbf{CF}})$ to denote the potential function given $\mathbf{CF}$.

[2] For both the cases of $y_i = 1$ and $y_i = 0$.

Table 1: An explanation of the various datasets that we use for empirical evaluation.

Domain	Target Attribute	# Exs	# of PF	# of CF
Heart Disease	Diagnosis of Heart Disease	297	7	6
Glass Identification	float/non float(Windows)	214	4	5
Car Evaluation	Acceptable or not	1728	1	5
Ecoli	Protein localization	336	3	4
Fertility	Diagnosis normal or altered	100	3	6
Pima Indians Diabetes	Diabetes status	768	4	4
Seeds	Length of Kernel Groove $\geq$2	199	4	3
Galaxy	Spiral Galaxy or not	505	21	128



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 2: Comparison of the accuracy of our proposed privileged classifiers (DT+,GB+) and each of the baseline methods including the privileged method of SVM+ as well as the standard methods (DT,GB).

We compare against 7 standard classification datasets found in the UCI repository [3] as well as a novel dataset that aims to predict the types of galaxies from features derived from telescope images of those galaxies. A broad overview of the different datasets, their prediction problem, dataset size information, and size of $\mathbf{PF}$ and $\mathbf{CF}$ is shown in Table 1.

The galaxy dataset features are derived from the Sky Survey Database [Willett *et al.*, 2013] which contains images of galaxies. We use features derived from these images to classify galaxy images into spiral/non-spiral shapes [Dhami, 2015]. $\mathbf{CF}$ includes all of the color coherence values (128 total features), and $\mathbf{PF}$ includes shape features like circulatiy, convexity, etc. and color features including color intensity ratios, max color channel values, etc.

4.1 Privileged Decision Trees (DT+)

First, we compare our privileged decision tree learner to standard decision trees. As described previously, we use cross validation and line search in each dataset to set the trade-off between the classifier label entropy gain and the entropy gain of the privileged label (denoted as α in section 3.1) . This means that our algorithm will reduce the impact of the privileged information if it is not useful. A comparison between DT+ and DT can be seen in Figure 2. Note that our approach never performs worse than DT.

Figure 2 shows the comparison across all methods. In many datasets DT+ performs slightly better while in the heart dataset they perform significantly better. It is important to note that previous privileged learning algorithms often fail to significantly outperform the standard approach [Sharmanska *et al.*, 2013] as privileged information may not always be useful. DT+ is effective in using privileged information to learn a better model as shown by DT+ outperforming DT (**Q1**).

4.2 Gradient Boosted Trees (GB+)

Similarly, we compare our gradient boosted trees (GB+) to standard gradient boosting (GB) in Figure 2. Notice that our method, GB+, outperforms GB in most of the domains. How-

[3] http://archive.ics.uci.edu/ml/

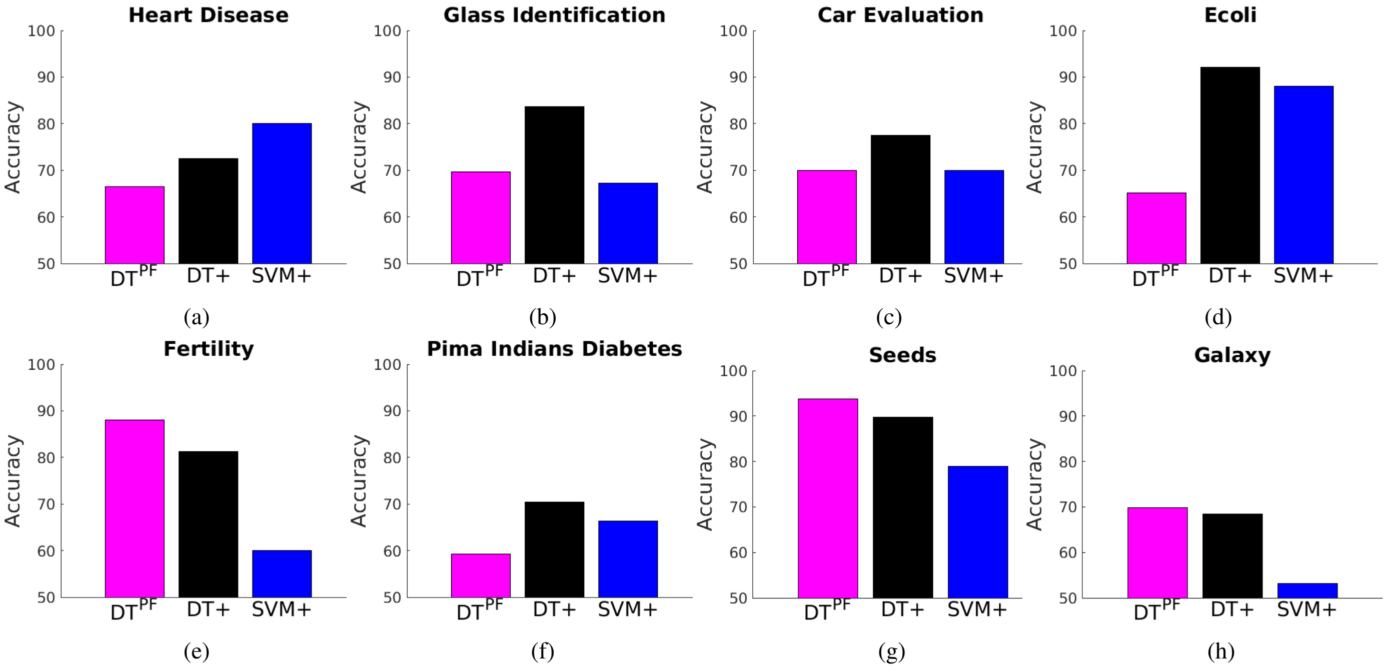


Figure 3: We show the quality of the privileged information (when used to build a classifier), and compare the privileged learner DT+ and the previous privileged approach SVM+.

ever, this improvement is statistically significant only in the diabetes and seeds datasets. Similar to DT+ and DT, GB+ never performs (statistically significantly) worse than GB.

Again, Figure 2 shows the comparisons across all methods. As expected, the boosted methods typically outperform the standard approaches with the notable exception of the car dataset. Thus, iteratively updating the privileged model and learning an ensemble can further improve the learning process (**Q2**). Next, we focus on how our privileged approaches compare with another privileged classifier.

4.3 Privileged Information and SVM+

We now discuss the quality of the privileged information and compare our approach against the previous privileged framework of SVM+. The performance of our algorithm (DT+) is compared to SVM+ and learning only in the privileged space ($DT^{\mathbf{PF}}$) in Figure 3. Notice that across all domains except the heart disease domain, our algorithm outperforms SVM+. Previous results with SVM+ show that the algorithm performs well with high-quality privileged information [Sharmanska *et al.*, 2013] as well as a large number of features. However, many of our experimental domains do not have a large number of features (Table 1) and the performance of $\mathbf{PF}$ is often lackluster (Figure 3).

Even when $\mathbf{PF}$ is not informative, our proposed approach is not hindered. Often, DT+ is still able to utilize the privileged information to learn a better model. DT+ is robust to low-quality privileged information because of 1) use of the combination function (Eqn 3) that tends toward the gain of the classifier features and 2) α parameter that weighs the gain of the privileged information. These factors make our algorithm robust to the quality of privileged information (**Q3**).

5 Conclusion

We presented the first set of approaches to exploit privileged information while learning trees. The key idea in our approaches to view privileged information as preferential advice - one that allows us to prefer some examples to share the same label. Consequently, these labels were used in guiding the learning algorithms (in our case tree-based learners), to learn better models. We derive the theory and empirically validate the effectiveness of the resulting learning approach both in the case of decision-trees and boosting. Our extensive experimental results demonstrate that the resulting methods outperform the SVM+.

Adapting this algorithm to more real-world data sets such as Electronic Health Records where a physician could provide privileged information for learning is a possible next step. Exploring the deeper connections to the other advice-based and non-traditional learning settings is an interesting theoretical research direction. Employing such approaches to other machine learning methods could be potentially interesting. Finally, extending this work to relational data where one could learn more powerful statistical relational learning models is an exciting direction of future research.

Acknowledgments

The authors thank Viktoriia Sharmanska for providing code for comparison. The authors also thank the Army Research Office (ARO) grant number W911NF-13-1-0432 under the Young Investigator Program and the CwC Program Contract W911NF-15-1-0461 with the US Defense Advanced Research Projects Agency (DARPA) and the ARO. Any opinions, findings and conclusion or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the DARPA, ARO or the US government.

References

[Altendorf *et al.*, 2005] Eric Altendorf, Angelo Restificar, and Thomas Dietterich. Learning from sparse data by exploiting monotonicity constraints. In *UAI*, 2005.

[Boutilier *et al.*, 2010] Craig Boutilier, Kevin Regan, and Paolo Viappiani. Simultaneous elicitation of preference features and utility. In *AAAI*, 2010.

[Boutilier, 2002] Craig Boutilier. A POMDP formulation of preference elicitation problems. In *AAAI*, pages 239–246, 2002.

[Chen *et al.*, 2012] Jixu Chen, Xiaoming Liu, and Siwei Lyu. Boosting with side information. In *Computer Vision - ACCV 2012 - 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I*, pages 563–577, 2012.

[Dhami, 2015] Devendra Dhami. Morphological classification of galaxies into spirals and non-spirals. Master's thesis, Indiana University, 2015.

[Dietterich *et al.*, 2008] Thomas Dietterich, Guohua Hao, and Adam Ashenfelter. Gradient tree boosting for training conditional random fields. *JMLR*, 9(10), 2008.

[Drummond and Boutilier, 2014] Joanna Drummond and Craig Boutilier. Preference elicitation and interview minimization in stable matchings. In *AAAI*, 2014.

[Friedman, 2001] Jerome Friedman. Greedy function approximation: A gradient boosting machine. In *Annals of Statistics*, 2001.

[Fung *et al.*, 2002] Glenn Fung, Olvi L. Mangasarian, and Jude W. Shavlik. Knowledge-Based support vector machine classifiers. In *NIPS*, pages 01–09, 2002.

[Hernández-Lobato *et al.*, 2014] Daniel Hernández-Lobato, Viktoriia Sharmanska, Kristian Kersting, Christoph H. Lampert, and Novi Quadrianto. Mind the nuisance: Gaussian process classification using privileged noise. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 837–845, 2014.

[Joachims, 1999] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999.

[Kunapuli *et al.*, 2013] Gautam Kunapuli, Phillip Odom, Jude Shavlik, and Sriraam Natarajan. Guiding autonomous agents to better behaviors through human advice. In *ICDM*, 2013.

[Lapin *et al.*, 2014] Maksim Lapin, Matthias Hein, and Bernt Schiele. Learning using privileged information: SV M+ and weighted SVM. *Neural Networks*, 53:95–108, 2014.

[Liang *et al.*, 2009] Lichen Liang, Feng Cai, and Vladimir Cherkassky. Predictive learning with structured (grouped) data. *Neural Networks*, 22(5-6):766–773, 2009.

[MacKay, 2003] David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge, 2003.

[Maclin *et al.*, 2005] Richard Maclin, Jude Shavlik, Lisa Torrey, Trevor Walker, and Edward Wild. Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *AAAI*, 2005.

[Natarajan *et al.*, 2012] Sriraam Natarajan, Tushar Khot, Kristian Kersting, Burnd Gutmann, and Jude Shavlik. Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning*, 86(1), 2012.

[Natarajan *et al.*, 2015] Sriraam Natarajan, Kristian Kersting, Tushar Khot, and Jude Shavlik. *Boosted Statistical Relational Learners: From Benchmarks to Data-Driven Medicine*. Springer, 2015.

[Odom *et al.*, 2015] Phillip Odom, Tushar Khot, Reid Porter, and Sriraam Natarajan. Knowledge-based probabilistic logic learning. In *AAAI*, 2015.

[Pechyony and Vapnik, 2010] Dmitry Pechyony and Vladimir Vapnik. On the theory of learnining with privileged information. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1894–1902, 2010.

[Settles, 2012] Burr Settles. *Active Learning*. Morgan & Claypool, 2012.

[Sharmanska *et al.*, 2013] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H. Lampert. Learning to rank using privileged information. In *CVPR*, 2013.

[Towell and Shavlik, 1994] G Towell and Jude Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 69:119–165, 1994.

[Vapnik and Vashist, 2009] V Vapnik and A Vashist. A new learning paradigm: Learning using privileged information. In *Neural Networks*, 2009.

[Willett *et al.*, 2013] Kyle W Willett, Chris J Lintott, Steven P Bamford, Karen L Masters, Brooke D Simmons, Kevin RV Casteels, Edward M Edmondson, Lucy F Fortson, Sugata Kaviraj, William C Keel, et al. Galaxy zoo 2: detailed morphological classifications for 304 122 galaxies from the sloan digital sky survey. *MNRAS*, 2013.

[Yang *et al.*, 2013] Shuo Yang, Tushar Khot, Kristian Kersting, and Sriraam Natarajan. Knowledge intensive learning: Combining qualitative constraints with causal independence for parameter learning in probabilistic models. In *ECML*, 2013.