ADAPTIVE FEATURE SPLIT SELECTION FOR CO-TRAINING: APPLICATION TO TIRE IRREGULAR WEAR CLASSIFICATION

Wei Du, Ronald Phlypo, and Tülay Adalı

Department of CSEE, University of Maryland, Baltimore County, MD 21250, USA

ABSTRACT

Co-training is a practical and powerful semi-supervised learning method. It yields high classification accuracy with a training data set containing only a small set of labeled data. Successful performance in co-training requires two important conditions on the features: diversity and sufficiency. In this paper, we propose a novel mutual information (MI) based approach inspired by the idea of dependent component analysis (DCA) to achieve feature splits that are maximally independent between-subsets (diversity) or within-subsets (sufficiency). We evaluate the relationship between the classification performance and the relative importance of the two conditions. Experimental results on actual tire data indicate that compared to diversity, sufficiency has a more significant impact on their classification accuracy. Further results show that co-training with feature splits obtained by the MI-based approach yields higher accuracy than supervised classification and significantly higher when using a small set of labeled training data.

Index Terms— Co-training, semi-supervised classification, feature splits, DCA, LTM tire data

1. INTRODUCTION

Semi-supervised learning has recently attracted much attention in the machine learning field. It is designed to achieve high classification accuracy with reduced effort from experienced human annotators, since only a small size of labeled training data is required. Among several semi-supervised learning methods [1–3], co-training, as a data-driven method, provides a practical and powerful approach for real-world problems. Co-training is based on the training of two classifiers, each using a subset of the features. First, both classifiers are trained on the available labeled data. Then, the unlabeled data samples with the most confident predictions in one classifier are cross-fed to the other classifier as newly labeled samples on which the training stages can re-iterate. Success in co-training is guaranteed under two important conditions [4]:

Diversity: Features can be split into two sets that are conditionally independent given the class;
Sufficiency: Each subset of features attributed to a single classifier is sufficient to train a good classifier.

Generally, we could achieve high classification accuracy when one of the aforementioned conditions is met while the other is only weakly satisfied [5, 6]. However, in many cases, it is difficult to find sufficiently powerful features that naturally split into two sets. Hence, features are often split into subsets based on heuristic criteria [7–9]. To the best of our knowledge, this is the first time feature splits are obtained based on either the diversity or the sufficiency condition for co-training, allowing to investigate their respective influence on the performance.

We propose a novel mutual information (MI) based approach inspired by the idea of dependent component analysis (DCA) [10, 11] to successfully address the problem of feature split selection. The MI-based method automatically constructs hierarchical clusters and achieves feature splits with maximal between- or within-classifier independence that satisfies diversity or sufficiency. Experimental results on laser tread mapping (LTM) tire data indicate that among the two conditions, sufficiency has a more significant impact on the classification accuracy than diversity. Further results show that co-training with feature splits obtained by the MI-based approach yields significantly higher classification accuracy than supervised learning when only few labeled training data are available.

2. DATA AND PREPROCESSING

2.1. Data set

We analyze LTM data obtained from 22 tires at specific mileages in their service life. Laser mapping is used to measure the progress of the tread wear. To obtain the LTM data, a single point conical laser is used to scan the surface of the tire at 1mm lateral spacing and 4140 points per single wheel revolution (360 degrees) and measures the distance to the tire surface along the normal to the tread surface. Tires with irregular wear (IW)—i.e., non-uniform or uneven wear patterns, resulting in locally depressed regions—should be labeled as a bad tire. Example tire images exhibiting IW are shown in

![Fig. 1](image-url)
2.2. Data preprocessing

We implement five preprocessing steps. We pay special attention not to introduce any bias to the subsequent analysis stages, thus preserving the information as much as possible. The preprocessing steps are shown in Figure 2, which are rib detection, polynomial detrending, outlier elimination, segmentation, and data smoothing. (1) **Rib detection**: We are interested in studying the IW as it manifests itself on the ribs. Each tire has different positions of grooves and ribs as shown in Figure 1 (b). We retain samples on ribs and discard samples corresponding to grooves. For further analysis, we also combine sacrificial ribs with their neighboring ribs to be able to analyze the former for possible IW. (2) **Polynomial detrending**: For each rib, we select points within two times the standard deviation (std) of samples and compute the regression polynomials for the mean section. We use a second-order polynomial as the trend of a given tire rib to avoid matching points in grooves and IW. We then subtract the trends from each section of the given tire data and obtain flattened tire data. (3) **Outlier elimination**: We detect and remove outliers from the flattened data. After we locate several ribs in the first step, there still exist outliers belonging to grooves. Those typical outliers on the edges of a rib are eliminated by a Grubbs test [12]. (4) **Segmentation**: To obtain more homogeneous samples, we are interested in classifying small units of tires as shown in Figure 3 (d). Thus, we cut each tire into patches according to the general contact length between a tire and the road, then each patch into several ribs and each rib into two halves. (5) **Data smoothing**: We smooth sample images by 2D median filtering. Since IW is expected to be smooth and to have a reasonably large area, we apply 2D median filtering using a window size of 7-by-7 to reduce noise and preserve edges of IW. The classification samples obtained from the 4th step are normalized to have zero mean and unit variance. Thus, IW—usually observed as a depression of the tire—now is associated mainly with negative values.

### 3. FEATURE EXTRACTION

The aim of the classification task is to detect samples with IW patterns. For each sample, we extract and select 14 relevant features to distinguish between the two groups. Some of our features are straightforward, such as the minimum and mean of negative values for a given sample, and the Euclidean distance from the test sample to the mean of good training samples. Other features are defined as follows.

#### 3.1. KPCA-LDA

Kernel principal component analysis (KPCA) maps the input data into a higher dimension space, called the feature space, by using a non-linear mapping and then applies linear PCA in this feature space [13]. First, for the training matrix \( X = [x_1, \ldots, x_n]^T \), we define an \( n \)-by-\( n \) matrix \( K \) with entries \( k(x_i, x_j) \), where \( i, j = 1, \ldots, n \) and \( k(x_i, x_j) \) is the kernel representation. In this paper, we use a Gaussian kernel. Second, we obtain the \( m \) largest positive eigenvalues and the corresponding normalized eigenvectors through the eigen-decomposition of the kernel matrix \( K \). The dimension \( m \) is selected automatically according to the gap in the eigen spectrum instead of using a fixed number [14]. Third, the KPCA transformed feature is calculated by \( y = \sum_{i=1}^{m} \beta_i k(x, x_i) \), where \( \beta_i = [\beta_1, \beta_2, \ldots, \beta_m]^T \) and \( \beta_m \) is the \( i \)-th entry of the \( m \)-th largest eigenvector.

Linear discriminant analysis (LDA) aims to achieve an optimal linear dimensionality reduction [15, 16]. According to Fisher’s criterion \( J(w) = w^T S_B w (w^T S_W w)^{-1} \), where \( S_B \) and \( S_W \) are the between-class and within-class scatter matrices, respectively. The solution is obtained as the largest eigenvector of the matrix \( S_B^{-1} S_W \). In this paper, we use the optimal linear dimensionality reduction to reduce the dimensionality of the feature space.
is the between-class covariance matrix and $S_W$ is the total within-class covariance matrix. We can find a linear combination $w$ of features that provides a balance between maximum class compactness and class separability by maximizing $J(w)$, where $w$ is the generalized eigenvector of $(S_W, S_B)$ corresponding to the largest generalized eigenvalue. Then the KPCA-LDA feature is obtained by $f = w^T y$, where $y$ is the output of KPCA.

3.2. KL divergence and global statistics
The Kullback-Leibler (KL) divergence is a non-symmetric measure of the information-theoretic distance between two probability distributions [17, 18]. For probability mass functions $p$ and $q$ of a discrete random variable, their KL divergence is defined as $D_{KL}(p \parallel q) = \sum_i p(i) \log \frac{p(i)}{q(i)}$ which is non-negative and $D_{KL}(p \parallel q) = 0$ if and only if $p = q$.

From visual inspection, good and bad samples have significant differences between their probability distributions as shown in Figure 4. Hence, we calculate KL divergence between histograms of the test sample and the selected good/bad samples by kernel density estimate. The $x$-axis shows the range of the data and $y$-axis shows the corresponding density values.

4. CO-TRAINING

4.1. Co-training algorithm
The goal of co-training is to learn a classification mapping from the training set including labeled and unlabeled data. Each classifier is initialized using only the typically few available labeled examples. At every iteration of co-training, each classifier chooses a set of unlabeled examples to add to the training set. The selected set includes those with the highest classification confidence provided by the other classifier. Then, each classifier learns from their augmented labeled set, and the process repeats. The intuition behind the co-training algorithm is that one classifier adds examples to the labeled set that the other classifier will then be able to use successfully for learning [21].

We implement several classifiers to perform co-training, including Naive Bayes (NB), probabilistic multilayer perceptron (MLP) [22, 23] and a support vector machine (SVM) classifier. If we apply NB and MLP in co-training, the confident examples are computed based on the posterior probabilities which are the classifier outputs. The probability of a test example belonging to one class is obtained as the product of the classifier outputs. The probability of a test example belonging to one class is obtained as the product of the classifier outputs. The probability of a test example belonging to one class is obtained as the product of the classifier outputs. The probability of a test example belonging to one class is obtained as the product of the classifier outputs.

4.2. Feature splits based on MI
Co-training requires constructing and splitting two sets of features from original data to perform successful classification. However, it is not easy to construct feature sets that satisfy both diversity and sufficiency. Hence, we propose an MI-based approach inspired by the idea of DCA for the task.

DCA model relaxes the independence assumption by decomposing the data into independent subsets where within each subset, the components are dependent. A practical and effective way to obtain DCA decomposition is by first performing independent component analysis (ICA) [24, 25] and then grouping the independent components into clusters by using MI as the metric between components [26, 27]. We propose to use the grouping part of DCA to split features by maxi-
imization of the MI between two sub-feature sets denoted as $I((f_i, i \in S), (f_j, j \in \{1, 2, \ldots, F\} - S))$, where $F$ is the total number of features, involving several stages:

1. Select and extract feature $f_i[k] = f_i(x_k)$ from each of $L$ labeled LTM samples $x_k$, $1 \leq k \leq L$;
2. Calculate $I(f_i, f_j), i, j = 1, \ldots, F$ (normalized through $\sqrt{1 - e^{-2I(f_i,f_j)}} \in [0, 1]$);
3. Construct $F \times F$ MI matrices $M^{[e]}$ and $M^{[b]}$ using $L_1$ good and $L_2$ bad labeled samples, respectively;
4. Calculate the MI matrix given class information, denoted as $M = (M^{[e]} \times L_1 + M^{[b]} \times L_2)/L$;
5. Generate dendrograms using hierarchical clustering based on the distance measure $1 - M$ and $M$, respectively, where $I$ is the $F \times F$ matrix with all entries equal to 1.

After applying this MI-based approach, we perform classification using the co-training algorithm with the feature splits, we thus obtain.

5. EXPERIMENTAL RESULTS

Labels for LTM samples are assigned by an expert. Then we select 270 samples for which we have confidence in their labels including 47 bad and 223 good half-ribs from a total of 1320 samples. For each experiment, we take the average of 100 runs as the final classification accuracy and report the std. In the co-training procedure, we select labeled training and test data randomly from 270 samples and allow others to be unlabeled data for each run. Also, we keep the same proportion (1/5) of bad/good samples in the training set of each run to make sure co-training yields consistent results.

5.1. Evaluation of feature splits

We obtain two feature splits that satisfy two conditions of co-training from the MI-based method. To investigate the significance of these two splits for co-training, we randomly select 11 features from 14 features to construct feature splits and perform co-training with NB classifier using 36 labeled training data. For each split, we require that each classifier has at least three features. Then we analyze the experimental results of 364 feature combinations, each combination containing two splits. The results include MI between-classifier, the average MI within-classifier and classification accuracy. In the analysis, we perform (1) a paired $t$-test between classification accuracy of two splits (Split 1 represents the split based on $1 - M$; Split 2 represents the split based on $M$), and (2) a permutation test on multiple regression coefficients. The multiple regression is defined as $m = \alpha n_1 + \beta n_2$, where $m$ represents a random subset of obtained classification accuracy, $n_1$ and $n_2$ denote the corresponding subsets of between- and within-classifier MI, and $\alpha$ and $\beta$ are the coefficients. We also perform co-training on two splits of 14 features using several classifiers, including NB, MLP and SVM with different kernels, and give the comparison in Figure 6 and Table 1.

The results in Table 2 show that within-class MI has significantly negative correlation with the classification accuracy. In other words, the more powerful the retained features within a classifier, the higher is the obtained classification accuracy. The between-class independence is also important since the $t$-value of between-class MI is not significant and it is very small compared to the result of within-class. Even though the classification rates yielded by Split 2 are not significantly higher than the results of Split 1, our results indicate that sufficiency is more important than diversity. Thus, in the following analysis, we apply Split 2 in co-training to evaluate the classification performance.

5.2. Performance of co-training

One of the advantages of co-training is that even a few labeled training data may lead to high classification accuracy. We thus evaluate the performance of co-training with increasing number of initially available labeled training data and compare the results with supervised learning as shown in Figure 7. The results indicate that co-training has great power using a few labeled training samples compared to supervised learning with the NB classifier.

6. CONCLUSION

In this work, we propose a novel MI-based approach to split features for co-training. These features are extracted from LTM data using several feature extraction methods. We introduce an efficient method to perform co-training when features are not naturally separated into two subsets. In earlier studies, few methods of feature splits have been proposed for co-training [7–9]. In these methods, best splits are evaluated or selected among a huge amount of random feature splits according to their criteria. Additionally, our experimental result indicates that sufficiency has a more significant contribution to classification accuracy compared to diversity, which clarifies the dependence of co-training performance on two conditions diversity and sufficiency.

Table 1. Results of two splits in co-training(%)  
<table>
<thead>
<tr>
<th></th>
<th>NB</th>
<th>SVM</th>
<th>MLP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Linear</td>
<td>RBF ($\sigma = 2.5$)</td>
<td>Poly (2)</td>
</tr>
<tr>
<td>Split 1</td>
<td>97.7 ± 2.1</td>
<td>97.7 ± 2.0</td>
<td>98.5 ± 1.9</td>
</tr>
<tr>
<td>Split 2</td>
<td>98.4 ± 2.0</td>
<td>97.8 ± 2.4</td>
<td>98.1 ± 1.8</td>
</tr>
</tbody>
</table>

Table 2. Evaluation of two feature splits

<table>
<thead>
<tr>
<th></th>
<th>Between-class</th>
<th>Within-class</th>
<th>Split 1 − Split 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$-values</td>
<td>2.6</td>
<td>-42.8</td>
<td>-1.1</td>
</tr>
<tr>
<td>$p$-values</td>
<td>0.06</td>
<td>0.18 $\times 10^{-5}$</td>
<td>0.28</td>
</tr>
</tbody>
</table>
7. REFERENCES


