DPPred: An Effective Prediction Framework with Concise Discriminative Patterns

Jingbo Shang, Meng Jiang, Wenzhu Tong, Jinfeng Xiao, Jian Peng, Jiawei Han, *Fellow, IEEE*, and the Pooled Resource Open-Access ALS Clinical Trials Consortium

Abstract—In the literature, two series of models have been proposed to address prediction problems including classification and regression. Simple models, such as generalized linear models, have ordinary performance but strong interpretability on a set of simple features. The other series, including tree-based models, organize numerical, categorical, and high dimensional features into a comprehensive structure with rich interpretable information in the data. In this paper, we propose a novel Discriminative Pattern-based Prediction framework (DPPred) to accomplish the prediction tasks by taking their advantages of both effectiveness and interpretability. Specifically, DPPred adopts the concise discriminative patterns that are on the prefix paths from the root to leaf nodes in the tree-based models. DPPred selects a limited number of the useful discriminative patterns by searching for the most effective pattern combination to fit generalized linear models. Extensive experiments show that in many scenarios, DPPred provides competitive accuracy with the state-of-the-art as well as the valuable interpretability for developers and experts. In particular, taking a clinical application dataset as a case study, our DPPred outperforms the baselines by using only 40 concise discriminative patterns out of a potentially exponentially large set of patterns.

Index Terms—Discriminative pattern, generalized linear model, tree-based models, classification, regression

1 Introduction

Accuracy and interpretability are two desired goals in predictive modeling, including both *classification* and *regression*. Previous work can be characterized into two lines. One line has ordinary performance with strong interpretability on a set of simple features, but meets a serious bottleneck when modeling complex high-order interactions between features, such as linear regression, logistic regression [18], and support vector machine [34]. The other line consists of models that are more often studied for their high accuracy, for example, tree-based models including random forest [2] and gradient boosted trees [16] as well as the neural network models [20], which model nonlinear relationships with high-order combinations of different features. However, their lower interpretability and high complexity prevent practitioners from deploying in practice [18]. In the real-world scientific and medical applications which require both intuitive understanding of the features and high accuracies, the practitioners are not satisfied with neither line of models, and thus, it is important and challenging to develop an effective prediction framework with high interpretability when dealing with high-order interactions with features.

Many pattern-based models have been proposed in the last decade to construct high-order patterns from the large set of features, including association rule-based methods on categorical data [6], [25], [29], [36], [38], [39] and frequent pattern-based algorithms on text data [24], [26] and graph data [8], [21]. Recently, a novel series of models, the discriminative pattern-based models [3], [4], have demonstrated their advantages over the traditional models. They prune non-discriminative patterns from the whole set of frequent patterns, however, the number of discriminative patterns used in their classification or regression models is still huge (at the magnitude of thousands). How to select concise discriminative patterns for better interpretability is still an open issue.

To address the above challenges, in this paper, we propose a novel discriminative patterns-based learning framework (DPPred) that extracts a concise set of discriminative patterns from high-order interactions among features for accurate classification and regression. In DPPred, first, we train tree-based models to generate a large set of high-order patterns. Second, we explore all prefix paths from root nodes to leaf nodes in the tree-based models as our discriminative patterns. Third, we compress the number of discriminative patterns by selecting the most effective pattern combinations that fit into a generalized linear model with high classification accuracy or small regression error. This component of fast and effective pattern extraction enables the strong predictability and interpretability of DPPred.

Intuitively speaking, DPPred selects the robust discriminative patterns in multi-tree based models by fitting them into a generalized linear model. Our extensive experiments demonstrate that DPPred achieves comparable or even better performance when competing with the traditional tree-based models. Besides the effectiveness, we want to highlight that

- *J. Shang, M. Jiang, W. Tong, J. Xiao, J. Peng, and J. Han are with the Department of Computer Science in University of Illinois at Urbana-Champaign, Champaign, IL 61820.*
E-mail: {shang7, mjiang89, wtong8, jxiao13, jianpeng, hanj}@illinois.edu.

Manuscript received 5 June 2016; revised 24 Aug. 2017; accepted 16 Sept. 2017. Date of publication 28 Sept. 2017; date of current version 1 June 2018.
(Corresponding author: Jingbo Shang.)
Recommended for acceptance by V.S.S. Tseng.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2757476

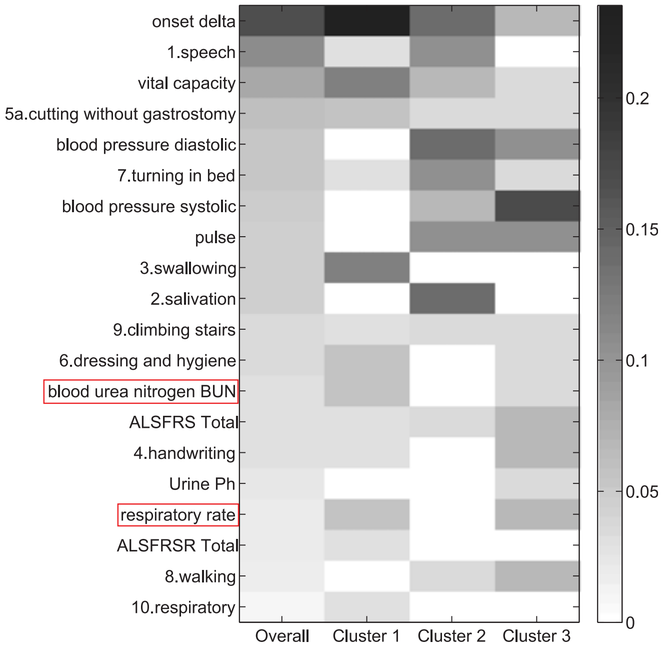


Fig. 1. *Two new important factors with the ALS disease that we found with* DPPred. Among the set of important clinical variables (rows) that DPPred discovered from the dataset of the Prize4Life Challenge 2012, two highlighted ones have later been experimentally verified that they have extremely high correlations with the ALS disease [17], [22], [40]. The columns are patient clusters.

our DPPred framework is applicable in the real-world tasks where the model storage and computational cost are highly restricted.

Discovering Robust Patterns in the Prize4Life Challenge. We apply DPPred to analyze the prognosis and perform stratification for Amyotrophic Lateral Sclerosis (ALS) patients on the public dataset from the DREAM-Phil Bowen ALS Prediction Prize4Life Challenge 2012. Our DPPred makes the following achievements.

- DPPred achieves a smaller error, a RMSE of 0.5306, than the method ranked at #7 with a RMSE of 0.5664. The RMSE of DPPred is less than 4 percent higher than the winner with a RMSE of 0.5113.
- The robust discriminative patterns found by our DPPred are well interpretable, while the other methods including the winner cannot interpret their performances. Note that our DPPred selects only 40 concise discriminative patterns involving 28 clinical variables from an exponentially large set, while other models used as many as 2 to 3 times variables.
- As show in Fig. 1, DPPred discovers two new important clinical factors, the Blood Urea Nitrogen (BUN) and the respiratory rate. These two factors were not found by the top teams in the Challenge but there is indirect experimental and logical evidence for their being actually worth further study [17], [22], [40]. Also, from the figure we can observe that each patient cluster generates different diagnosis patterns.

Our DPPred accurately predicts the ALS prognosis and systematically identifies clinically-relevant features for the ALS patient stratification in an interpretable manner. The distinct diagnosis patterns can significantly benefit the treatment of the ALS and precision medicine.

It is worthwhile to highlight the advantages of our proposed machine learning framework DPPred.

- *Interpretability.* DPPred learns a small number of robust discriminative patterns involving high-order interactions among original features.
- *Efficiency.* DPPred compresses multi tree-based models into a low-dimensional generalized linear model, making the online prediction extremely fast.
- *Effectiveness.* Experimental results on several real-world datasets demonstrate that DPPred has comparable or even better performances than the state-of-the-art models on the standard tasks of classification and regression.
- *Clinical pattern discovery.* DPPred has been successfully applied to discover patient clusters and crucial clinical signals for the Amyotrophic Lateral Sclerosis disease.

The remaining of this paper is organized as follows. In Section 2 we survey the related work. In Section 3 we provide the problem definition and our preliminary study. Section 4 presents our proposed DPPred framework and the details of its algorithms. Section 5 reports empirical results on synthetic and real-world datasets. Section 6 shows our discovery in the prognosis analytic for ALS patients. Section 7 concludes the study.

2 Related Work

In this section we review existing methods that are related to DPPred, including pattern-based classification models, tree-based models and pattern selection approaches.

2.1 Pattern-Based Classification

The philosophy of frequent pattern mining has been widely adopted to study the problem of pattern-based classification. CMAR [25] and CAEP [10] are classification methods based on multiple class-association rules and emerging patterns. Yin et al. extended it to CPAR based on predictive association rules [39]. Besides the association rules, direct discriminative pattern mining was proposed to generate effective performance [3], [4], [12]. However, these approaches have several serious issues. First, the huge number of frequent patterns leads to an expensive computational cost of pattern generation and selection. Second, the number of the selected patterns can be still as large as thousands, which limits the interpretability and causes the inefficiency of the classification model. Third, these models are not capable to address the regression tasks. Moreover, the discretization of continuous variables depends too much on parameter tuning to generate robust performances. Recently, Dong et al. proposed to utilize patterns in a different angle, where data are partitioned based on patterns, and local models are trained independently in different partitions [9]. Although this type of pattern aided models sheds lights on a different usage of patterns, the model still lacks interpretability, because the local models are trained in the original feature space. This work is based on our previous work [32], [33].

2.2 Tree-Based Models

Tree-based models are popular in the classification tasks. Both decision tree and boosted tree models are explainable but quite sensitive to the training data. Traditional ensemble methods using multiple trees, such as random forest [2] and gradient boosting decision trees [15], alleviate the over-fitting issue. Ren et al. showed that the global refinement could provide better performance because the growth and pruning processes in different trees are independent [31]. However, the increased model size of those multi-tree based models sacrifices the interpretability. Our proposed DPPred is different from this category of models.

There are post-pruning techniques for multi-tree based models to induce new feature spaces. Typically, they encoded each tree as a flat index list and each instance as a binary vector indexed by the trees [11], [19], [27], [30], [31]. Vens et al. transferred the binary vectors into an inner product kernel space using a support vector machine and showed the increase of classification accuracy [37]. Furthermore, pairwise interactions have also been studied to fit a two-layer-tree model for accurate classification and regression [28]. Though the number of features is reduced by pruning, the dimension of the newly-created feature space is still high due to a large number of constructed trees. For example, in [31], after many efforts on pruning, the model size of the pruned random forest was still at megabytes and thus the prediction was too slow to support real-time applications. Our experimental results will later show that DPPred delivers comparable results using as few as the top 20 discriminative patterns, which is substantially reduced even compared to the state-of-the-art models.

2.3 Pattern Selection

Simply selecting patterns with the highest independent heuristics such as information gain and Gini index is limited to very simple tasks due to the redundancy and over-fitting problems [23]. Given the labels, i.e., the types for classification or the real numbers for regression, LASSO [35] is widely used in feature selection tasks as well as forward selection [7]. Due to the relatively large number of candidate discriminative patterns, the backward selection is not suitable in our problem setting. Our proposed DPPred framework adopts the LASSO and forward selection methods to select discriminative patterns. Their performances have been compared and discussed in the experimental section.

3 Preliminaries

This section defines the problem as well as the important concepts used throughout this paper.

3.1 Problem Formulation

For a prediction task (classification or regression), the data is a set of n examples in a d-dimensional feature space together with their labels $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$, for $\forall i$ $(1 \leq i \leq n)$, $\mathbf{x}_i \in \mathbb{R}^d$. It is worth noting that the values in the example $\mathbf{x}_i$ can be either continuous (numerical) or discrete (categorical). As categorical features can be transformed into several binary dummy indicators, we can assume $\mathbf{x}_i \in \mathbb{R}^d$ without the loss of generality. The label y_i is either a class (type) indicator or a real number depending on the specific task. In previous pattern-based models, e.g., DDPMine [4], patterns are extracted from categorical values and thus they are only able to handle the continuous variables after careful manual discretization, which is tricky and often requires prior knowledge about the data.

The goal of our proposed framework DPPred is to learn a concise model that consists of a small set of discriminative patterns from the training data, which learns and predicts the examples as accurately as possible, i.e., predict the correct class indicator in *classification* tasks and predict close to the true number in *regression* tasks. Formally, given a dataset $\mathcal{D}$, DPPred returns a set of k discriminative patterns $\mathcal{P}$ using a generalized linear model $f(\cdot)$ that minimizes $\sum_{i=1}^{n} l(f(M(\mathbf{x}_i)), y_i)$, where $l(\cdot,\cdot)$ is the general loss function, $M(\cdot)$ is a mapping function that maps the original feature vector $\mathbf{x}$ to the pattern space using patterns $\mathcal{P}$.

DPPred generates a pool of discriminative patterns within a reasonable size, and selects top-k patterns based on their learning performance on training data, using a generalized linear learning model. Since the number of selected patterns is very limited, these patterns are able to provide informative interpretability with reasonable predictive power. In addition, for the coming testing data, by evaluating only a very small set of the selected discriminative patterns, DPPred is enabled to make predictions with a generalized linear model efficiently.

3.2 Definition

First, we define a series of concepts to derive the discriminative patterns. Traditional frequent pattern mining works on categorical data and itemset data, in which discretization is required to deal with continuous variables. Instead of roughly discretizing the numerical values, we adopt the thresholding boolean function in DPPred.

Definition 1. Condition *is a thresholding boolean function on a specific feature dimension. The condition is in the form of* $(x_{\cdot,j} < v)$ *or* $(x_{\cdot,j} \geq v)$, *where* j *indicates the specific dimension and* v *is the threshold value. The relational operator in a condition is either* $<$ *or* $\geq$. *For any dimension* j *in features corresponding to binary indicators, we restrict* v *to be 0.5.*

Note that the threshold values in DPPred are not specified by users beforehand. In previous pattern-based models, e.g., DDPMine [4], the practitioners have to discretize values of continuous variables prior to pattern mining. DPPred automatically determines these values in the tree model, completely based on the training data without any human intervention.

Example 1. Suppose $\mathbf{x}_i \in \mathbb{R}^{10}$, one possible condition is that $\mathbf{x}_{\cdot,1} < 0.5$. Another example could be $\mathbf{x}_{\cdot,2} \geq 0.8$.

We define a pattern as a set of conditions. Formally, we use conjunctions to concatenate different conditions: it is consistent with the prefix path in the decision tree that represents the conjunction of the conditions in the nodes along the path.

Definition 2. Pattern *is a conjunction clause of conditions on specific feature dimensions. Formally, it is defined as follows:*

$$(x_{\cdot,j_1} < v_1) \wedge (x_{\cdot,j_2} \geq v_2) \wedge \ldots \wedge (x_{\cdot,j_m} \geq v_m),$$

where m *is the number of conditions within this pattern. Different patterns are allowed to have different* m *values.*

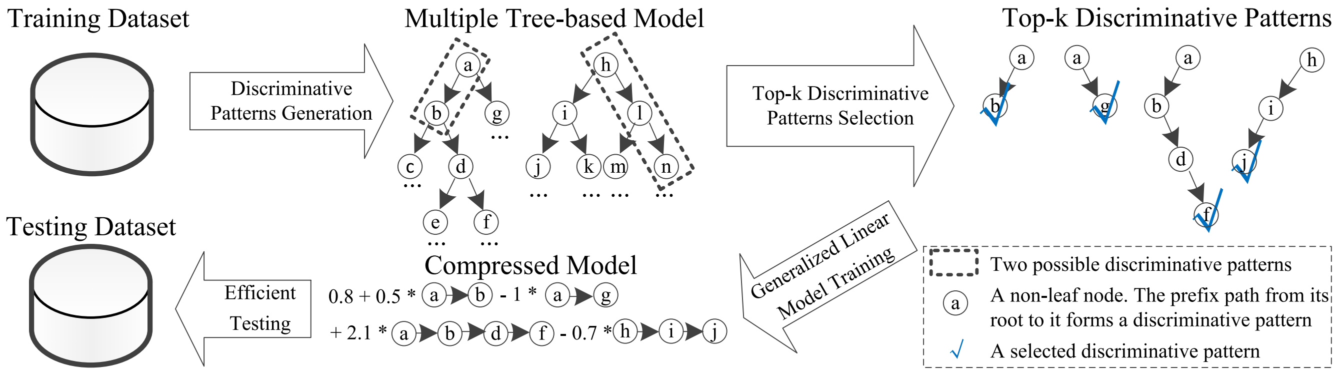

Training Dataset
Discriminative Patterns Generation
Multiple Tree-based Model
a
b
g
c
d
e
f
h
i
l
j
k
m
n
Top-k Discriminative Patterns Selection
Top-k Discriminative Patterns
Testing Dataset
Efficient Testing
Compressed Model
0.8 + 0.5 * a → b - 1 * a → g
+ 2.1 * a → b → d → f - 0.7 * h → i → j
Generalized Linear Model Training
Two possible discriminative patterns
A non-leaf node. The prefix path from its root to it forms a discriminative pattern
✓ A selected discriminative pattern

Fig. 2. *The overview of our* DPPred *framework.* With the training data, the multi-tree based model (e.g., random forest) is trained for discriminative pattern generation. For each tree, all prefix paths from its root to non-leaf nodes are treated as discriminative patterns. After a large pool of discriminative patterns is generated, DPPred conducts top-k pattern selection to identify the most informative and interpretable patterns. Here, k is typically small (20 or 30). Finally, it trains a generalized linear model based on the 2^k pattern space representation.

Example 2. Suppose $\mathbf{x}_i \in \mathbb{R}^{10}$, one possible pattern is that $(\mathbf{x}_{\cdot,1} < 18) \wedge (\mathbf{x}_{\cdot,3} \geq 100) \wedge (\mathbf{x}_{\cdot,9} < 0.5)$.

Now we define discriminative patterns as follows.

Definition 3. Discriminative Patterns *refer to those patterns which have strong signals on the learning tasks, given the labels of data. For example, a pattern with very high information gain on the classification training data, or a pattern with very small mean square error on the regression training data, is a discriminative pattern.*

Example 3. Suppose $\mathbf{x}_i \in \mathbb{R}^{10}$ and the labels are generated as follows:

$$y_i = [(\mathbf{x}_{i,1} \geq 1) \wedge (\mathbf{x}_{i,2} < 0)] \vee [(\mathbf{x}_{i,1} < 18) \wedge (\mathbf{x}_{i,3} \geq 100)].$$

Both patterns $(\mathbf{x}_{i,1} \geq 1) \wedge (\mathbf{x}_{i,2} < 0)$ and $(\mathbf{x}_{i,1} < 18) \wedge (\mathbf{x}_{i,3} \geq 100)$ are two of the most discriminative patterns. Similar patterns that contain or have overlaps with these two patterns are also discriminative patterns.

Discriminative patterns have overlapped predictive effects. Specifically, a few discriminative patterns are special cases of other patterns. For example, in the previous example, both patterns $(\mathbf{x}_{i,1} \geq 1) \wedge (\mathbf{x}_{i,2} < 0)$ and $(\mathbf{x}_{i,1} \geq 1) \wedge (\mathbf{x}_{i,2} < 0) \wedge (\mathbf{x}_{i,3} < 0)$ indicate a positive label. However, the second pattern only encodes a subset of data points that the first pattern encodes, and thus, it does not provide extra information for the learning process. This common phenomenon shows that roughly taking the top discriminative patterns based on independent heuristics wastes the budget of the number of patterns, when the linear combination of these patterns is not synergistic. Therefore, our DPPred selects the top-k patterns by their predictive performance to make the selected patterns complementary and compact.

Definition 4. Top-k Patterns *are formalized as a size-k subset of discriminative patterns, which has the best performance (i.e., the highest accuracy in classification tasks or the least rooted mean square error in regression tasks) based on the training data.*

Here we assume that the training and testing data share the same distribution, which is widely acknowledged in the classification and regression problems. In this case, the accuracy on the testing data is approaching the accuracy on the training data and our model is able to alleviate the over-fitting issue.

Example 4. In the last example, the top-2 patterns are $\{(\mathbf{x}_{i,1} \geq 1) \wedge (\mathbf{x}_{i,2} < 0), (\mathbf{x}_{i,1} < 18) \wedge (\mathbf{x}_{i,3} \geq 100)\}$.

4 Our DPPred Framework

This section first presents the overview of DPPred and then introduces the details of every component in this framework as well as the theoretical time complexity.

4.1 The Overview of DPPred

Fig. 2 presents the overview of our DPPred framework. First it learns a constrained multi-tree based model with the training data. By adopting every prefix path from the root of a tree to any of its non-leaf nodes as a discriminative pattern, a large pool of discriminative patterns is ready for further top-k discriminative pattern selection. Two different solutions, forward selection and LASSO, are utilized to select top-k discriminative patterns based on their performances using a generalized linear model. Both solutions have shown high accuracies in the experiments. The corresponding linear model with the selected top-k discriminative patterns is adopted to make predictions on new examples. Our DPPred is extremely fast and memory-efficient.

4.2 Discriminative Pattern Generation

The first component in the DPPred framework is the generation of high-quality discriminative patterns, as shown in Algorithm 1. We use *tree bag* to refer the set of instances falling into a specific node in the decision tree. The random decision tree [2] introduces the randomness via bootstrapping training data, randomly selecting features and splitting values when dividing a large tree bag into two smaller ones. There are various ways to handle continuous variables, such as [13]. In our implementation, once the splitting feature is selected, all mid-points between any two consecutive feature values after sorting are considered as potential splitting points. T random decision trees are generated, and for each tree, all prefix paths from its root to non-leaf nodes are treated as discriminative patterns. Due to the predictivity of decision trees, so-generated patterns are highly effective in the specific prediction task. Note that the decision tree is built with different loss functions in different tasks, which could be entropy gain in classification tasks or mean square error in regression tasks.

Algorithm 1. Discriminative Pattern Generation

```
Require: $n$ training instances $(\mathbf{x}_i, y_i)$, the number of trees $T$, the depth threshold $D$, and minimum tree bag size $\sigma$
Return: a set of discriminative patterns for further selection.
$\mathcal{P} \leftarrow \emptyset$
for $t = 1$ to $T$
    Build a random decision tree [2] with maximum depth $D$ and minimum tree bag size $\sigma$.
    for each non-leaf node $u$
        $\mathcal{P} \leftarrow \mathcal{P} \cup \{root \rightarrow u\}$
return $\mathcal{P}$
```

In real-world datasets, the discriminative patterns are frequently emerging, and the length of such patterns are not too long. Specifically, we assume that the number of instances satisfying a given discriminative pattern should be at least σ, and the length of discriminative patterns is no more than D. The returned patterns are discriminative to ensure prediction accuracy and diverse to ensure sufficient condition coverage. As one of the most famous multi-tree based models, random forest [2] is the best fit addressing all the requirements if we treat every prefix path from the root of a tree to its non-leaf node as a discriminative pattern. First, distributions of labels of instances in a tree bag always have low entropy. Therefore, the patterns are discriminative on the training data. Second, it provides many putative patterns from various random decision trees trained on different bootstrapped datasets. Third, the depth threshold D and the minimum tree bag size σ can be naturally added as constraints during the growth of trees.

Applying feature reduction techniques in the preprocessing can shrink the search space of patterns, however, some features, which are only useful when combined into long patterns, may be pruned out. Therefore, we leave it to users as an option in their pre-processing.

4.3 Pattern Space Construction

After the pattern generation, DPPred maps the instances in the original feature space to a new pattern space using the set of discriminative patterns discovered by tree models, as shown in Algorithm 2. For each discriminative pattern, there is one corresponding binary dimension describing whether the instances satisfy the pattern or not. Because the dimension of the pattern space is equal to the number of discriminative patterns which is a very large number after the generation phase, we need to further select a limited number of patterns and thus make the pattern space small and efficient. It is also worth a mention that this mapping process is able to be fully parallelized for speedup.

Algorithm 2. Pattern Space Construction

```
Require: $n$ instances $(\mathbf{x}_i)$, a discriminative patterns set $\mathcal{P}$
Return: $n$ instances in pattern space $(\mathbf{x'}_i)$
for $i = 1$ to $n$ do
    $\mathbf{x'}_i \leftarrow \mathbf{0}$
    for $j$-th pattern $P_j$ in $\mathcal{P}$ do
        if $\mathbf{x}_i$ satisfies pattern $P_j$ then
            $\mathbf{x'}_{i,j} \leftarrow 1$
return $(\mathbf{x'}_i)$
```

4.4 Top-k Pattern Selection

After a large pool of discriminative patterns is generated, further top-k selection needs to be done to identify the most informative and interpretable patterns. A naive way is to use heuristic functions, such as information gain and Gini index, to evaluate the significance of different patterns on the prediction task and choose the top-ranked patterns. However, the effects of top-ranked patterns based on the simple heuristic scores may have a large portion of overlaps and thus their combination does not work optimally. Therefore, to achieve the best performance and find complementary patterns, we propose two effective solutions: forward selection and LASSO, which make decisions based on the effects of the pattern combinations instead of considering different patterns independently.

4.4.1 Forward Pattern Selection

Instead of an exhausted search of all possible combinations of k discriminative patterns, forward selection gradually adds the discriminative patterns one by one while each newly added discriminative pattern is the best choice at that time [7], which provides an efficient approximation of the exhausted search. To be more specific, when the first k' discriminative patterns are fixed, the algorithm empirically adds one more discriminative pattern so that the new set of $k' + 1$ patterns achieves the best training performance in the generalized linear model, as shown in Algorithm 3. As mentioned before, when assuming training and testing data have the same distribution, using training accuracy is very reasonable.

Algorithm 3. Top-k Pattern Selection: Forward

```
Require: $n$ training examples $(\mathbf{x}_i, y_i)$, a set of discriminative patterns $\mathcal{P}$ and $k$
Return: top-$k$ discriminative patterns set $\mathcal{P}_k$ and a generalized linear model $f(\cdot)$
$\mathcal{P}_k \leftarrow \emptyset$
for $t = 1$ to $k$ do
    for each pattern $p$ in $\mathcal{P}$ do
        $\mathbf{x'} \leftarrow$ construct pattern space($\mathbf{x}, \mathcal{P}_k \cup \{p\}$) using Algorithm 2
        $g(\cdot) \leftarrow$ a generalized linear model [34] on $(\mathbf{x'}_i, y_i)$
        $per_p \leftarrow g(\cdot)$'s training performance
    $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup \{\arg\max_p per_p\}$
$\mathbf{x'} \leftarrow$ construct pattern space($\mathbf{x}, \mathcal{P}_k$)
$f(\cdot) \leftarrow$ a generalized linear model on $(\mathbf{x'}_i, y_i)$
return $\mathcal{P}_k$, $f(\cdot)$
```

4.4.2 LASSO Based Pattern Selection

L1 regularization (i.e., LASSO [35]) is designed to make the weight vector sparse by tuning a non-negative parameter λ, where the features with non-zero weight will be the selected ones. Since we are actually selecting features in the pattern space, for a given λ, we optimize the following loss function to get a subset of important patterns

$$\mathcal{L} = \sum_{i}^{n} l(\mathbf{x'}_i^T \mathbf{w}, y_i) + \lambda \cdot \|w\|_1, \tag{1}$$

where $\mathbf{x'}_i$ is the mapped binary feature representation in pattern space of ith example; $\mathbf{w}$ is the weight vector in the generalized linear model; $l(\cdot, \cdot)$ is a general loss function such as logistic loss. To ensure there are at most k patterns having non-zero weights in the pattern space, we should carefully choose a value for λ. We assume that there exists hidden importance among the features. Therefore, we propose an empirical assumption that holds in many real-world cases: if the weight of a feature is non-zero in a given $\lambda = v$, it is also non-zero for any smaller $\lambda < v$. In practice, we can verify this assumption by a coarse-grained grid search for λ. When the assumption is true, a binary search algorithm is shown in Algorithm 4. Otherwise, as the backup, we will apply the fine-grained grid search for an appropriate λ. The LASSO implementation in GLMNET [14] is adopted in this thesis, whose loss function is the cross entropy.

Algorithm 4. Top-k Pattern Selection: LASSO

```
Require: $n$ training examples $(\mathbf{x}_i, y_i)$, a set of discriminative patterns $\mathcal{P}$, $k$, and a small value $\epsilon$
Return: top-$k$ discriminative patterns $P_i$ and a generalized linear model $f(\cdot)$
$\mathcal{P}_k \leftarrow \emptyset$
$l \leftarrow 0, r \leftarrow +\infty$
$\mathbf{x}' \leftarrow$ construct pattern space($\mathbf{x}, \mathcal{P}$) using Algorithm 2
while $l + \epsilon < r$ do
    $\lambda \leftarrow (l + r)/2$
    $\mathbf{w} \leftarrow \arg\min_{\mathbf{w}}$ Equation (1)
    if non-zero weighted patterns $\leq k$ then
        $\mathcal{P}_k \leftarrow \{p | p\text{'s weight is non-zero}\}$
        $r \leftarrow \lambda$
    else
        $l \leftarrow \lambda$
$\mathbf{x}' \leftarrow$ construct pattern space($\mathbf{x}, \mathcal{P}_k$)
$f(\cdot) \leftarrow$ a generalized linear model on $(\mathbf{x}'_i, y_i)$
return $\mathcal{P}_k$, $f(\cdot)$
```

4.5 Prediction

Once the top-k discriminative patterns are determined, for any upcoming new test instance, DPPred first maps it into the learned pattern space, and then applies the pre-trained generalized linear model to compute the prediction, as shown in Algorithm 5. As the number of patterns is limited, both the mapping into the pattern space and the prediction of the generalized linear model will be extremely fast.

Algorithm 5. Prediction

```
Require: $n$ testing examples $(\mathbf{x}_i)$, top-$k$ discriminative patterns set $\mathcal{P}_k$, and the generalized linear model $f(\cdot)$
Return: predictions of testing instances $\hat{y}_i$
$\mathbf{x}' \leftarrow$ construct pattern space($\mathbf{x}, \mathcal{P}_k$) using Algorithm 2
for $i = 1$ to $n$ do
    $\hat{y}_i \leftarrow f(\mathbf{x}'_i)$
return $\hat{y}$
```

4.6 Time Complexity Analysis

To build up a single random decision tree with depth threshold D and minimum tree bag size σ, by assuming both numbers of random features and random partitions are small and fixed constants, the time complexity is $O(nD)$, because the total number of instances on each level of the tree is n. Therefore, the time complexity of generating T trees is $O(TnD)$ in the generation step.

For the selection step, the complexity is mainly determined by the number of discriminative patterns induced by T random decision trees, which is dependent on the total number of non-leaf nodes. As the maximum depth of a single tree is D, there is an upper bound on the number of leaf nodes 2^D. Starting from the tree bag size, the number of leaf nodes should be no more than $\lceil \frac{n}{\sigma} \rceil$. Since the trees here are all binary trees, the number of leaf nodes is one more than the number of non-leaf nodes. Therefore, the number of discriminative patterns $|\mathcal{P}|$ (i.e., the number of non-leaf nodes) is bounded by $T \cdot \min\{2^D, \lceil \frac{n}{\sigma} \rceil\} - 1$. If we solve logistic regression and LASSO using (sub-)gradient descent algorithm, and thus the time complexity per gradient step is only linear to the dimension of features and the number of examples. The time complexity is proportional to $O(|\mathcal{P}| \cdot n \cdot k^2)$ if the forward selection is used, while it is proportional to $O(n \cdot k \cdot |\mathcal{P}|)$ if the LASSO is used. By assuming the numbers of iterations to converge are similar in the LASSO and the forward selection, the LASSO will be a little more efficient than the forward selection.

When predicting new test instances, one can easily figure out the bottleneck is mapping instances into the learned pattern space. Therefore, in the batch mode where examples are considered together, the time complexity is $O(n \cdot k \cdot D)$. In the streaming (or online) mode where instances come one by one, the time complexity is $O(k \cdot D)$, where k is the number of discriminative patterns and D is the maximum tree depth, which is equivalent to the maximum number of conditions in a single pattern.

It is worth mentioning that all modules can be fully parallelized, leading to further speedup in practice.

5 Experiments

In this section, we conduct extensive experiments to demonstrate the interpretability, efficiency, and effectiveness of our proposed DPPred framework. The implementation of DPPred and datasets are released in the author's Github, including both the new version[1] and its previous binary classification version.[2]

5.1 Experimental Settings

This section presents the datasets, baseline methods, and learning tasks in our experiments.

5.1.1 Datasets

First, we generate synthetic datasets where the features are demographics and lab test results of patients and the label is whether the patient has a disease, in order to demonstrate the interpretability of DPPred. Assuming doctors can diagnose the disease using some rules based on these information, it can be verified whether the top discriminative patterns selected by DPPred are consistent with the actual diagnosing rules.

Several real-world classification and regression datasets from UCI Machine Learning Repository are used in the experiments, as shown in Table 1 with statistics of the number of instances and the number of features. In the datasets

[1] https://github.com/shangjingbo1226/DPPred
[2] https://github.com/shangjingbo1226/DPClass

TABLE 1
The Statistics of Our Real-World Datasets from UCI Machine Learning Repository for Classification and Regression

Type	Dataset	# Instances	# Dimensions	Variable type
Classification	Adult	45,222	14	Mixed
	Hypo	3,772	19	Mixed
	Sick	3,772	19	Mixed
	Chess	28,056	6	Mixed
	Crx	690	15	Mixed
	Sonar	208	60	Numeric
High dimension	Nomao	29,104	120	Mixed
	Musk	7,074	166	Numeric
	Madelon	1,300	500	Numeric
Regression	Bike	17,379	10	Mixed
	Parkinsons	5,875	16	Numeric
	Crime	1,994	99	Numeric

adult, *hypo* and *sick*, the ratio of standard train/test splitting is $2:1$. Therefore, for the other classification and regression datasets, we divide the datasets into train/test $(2:1)$ by unbiased sampling as preprocessing.

For classification tasks, to compare with DDPMine, we use the same datasets including *adult*, *hypo*, *sick*, *crx*, *sonar*, *chess*, *waveform*, and *mushroom*. Because both DDPMine and DPPred achieve almost perfect accuracy (very close to 100 percent) on the datasets *waveform* and *mushroom*, these two datasets are omitted. In addition, the performance of DPPred on high-dimensional datasets (*nomao*, *musk* and *madelon* datasets) is also investigated, since DDPMine performs poorly on high-dimensional data. The metric is the accuracy on the testing data: higher accuracy means better performance.

For regression datasets, we choose general datasets such as *bike* and *crime*, as well as clinical datasets where patterns are more likely to be present, such as *parkinsons*. Furthermore, to make the errors in different datasets comparable, min-max normalization is adopted to scale the continuous labels into $[0, 1]$. The metric is the rooted mean square error (RMSE) on the testing data: a lower RMSE means better performance.

5.1.2 Baseline Methods

DDPMine [4] is a previous state-of-the-art discriminative pattern-based algorithm. It first discretizes the continuous variables such that frequent pattern mining algorithm could be applied. Using frequent and discriminative patterns, new feature space is constructed and any classical classifiers could be further utilized. DDPMine only focuses on classification tasks and it is not applicable in regression experiments.

Random Forest (RF) [2] is another baseline method using same parameters as those in the random forest used in DPPred, except for D. There is no limit on the depth in RF. Moreover, we are interested in the limited-depth random forest model (LRF) built in the top-k generation step of global patterns. These two tree-based methods are capable in both classification and regression tasks. It is expected if these two complex models (i.e., hard to interpret) have slightly better performance than DPPred, because the major contributions of DPPred are the concise interpretable patterns instead of solely the accuracy. To make a fair comparison, Decision Tree (DT) with a similar number of nodes with DPPred is also listed as a baseline.

TABLE 2
Model Complexity and Time Complexity

Model	Model complexity (# Patterns)	Time complexity
DPPred	$k \approx 20 \sim 50$	$O(k \cdot D)$
DT	# of nodes ≈ 64	$O(D)$
DDPMine	$k \approx 100 \sim 1{,}000$	$O(k \cdot D)$
LRF	# of nodes $\approx 6{,}400$	$O(T \cdot D)$
RF	# of nodes $\geq 10{,}000$	$O(T \cdot D)$

Model complexity is measured by the number of encoded patterns. Here, D is the number of dimensions, k is the number of top patterns, and T is the number of trees.

5.1.3 Classification and Regression Tasks

In DPPred, for the classification tasks, the default parameter setting is $T = 100, D = 6, \sigma = 10, k = 20$. For the regression tasks, because the continuous labels are more complex than those discrete class labels in classification, it is natural to incorporate more patterns. Therefore, the default setting is $T = 100, D = 6, \sigma = 10, k = 30$.

We will show results using both forward selection (DPPred-F) and LASSO (DPPred-L) to select the top-k discriminative patterns. We deeply study the impact of the parameters such as the number of selected discriminative patterns k and the number of trees in the random forest T. Therefore, we fix the other parameters as their default values and vary the parameter value to study their impacts, respectively.

5.2 Efficiency and Interpretability

Efficiency. The test running time is linearly proportional to the model complexity, which is related to the number of patterns the model used. In the experiments, DDPMine needs 100 to 1,000 patterns while DPPred only needs 20, which indicates a significant reduction of prediction runtime. Moreover, the random forest without any constraints will contain more than 10,000 nodes (i.e., patterns), which is far more expensive. Although the evaluation of random forest for a single testing instance will traverse only a number of nodes equals to the sum of depths in different trees, it always needs more than 1,000 traverses in the experiments. Therefore, DPPred is the most efficient model for testing new instances, compared to DDPMine and random forest, by achieving about *20 to 50 times speedup* in practice. Furthermore, DPPred could be fully parallelized for further speedup. The empirical results are presented in Table 2.

Interpretability: our discovery of interpretable patterns. We generate a small medical dataset for binary classification to demonstrate the interpretability. For each patient, we draw several uniformly sampled features as follows:

1) Age (A): positive integers no more than 60.
2) Gender (G): male or female.
3) Lab Test 1 (LT1): blood types (categorical values) from {A, B, O, AB}.
4) Lab Test 2 (LT2): continuous values in $[0, 1]$.

Totally, there are 10^5 random patients for training and $5 \cdot 10^4$ patients for testing.

TABLE 3
Test Accuracy on Classification Datasets

Dataset	adult	hypo	sick	crx	sonar	chess
DPPred-F	**85.66%**	**99.58%**	98.35%	**89.35%**	**85.29%**	92.25%
DPPred-L	84.33%	99.28%	**98.87%**	87.96%	83.82%	92.05%
DT	83.33%	92.90%	93.82%	77.78%	67.65%	89.86%
DDPMine	83.42%	92.69%	93.82%	87.96%	73.53%	90.04%
LRF	83.51%	95.78%	93.93%	**89.35%**	83.82%	90.04%
RF	85.45%	97.22%	94.03%	**89.35%**	83.82%	**94.22%**

DDPMine *outperforms Decision Tree and Support Vector Machine on all these datasets.* DPPred *performs best on almost every dataset, while RF is the best on the chess dataset.*

TABLE 4
Testing RMSE on Regression Datasets

Dataset	bike	crime	parkinsons	Diff
DPPred-F	0.0872	0.1515	0.1969	N/A
DPPred-L	0.0974	0.1465	0.1951	N/A
DT	0.1186	0.1971	0.2129	+24.74%
LRF	0.1211	0.1367	0.1976	+16.64%
RF	**0.0836**	0.**1372**	**0.1865**	−6.77%

Min-max Normalization is adopted to scale the Continuous Labels into $[0, 1]$. DPPred *takes much fewer patterns than RF and perform significantly better than DT and LRF.*

The positive label of the disease is assigned to a patient if at least one of the following rules holds:

1) $(A > 18)$ and $(G = \text{Male})$ and $(LT1 = AB)$ and $(LT2 \geq 0.6)$,
2) $(A > 18)$ and $(G = \text{Female})$ and $(LT1 = O)$ and $(LT2 \geq 0.5)$,
3) $(A \leq 18)$ and $(LT2 \geq 0.9)$.

To make the classification tasks more challenging, 0.1 percent noise is added to the training data. That is, 0.1 percent labels in training will be flipped.

We apply both DPPred-F and DPPred-L on this dataset. Both give the test accuracy 99.99 percent. The top-3 discriminative patterns found in both DPPred-F and DPPred-L are listed as below. We observe that the found patterns are quite close to the groundtruth rules. We demonstrate that the selected discriminative patterns provide a high-quality explanation:

1) $(A > 18)$ and $(G = \text{Female})$ and $(LT1 = O)$ and $(LT2 \geq 0.496)$,
2) $(A \leq 18)$ and $(LT2 \geq 0.900)$,
3) $(A > 18)$ and $(G = \text{Male})$ and $(LT1 = AB)$ and $(LT2 \geq 0.601)$.

We apply DDPMine to this dataset but its accuracy is only 95.64 percent, because the discretization brings too much noise. The top-3 patterns mined by DDPMine are as follows, which are quite different from expectation:

1) $(LT2 > 0.8)$,
2) $(G = \text{Male})$ and $(LT1 = AB)$ and $(LT2 \geq 0.6)$ and $(LT2 < 0.8)$,
3) $(G = \text{Female})$ and $(LT1 = O)$ and $(LT2 \geq 0.6)$ and $(LT2 < 0.8)$.

The interpretability has also been verified on a real-world cardiopulmonary patients' health status dataset. Clinicians provide positive feedback after verifying the top-30 discriminative patterns as well as the variable frequencies in these patterns [5].

5.3 Effectiveness in Classification

DDPMine is a previous state-of-the-art pattern-based classification method, which outperforms traditional classification models including decision tree and support vector machine [3], [4]. We compare DPPred, DDPMine and RF on the same datasets used in DDPMine. The results are shown in Table 3. DPPred-F and DPPred-L always have higher accuracy over DDPMine. An important reason of this advantage is that the candidate patterns generated by tree-based models in DPPred are much more discriminative and thus more effective on the specific classification task than those frequent but less useful patterns extracted in DDPMine. Except for *sick* dataset, DPPred-F has the highest accuracy, while DPPred-L works best on *sick* dataset. It seems that DPPred-F works a little better than DPPred-L. However, their results are quite close to each other and are both better than those of DDPMine on most datasets.

More surprisingly, DPPred demonstrates even better performance than the complex model random forest on several datasets, while its accuracies on other datasets are still comparable with RF, which is due to the effectiveness of the pattern selection module where we select the optimal pattern combination instead of selecting patterns independently. This shows that the proposed model is very effective in classification tasks while it is highly concise and interpretable.

5.4 Effectiveness in Regression

Since DDPMine is not applicable on regression tasks, we only compare DPPred with DT, RF, and LRF. Note that these two methods are highly complicated and thus preserve very limited interpretability. The RMSE results and the average differences compared to DPPred are shown in Table 4.

Unlike the results in classification datasets, complex models outperform DPPred on all datasets although the difference is not very significant. This is reasonable because, different from the discrete class labels, the real-valued prediction increases the level of difficulty. Although we have raised the number of top patterns a little, bag-of-patterns feature representations based on a small number of patterns still have some limitations to predict a real value. For example, there are at most 2^{30} different examples in the constructed pattern space, which means there are at most 2^{30} different predicted values, but infinite real numbers are likely to be the true value for a new example. However, it is worth noting that DPPred (both DPPred-F and DPPred-L) always achieves comparable performance with RF, and work better than or similar to DT and LRF, which still demonstrates the effectiveness of DPPred to some extent while the model is more compact and interpretable than RF and LRF.

5.5 Effectiveness in High Dimensions

We are interested in high-dimensional datasets (i.e., at least 100 dimensions) because DDPMine is not effective in large dimensional data. To compare with DDPMine, we use classification datasets whose number of dimensions is at least 100 and no regression datasets are used. As the dimension of the original feature space grows, it is reasonable to increase the depth threshold D, as well as the number of

TABLE 5
Testing Accuracy on High-Dimensional Datasets

Dataset	nomao	musk	madelon
DPPred-F	97.17%	95.92%	74.50%
DPPred-L	96.94%	95.71%	**76.00%**
DT	92.98%	87.82%	50.34%
DDPMine	96.83%	93.29%	59.83%
LRF	95.56%	90.49%	59.17%
RF	**97.86%**	**96.60%**	56.50%

DPPred *performs consistently better than* DDPMine, *and it is comparable with the Complex RF and better on Madelon.*

trees T, to involve higher order interactions and increase the number of candidate discriminative patterns. Therefore, we set $D = 10$ and $T = 200$. Meanwhile, the dimension of mapped pattern space may also need to be increased due to the higher complexity of problems. As a result, we set $k = 50$ in *nomao* and *musk* datasets. However, we kept $k = 20$ in *madelon* dataset because many features are noises.

As shown in Table 5, DPPred can always outperform DDPMine and generate comparable results to those by RF. It is worth noting that in *madelon* dataset, DPPred-F and DPPred-L outperform RF significantly. As stated before, *madelon* is highly noisy. As a result, many patterns generated by random forest are not that reliable, which can be very poor at test data although they are discriminative in training data. On the other hand, DPPred compresses the patterns and only keeps the most discriminative ones, and thus alleviates this problem to some extent. This demonstrates the robustness of DPPred especially when the features are high-dimensional and noisy. It is also worth a mention that the training process of DPPred is at least 10 times faster than DDPMine in these datasets.

5.6 Parameter Analysis

In this section, we study the number of top patterns k and the number of trees in the random forest T.

5.6.1 The Number of Top Discriminative Patterns

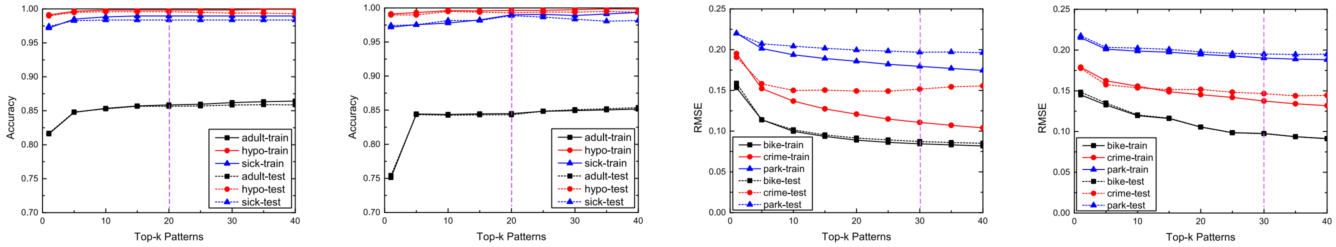
The most interesting parameter in DPPred is k, the number of discriminative patterns used in the final generalized linear model. It controls the model size of the generalized linear model used for prediction and thus affects its efficiency. Because the default value of k is 20 for classification tasks and 30 for regression tasks and its effectiveness has been proved in previous experiments, we vary k from 1 to 40 to see the trends of both training and testing accuracies on different datasets. Three representative classification datasets (*adult*, *hypo*, and *sick*) and three regression datasets (*bike*, *crime* and *parkinsons*) are used in this experiment.

As illustrated in Fig. 3, the performance on test data is always following the trend of performance on training data and the performance is increasing as k grows in both classification and regression tasks (accuracy is increasing on classification datasets while the error is decreasing on regression datasets). The discrepancy between training and test performance is more significant in regression tasks (right two in Fig. 3), which is reasonable due to the higher complexity of the problem, but the trends are quite similar. In addition, we argue that the larger difference could be caused by the insufficient size of training data, because the curves always overlap on *bike* dataset that is much bigger than the other two. It is also worth noting that DPPred-L performs more consistently than DPPred-F, especially in regression tasks, as a result of λ which is automatically learned in DPPred-L but is manually specified in DPPred-F. In summary, the similar trends in training and test data justify that our pattern selection based on training accuracy is reasonable. In real-world applications, k could be determined by cross validations.

Although the performance is becoming better almost all the time, it slows down much when k is greater than the default value. This is true for both classification and regression tasks. An even larger k will hurt the efficiency of both training process and online prediction, and might introduce overfitting issues in prediction (e.g., test accuracy on hypo dataset is 99.58 percent when $k = 20$ while it becomes 99.28 percent when $k = 40$ using forward selection). Therefore, we can conclude that a very small k (e.g., $k = 20$) is enough for these comprehensive real-world datasets, which further proves that the proposed DPPred can compress the model into a very tiny size while its accuracy remains comparable.

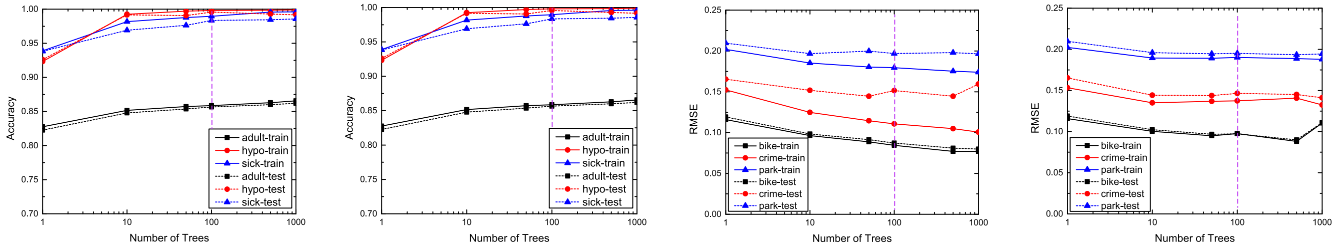
5.6.2 The Number of Trees in the Model

Another important parameter in DPPred is the number of trees needed to generate the large pool of discriminative patterns. As mentioned before, a single tree is not enough to generate that many patterns, and thus there is strong motivation to try $T = 1$ as an extreme case. The default value 100 works well in previous experiments, and thus we vary T in {1, 10, 50, 100, 500, 1,000} to see the trends of both training and testing accuracies. As before, three datasets for classification and regression tasks are presented in the experiments.



(a) Classification: DPPred-F (b) Classification: DPPred-L (c) Regression: DPPred-F (d) Regression: DPPred-L

Fig. 3. The impact of top-k patterns in classification and regression tasks. Training and testing performances are almost overlapped in some datasets. We observe that a small number of patterns (e.g., 20 for classification and 30 for regression) are enough to achieve stable performance.



(a) Classification: DPPred-F (b) Classification: DPPred-L (c) Regression: DPPred-F (d) Regression: DPPred-L

Fig. 4. The impact of the number of trees in classification tasks. Training and testing performances are almost overlapped. We can observe that a small number of trees (e.g., 100) are enough to achieve stable performance.

Fig. 4 visualizes the results on classification and regression datasets respectively. When $T = 1$, the performance is much lower than others, which means only a single decision tree is not enough for a diverse patterns pool. Too few trees generally cannot guarantee high coverage of effective patterns, especially when the dataset is large and the dimension is high. Increasing number of trees leads to a better diversity of candidate patterns. According to the curves, one can easily observe and conclude that the performance remains stable as long as the number of trees is sufficiently large, and a reasonably large T is enough to achieve a satisfying result. Similar to the number of patterns k, however, many noisy patterns will be generated if T becomes too large, which fit training data better while fail to characterize testing data and are harmful to generalization of the model (e.g., test RMSE is 0.0977 on hypo dataset when $T = 100$ while it becomes 0.1104 when $T = 1{,}000$ using LASSO). In addition, the more trees we have, the larger number of pattern candidates will be generated, which increases the time complexity of feature selection. T is by default set to 100 in our experiments, which performs consistently well on different datasets.

6 Novel Marker Discovery for ALS Patient Stratification

Unlike other diseases such as many cancers, which can be clearly classified into subtypes with distinct survival rates, no significant signals have been identified to explain the diverse survival times (ranging from less than a year to over 10 years) for ALS patients. Such a wide range makes it difficult to predict disease progression and survival, and suggests rather large underlying disease heterogeneity. There may exist different subgroups of patients, each having its unique disease causes and prognosis.

6.1 ALS Dataset

To solve this puzzle, the Pooled Resource Open-Access ALS Clinical Trials (PRO-ACT) platform[3] was created by Prize4-Life and the Neurological Clinical Research Institute at Massachusetts General Hospital to collect ALS data from existing completed ALS clinical trials. In 2012, a subset of PRO-ACT data was constructed with the aim to crowd-source the challenge of ALS prognosis as a data mining task, which is known as the DREAM-Phil Bowen ALS Prediction Prize4Life Challenge (“the 2012 challenge” for short in this section) [22].

The 2012 challenge aimed at improving the prediction of ALS progression rate, which is essentially a regression task. The participants built models with a training set of 918 patients, and submitted their models to the challenge organizers. The organizers ran the models on a separate leaderboard set of 279 patients and provided feedback on model performance to the participants. Several such submission-and-feedback cycles were run in 3 months, and then the last submissions from the participants were evaluated and ranked by the organizers on another separate validation set of 627 patients.

This challenge attracted more than 1,000 participants and received 37 unique algorithms during the submission-and-feedback leaderboard phase. Among them, only six algorithms demonstrated improved accuracy over the baseline (developed by the challenge organizers) on the final validation dataset.

The best prognosis model (“the Top Solution” for short in this section) developed in the 2012 challenge, which uses Bayesian trees with 484 predictive features constructed from 26 clinical variables, is a profound success. It has predicted ALS progression from clinical data better than clinicians do, and can potentially reduce the cost of future ALS trials by \$6-million [22]. The Top Solution is not perfect though. It is a uniform model for all patients and thus lacks the ability to make the personalized diagnosis. Also, it is hard to clinically interpret the Top Solution due to the high model complexity.

For a fair comparison, DPPred has been trained and evaluated in such a way that mimics the 2012 challenge. Training was performed with the same training set of 918 patients and evaluation was on the same validation set of 627 patients. The leaderboard set of 279 patients was used merely for feature calibration (described later).

The data used in the 2012 challenge consist of 2 parts: clinical variables and the actual ALS progression rate (which serves as the golden standard for model comparison). Available clinical variables of a patient can be grouped into 5 kinds: demographic information, vital signs, lab test results, family disease history and the Amyotrophic Lateral Sclerosis Functional Rating Scale (ALSFRS). A detailed description of the data can be found in the supplement of [22]. Some variables are excluded from our study because their units are not consistent for some patients.

ALSFRS is a quantitative clinical score ranging from 0 to 40 for evaluating the functional status of an ALS patient. It consists of 10 assessments of motor functioning, each evaluated within the range 0 (worst status, no function) to 4

[3] The data in the PRO-ACT Database are contributed by members of PRO-ACT Consortium, founded in 2011 by Prize4Life and the Northeast ALS Consortium with the funding from the ALS Therapy Alliance.

(normal function). Those 10 evaluated functions[4] are: "*1.speech*", "*2.salivation*", "*3.swallowing*", "*4.handwriting*", "*5.cutting food and handling utensils*" (with or without gastrostomy), "*6.dressing and hygiene*", "*7.turning in bed and adjusting bed clothes*", "*8.walking*", "*9.climbing stairs*" and "*10.respiratory*".

The rate of change in ALSFRS[5] with respect to time T (ΔALSFRS/ΔT) can be used as a quantitative measurement of ALS progression rate. The task is to predict ΔALSFRS/ΔT within 3 to 12 months from disease onset, given the clinical variables within the first 3 months. The RMSE between the predicted ΔALSFRS/ΔT and the actual value is used to evaluate the predictive performance.

6.2 Data Processing

The clinical variables about a patient contain 3 data types: static categorical, static continuous and longitudinal continuous variables. Static variables are time-independent, while longitudinal variables are measured multiple times for each patient and are likely to change over time. Any static categorical variable with k categories is replaced with k+1 binary features where the additional one indicates whether the variable is missing. A static continuous variable is simply a continuous feature.

Each longitudinal continuous variable $\{\mathbf{x}, \mathbf{t}\}$, where $\mathbf{x} \in \mathbb{R}^n$ is the n measured values and $\mathbf{t} \in \mathbb{R}^n$ is the times of n measurements in ascending order, is converted to 12 continuous features by taking some statistics of $\{\mathbf{x}, \mathbf{t}\}$ and a derivative sequence $\Delta \in \mathbb{R}^{n-1}$ whose ith element is defined as $\Delta_i = (x_{i+1} - x_i)/(t_{i+1} - t_i)$. 6 statistics are taken from $\mathbf{x}$: the average value $(\sum_{i=1}^n x_i)/n$, the first-measured value x_1, the last-measured value x_n, the maximum $\max_i\{x_i\}$, the minimum $\min_i\{x_i\}$, and the standard deviation $\sigma(x_i)$. Another 6 statistics are taken similarly from Δ.

After performing such variable conversion separately on the training, leaderboard and validation sets, features are calibrated across all 3 sets so that features completely missing in at least 1 of the 3 datasets are discarded. The number of features we finally feed into DPPred is 498, converted from 78 clinical variables.

6.3 Task Description

In the precision medicine setting, we assume there are some implicit groupings underlying the patients, such as the subtypes of a certain disease. Formally, we define the patient cluster as follows.

Definition 5. Diagnosis-Stratified Patient Clusters *are G disjoint patient groups, such that patients within the same group are similar and there are different top-k patterns of clinical variables across clusters that suggests distinct diagnoses. We use* patient cluster *for short.*

Considering different patient sets $\mathcal{S}$, we can define the global and local patterns respectively.

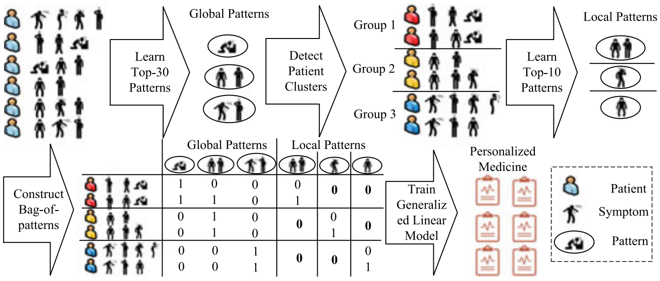


Fig. 5. Overview of the prognosis analysis and stratification for ALS patients. Based on the global patterns discovered by DPPred, diagnosis-stratified patient clusters are detected by clustering algorithm (e.g., LDA). Then, DPPred explores local patterns within a certain patient cluster. Last, a generalized linear model based on both global and local patterns is ready for testing data.

Definition 6. Global Patterns *are the top-K_g patterns by using all patients as training instances.*

The global patterns are expected to not only capture the general properties of the specific task, but also hopefully find the way to detect implicit groups of patients. For example, suppose a disease has 3 different subtypes, we expect some global patterns can handle the general diagnosis while others can help clinicians partition patients into the 3 subtypes.

Definition 7. Local Patterns *are the top-K_l patterns by using only the patients in a single patient clusteras training instances.*

Within different patient clusters (e.g., different subtypes of a disease), patients may have different root causes, and thus need different diagnoses and treatments. Therefore, we are motivated to discover local patterns.

In this application, our task is to first discover global patterns for all patients and then figure out the patient clusters as well as the local patterns in each patient cluster. The goal is to demonstrate that our DPPred can not only accurately predict ALS prognosis, but also systematically identify clinically-relevant features for ALS patient stratification in an interpretable manner, which will further facilitate personalized diagnosis and therapy.

6.4 DPPred for ALS Patient Stratification

As shown in Fig. 5, the prognosis analysis and stratification for ALS patients work as follows.

- Discover K_g global patterns based on all patients;
- Partition patients into G different patient clusters based on the discovered global patterns;
- Discover K_l local patterns inside each patient cluster;
- Construct the bag-of-patterns feature representation for each patient based on global patterns and only the local patterns discovered in his/her patient cluster;
- Train a generalized linear model based on the constructed features.

When a new patient comes, it is predicted as follows.

- Assign a patient clusterbased on K_g global patterns;
- Evaluate the corresponding K_l local patterns in the assigned patient cluster;
- Construct the bag-of-patterns feature representations based on these $K_g + K_l$ discriminative patterns;
- Predict by the generalized linear model.

[4] Some patients are evaluated instead with a modified version ALSFRS-R ranging from 0 to 48, where "*10.respiratory*" is replaced with "*R1.Dyspnea*", "*R2.Orthopnea*", and "*R3.Respiratory insufficiency*", each ranging from 0 to 4.

[5] To assure the consistency of scales across patients, for those patients with ALSFRS-R only but no ALSFRS, the sum of questions 1-9 and $R1$ are used in the calculation of the rate of change.

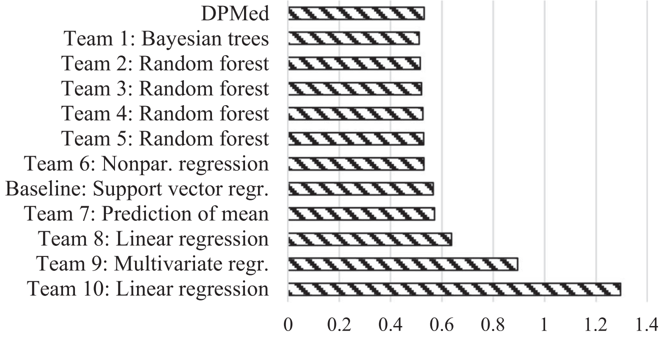


Fig. 6. Testing RMSE in the 2012 challenge. DPPred is comparable to the Top Solution (only < 4 percent away) and the other top-ranked algorithms (complicated and not interpretable). DPPred only relies on 28 clinical variables, forming a small subset of all 78 variables.

We utilize DPPred to discover global and local patterns. Since it is a regression task, similar to our previous experiments, we set $T = 100, D = 6, \sigma = 10, K_g = 30, K_l = 10$. Therefore, for each patient, we have $K_g + K_l = 40$ patterns. For the patient clustering, by making analogy from bag-of-words to bag-of-patterns, we adopt Latent Dirichlet Allocation (LDA) algorithm [1]. The number of patient clusters is determined by the domain knowledge and the cross-validation. First of all, based on the domain knowledge, although the number of patient clusters is unknown, it is usually not too large. Therefore, we can assume it is between 1 and 10. Then, based on the training data, we run a 10-fold cross validation under the certain number of patient clusters. The results suggest that $G = 3$ is a good choice. More specifically, observing global patterns of patients, in order to detect patient clusters, we design a generative process of the patterns incorporating patient clusters as latent variables. First, we assume the patterns in a particular patient cluster follow a multinomial distribution, where a pattern is a random variable drawn from a prior Dirichlet distribution. Inspired by bag-of-words, we use bag-of-patterns to represent observed patterns of patients, therefore, can apply LDA.

6.5 Results and Discussion

DPPred has obtained a predictive performance comparable to the Top Solution while gives interpretable discriminative patterns, as shown in Fig. 6. DPPred with 3 patient clusters achieves a RMSE of 0.5306 on the validation dataset, which is only < 4 percent away from the RMSE of the Top Solution, 0.5113, comparable to the other top-ranked algorithms which are also complicated and not interpretable, and better than the baseline RMSE, 0.5664. The linear combination of discriminative patterns trained with DPPred includes 28 clinical variables in total, which is a small subset of all 78 available variables.

Our top 20 most frequent clinical variable list (Fig. 1) reveals the importance of the blood urea nitrogen (BUN) and the respiratory rate, which are not among the most important features reported by any of the top 5 teams nor the organizers of the 2012 challenge. The other variables in our top 20 list agree well with the 2012 challenge findings. Some examples include the critical role of the onset delta (i.e., the time between the ALS onset and the first time the patient was tested in a trial), mouth-related ALSFRS assessments (including "*1.speech*", "*2. salivation*" and "*3.swallowing*") and vital capacity. A high degree of consistency with the 2012 challenge results proves the reliability of DPPred, while our newly reported important variables highlight the power of feature selection in DPPred and shed new light on ALS research.

There are other reasons to take our newly discovered important clinical variables seriously when designing future studies. It has been experimentally shown that the BUN level is elevated ($p < 0.05$) when minocycline, a drug that can delay the progression of ALS, is applied [17], [40]. Therefore the correlation between the BUN level and the ALS progression rate is likely to be true. The respiratory rate reflects respiratory muscle functioning and thus related to "*10.respiratory*", 1 of the 10 assessments in ALSFRS. Since the importance of "*10.respiratory*" is reported by several among the top 5 teams in the 2012 challenge [22] and also by DPPred, it should not be surprising that the respiratory rate is also in the list. Interestingly DPPred is the only algorithm among those that simultaneously selects both the respiratory rate and "*10.respiratory*".

Another point worth mentioning is the distinct local patterns of each patient cluster displayed in Fig. 1, indicating different diagnosis patterns across patient clusters. For example, the mouth-functioning-related scores are important overall but not locally in Cluster 3, while the blood pressure is important in Patient Clusters 2 & 3 but plays a less significant role in Cluster 1. Such distinct diagnosis patterns may not only aid personalized medicine but also shed light on the mechanism, underlying heterogeneity, and treatment of ALS. For the reference purpose, we also trained a DPPred model without clustering, and its RMSE, 0.5404, is worse.

All these results indicate that our DPPred not only accurately predicts ALS prognosis, but also systematically identifies clinically-relevant features for ALS patient stratification in an interpretable manner, which will facilitate personalized diagnosis and therapy.

7 Conclusions

In this paper, we propose an effective and concise discriminative pattern-based prediction framework (DPPred) to address the classification and regression problems and provide high interpretability with a small number of discriminative patterns. Specifically, DPPred first trains a constrained multi-tree model using training data and then extracts the prefix paths from root nodes to non-leaf nodes in all the trees as candidate discriminative patterns. The size of discriminative patterns is compressed by selecting the most effective pattern combinations according to their predictive performance in a generalized linear model. Instead of selecting the patterns independently using heuristics, DPPred finds the best combination using forward selection or LASSO, which avoids the overlapping effect between similar patterns. Extensive experiments demonstrate that DPPred is able to model high-order interactions and present a small number of interpretable patterns to help human experts understand the data. DPPred provides comparable or even better performance than the state-of-the-art model DDPMine and random forest model in classification and regression. DPPred has been successfully applied to discover patient clusters and crucial clinical signals for the amyotrophic lateral sclerosis disease.

ACKNOWLEDGMENTS

Research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), US National Science Foundation IIS 16-18481, IIS 17-04532, and IIS 17-41317, grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov), and a Google PhD Fellowship. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies of the U.S. Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. Data used in the preparation of this article were obtained from the Pooled Resource Open-Access ALS Clinical Trials (PRO-ACT) Database. As such, the following organizations and individuals within the PRO-ACT Consortium contributed to the design and implementation of the PRO-ACT Database and/or provided data, but did not participate in the analysis of the data or the writing of this report: (1) Neurological Clinical Research Institute, MGH; (2) Northeast ALS Consortium; (2) Novartis; (3) Prize4Life; (4) Regeneron Pharmaceuticals, Inc.; (5) Sanofi; and (6) Teva Pharmaceutical Industries, Ltd.

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.

[2] C. Chen, A. Liaw, and L. Breiman, "Using random forest to learn imbalanced data," University of California, Berkeley, 2004.

[3] H. Cheng, X. Yan, J. Han, and C.-W. Hsu, "Discriminative frequent pattern analysis for effective classification," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, 2007, pp. 716–725.

[4] H. Cheng, X. Yan, J. Han, and P. S. Yu, "Direct discriminative pattern mining for effective classification," in *Proc. IEEE 24th Int. Conf. Data Eng.*, 2008, pp. 169–178.

[5] Q. Cheng, J. Shang, J. Juen, J. Han, and B. Schatz, "Mining discriminative patterns to predict health status for cardiopulmonary patients," in *Proc. 7th ACM Conf. Bioinf. Comput. Biol. Health Inform.*, 2016, pp. 41–49.

[6] G. Cong, K.-L. Tan, A. K. Tung, and X. Xu, "Mining top-k covering rule groups for gene expression data," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2005, pp. 670–681.

[7] S. Derksen and H. Keselman, "Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables," *Brit. J. Math. Statistical Psychology*, vol. 45, no. 2, pp. 177–341, 1992.

[8] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis, "Frequent substructure-based approaches for classifying chemical compounds," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 8, pp. 1036–1050, Aug. 2005.

[9] G. Dong and V. Taslimitehrani, "Pattern aided classification," in *Proc. 2016 SIAM Int. Conf. Data Mining*, 2016, pp. 225–233.

[10] G. Dong, X. Zhang, L. Wong, and J. Li, "CAEP: Classification by aggregating emerging patterns," in *Proc. Int. Conf. Discovery Sci.*, 1999, pp. 30–42.

[11] T. Ebina, H. Toh, and Y. Kuroda, "DROP: An SVM domain linker predictor trained with optimal features selected by random forest," *Bioinf.*, vol. 27, no. 4, pp. 487–494, 2011.

[12] W. Fan, et al., "Direct mining of discriminative and essential frequent patterns via model-based search tree," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 230–238.

[13] U. M. Fayyad and K. B. Irani, "On the handling of continuous-valued attributes in decision tree generation," *Mach. Learn.*, vol. 8, no. 1, pp. 87–102, 1992.

[14] J. Friedman, T. Hastie, and R. Tibshirani, "Glmnet: Lasso and elastic-net regularized generalized linear models," *R package version*, vol. 1, no. 4, 2009.

[15] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232. 2001.

[16] Y. Ganjisaffar, R. Caruana, and C. V. Lopes, "Bagging gradient-boosted trees for high precision, low variance ranking models," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inform. Retrieval*, 2011, pp. 85–94.

[17] P. Gordon, et al., "Placebo-controlled phase I/II studies of minocycline in amyotrophic lateral sclerosis," *Neurology*, vol. 62, no. 10, pp. 1845–1847, 2004.

[18] D. W. Hosmer Jr and S. Lemeshow, *Appl. Logistic Regression*, Hoboken, NJ, USA: Wiley, 2004.

[19] M. Kobetski and J. Sullivan, "Discriminative tree-based feature mapping," *Intell.*, vol. 34, no. 3, 2011.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[21] T. Kudo, E. Maeda, and Y. Matsumoto, "An application of boosting to graph classification," in *Proc. 17th Int. Conf. Neural Inf. Process. Syst.*, 2004, pp. 729–736.

[22] R. Küffner, et al., "Crowdsourced analysis of clinical trial data to predict amyotrophic lateral sclerosis progression," *Nature Biotechnology*, vol. 33, no. 1, pp. 51–57, 2015.

[23] C. Lee and G. G. Lee, "Information gain and divergence-based feature selection for machine learning-based text categorization," *Inform. Process. Manag.*, vol. 42, no. 1, pp. 155–165, 2006.

[24] C. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classification," *Biocomputing 2002*, World Scientific, pp. 564–575, 2001.

[25] W. Li, J. Han, and J. Pei, "CMAR: Accurate and efficient classification based on multiple class-association rules," in *Proc. IEEE Int. Conf. Data Mining*, 2001, pp. 369–376.

[26] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *J. Mach. Learn. Res.*, vol. 2, pp. 419–444, 2002.

[27] Y. Lou, R. Caruana, and J. Gehrke, "Intelligible models for classification and regression," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 150–158.

[28] Y. Lou, R. Caruana, J. Gehrke, and G. Hooker, "Accurate intelligible models with pairwise interactions," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 623–631.

[29] B. Liu, W. Hsu, and Y. Ma, "Integrating classification and association rule mining," in *Proc. 4th Int. Conf. Knowl. Discovery Data Mining*, 1998, pp. 80–86.

[30] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *Proc. 20th Annu. Conf. Neural Inf. Process. Syst.*, 2007, pp. .985–992.

[31] S. Ren, X. Cao, Y. Wei, and J. Sun, "Global refinement of random forest," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 723–730.

[32] J. Shang, "DPPred: An effective prediction framework with concise discriminative patterns and its biomedical applications," 2017.

[33] J. Shang, W. Tong, J. Peng, and J. Han, "DPClass: An effective but concise discriminative patterns-based classification framework," in *Proc. SIAM Int. Conf. Data Mining*, 2016, pp. 567–575.

[34] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[35] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc.. Series B (Methodological)*, vol. 58, pp. 267–288, 1996.

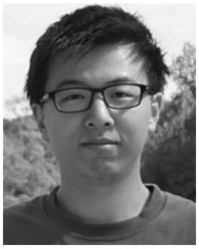
[36] A. Veloso, W. Meira, and M. J. Zaki, "Lazy associative classification," in *Proc. 6th IEEE Int. Conf. Data Mining*, 2006, pp. 645–654.

[37] C. Vens and F. Costa, "Random forest based feature induction," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 744–753.

[38] J. Wang and G. Karypis, "HARMONY: Efficiently mining the best rules for classification," in *Proc. 2005 SIAM Int. Conf. Data Mining*, pp. 205–216, 2005.

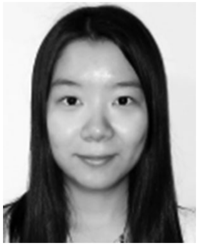
[39] X. Yin and J. Han, "CPAR: Classification based on predictive association rules," in *Proc. 2003 SIAM Int. Conf. Data Mining*, pp. 331–335, 2003.

[40] S. Zhu, et al., "Minocycline inhibits cytochrome c release and delays progression of amyotrophic lateral sclerosis in mice," *Nature*, vol. 417, no. 6884, pp. 74–78, 2002.

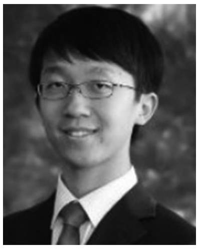


Jingbo Shang is working toward the PhD degree in the Department of Computer Science, University of Illinois, Urbana-Champaign. His research focuses on mining and constructing structured knowledge from massive text corpora. He is the recipient of the Computer Science Excellence Scholarship and Grand Prize of the Yelp Dataset Challenge in 2015. He received the Google PhD Fellowship in Structured Data and Database Management in 2017.

Meng Jiang received the BE and the PhD degrees from the Department of Computer Science and Technology, Tsinghua University, in 2010 and 2015, respectively. He is now an assistant professor in the Department of Computer Science and Engineering, University of Notre Dame. He worked as a postdoctoral research associate with the University of Illinois, Urbana-Champaign, from 2015 to 2017. He has published more than 20 papers on behavior modeling and information extraction in top conferences and journals of the relevant field such as the *IEEE Transactions on Knowledge and Data Engineering*, ACM Special Interest Group on Knowledge Discovery in Data, AAAI, ACM CIKM and IEEE ICDM. He also has delivered six tutorials on the same topics in major conferences. He was the best paper finalist in ACM SIGKDD 2014.



Wenzhu Tong received the BE degree from the Department of Computer Science and Technology, Wuhan University, China, and the MS degree from the Computer Science Department, University of Illinois, Urbana-Champaign, in 2014 and 2016, respectively. She is now working at Google.



Jinfeng Xiao received the BS degree in physics and mathematics from the Hong Kong University of Science and Technology, in 2014. He is now working toward the graduate degree in computer science with the University of Illinois, Urbana-Champaign. His current research lies at the interface of machine learning, data mining, and biomedical science.

Jian Peng received the BS degree in computer science from Wuhan University, and the PhD degree in computer science from the Toyota Technological Institute, Chicago, in 2013. He is an assistant professor in the Department of Computer Science, University of Illinois, Urbana-Champaign. Before joining the University of Illinois in 2015, he was a postdoc in the Berger Lab at MIT, and a visiting scientist in the Lindquist Lab, the Whitehead Institute for Biomedical Research. He was a student in the Xu Lab from 2007. He worked on HIV protein analysis with Drs. David Heckerman and Jonathan Carlson in the eScience group at Microsoft Research in 2010.

Jiawei Han is Abel Bliss professor in the Department of Computer Science, University of Illinois. He has been researching into data mining, information network analysis, and database systems, with more than 600 publications. He served as the founding editor-in-chief of the *ACM Transactions on Knowledge Discovery from Data*. He has received the ACM SIGKDD Innovation Award (2004), the IEEE Computer Society Technical Achievement Award (2005), the IEEE Computer Society W. Wallace McDowell Award (2009), and Daniel C. Drucker Eminent Faculty Award at UIUC (2011). He is a fellow of ACM and a fellow of the IEEE. He is currently the director of the Information Network Academic Research Center (INARC) supported by the Network Science-Collaborative Technology Alliance (NS-CTA) program of the U.S. Army Research Lab. His co-authored textbook *Data Mining: Concepts and Techniques* (Morgan Kaufmann) has been adopted worldwide.