Note

Feng-Yang Hsieh

1 Bootstrapping

The bootstrapping method involves an iterative process of training a classifier on mixed datasets. We start by training a classifier on the initial mixed datasets. The trained classifier is then used to reclassify the data, creating a new mixed training set. This process is repeated multiple times to increase the sample fraction differences in mixed datasets. We want to explore whether this process can improve CWoLa's performance.

We consider the events sampled from the normal distribution for the testing and implement this method. We found this method is unsuccessful. The model initially achieved the best performance, then worsened at subsequent iterations.

The reason is the reclassification step breaks the key assumption of the CWoLa approach: the signal and background events should have the same distributions in both mixed datasets. Figure 1 shows the initial signal and background distributions. Signal has the same distribution in mixed dataset M_1 and M_2, as does the background. Figure 2 shows signal and background distributions after the reclassification. We could observe the signal events have different distributions in M_1 and M_2. As a result, the assumption of the CWoLa approach is violated, leading to the failure of the bootstrapping method.

2 Multi-class CWoLa

The original CWoLa is only applied to the binary classification tasks. We want to extend the traditional CWoLa to more than two classes.

2.1 Dataset and model setup

We consider three pure samples, denoted by A, B, and C. The mixed datasets are denoted by M_1, M_2, and M_3. The training dataset is sampled from 5-dimensional normal

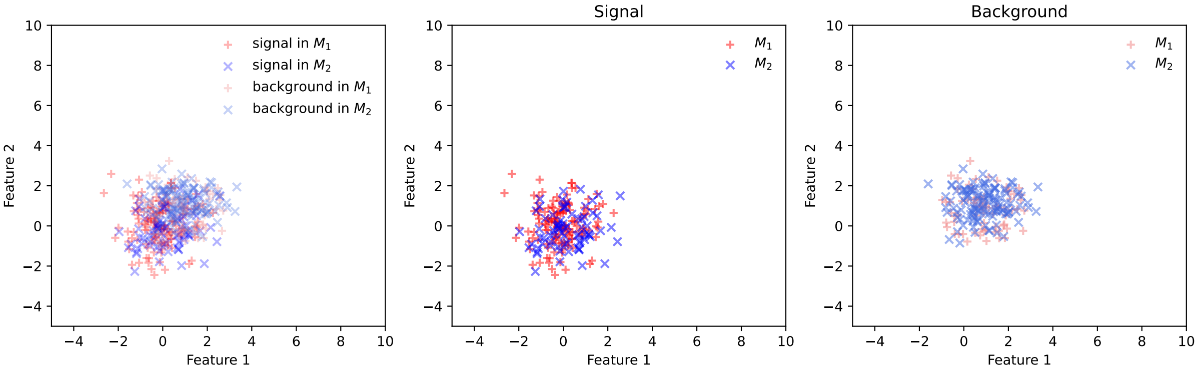


Figure 1: The signal and background samples distributions. The signal and background events are sampled from different two-dimensional normal distributions. They are randomly assigned to the mixed datasets M_1 or M_2.

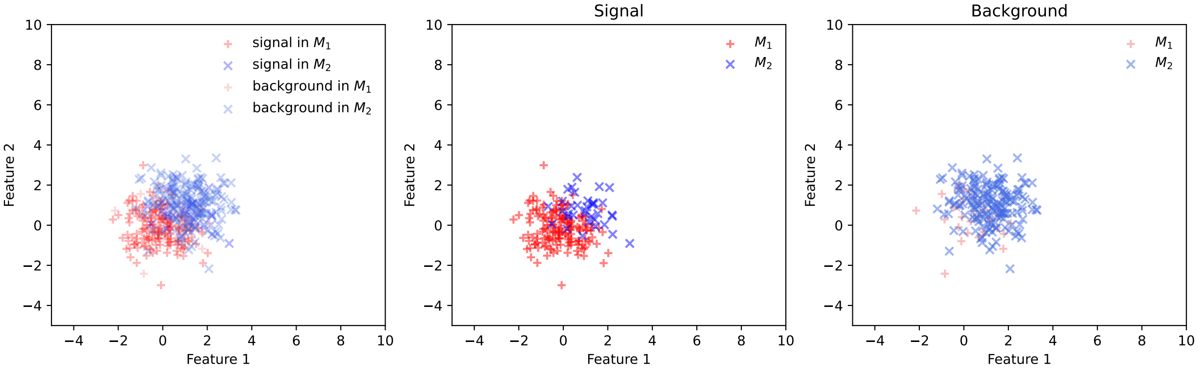


Figure 2: The signal and background samples distributions. The signal and background events are sampled from different two-dimensional normal distributions. The trained classifier assigns them to the M_1 or M_2.

distributions. The number of events in each mixed dataset is 10,000. The testing dataset consists of 1,000 pure samples for each class.

The neural network consists of two dense layers with eight hidden nodes. The loss function is categorical cross-entropy. To prevent over-training, we utilize the early stopping technique with patience 10. The output is a 3-dimensional vector, and each component can be interpreted as the probability belonging to each type.

2.2 Dominated case

To simplify the problem, we start by considering a case where each pure class dominates different mixed samples. This would help us understand the behavior of the multi-class classifier before going to more complex scenarios.

Table 1 lists the fractions of pure samples in mixed datasets. Each mixed dataset is dominated by one pure sample type. Figure 3 illustrates the distributions of the pure classes within these datasets.

Table 1: Fractions of pure samples in mixed datasets.

Dataset	A	B	C
M_1	0.70	0.20	0.10
M_2	0.10	0.70	0.20
M_3	0.20	0.10	0.70

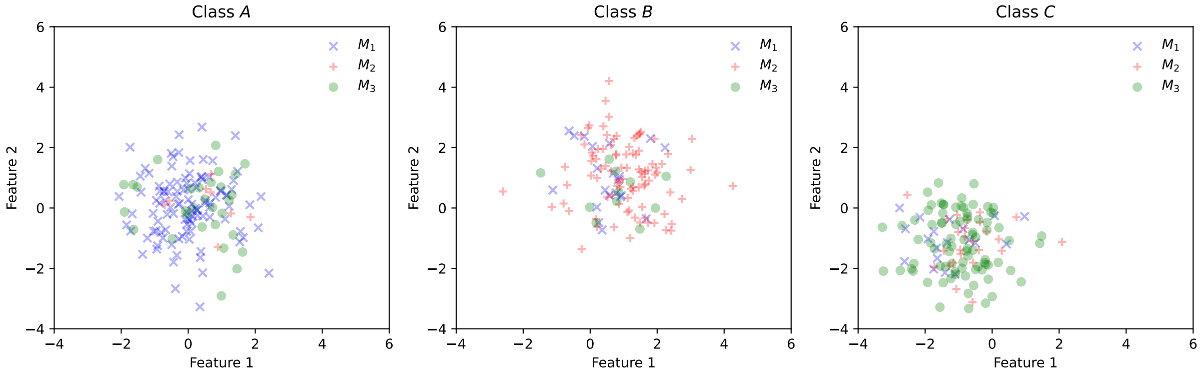
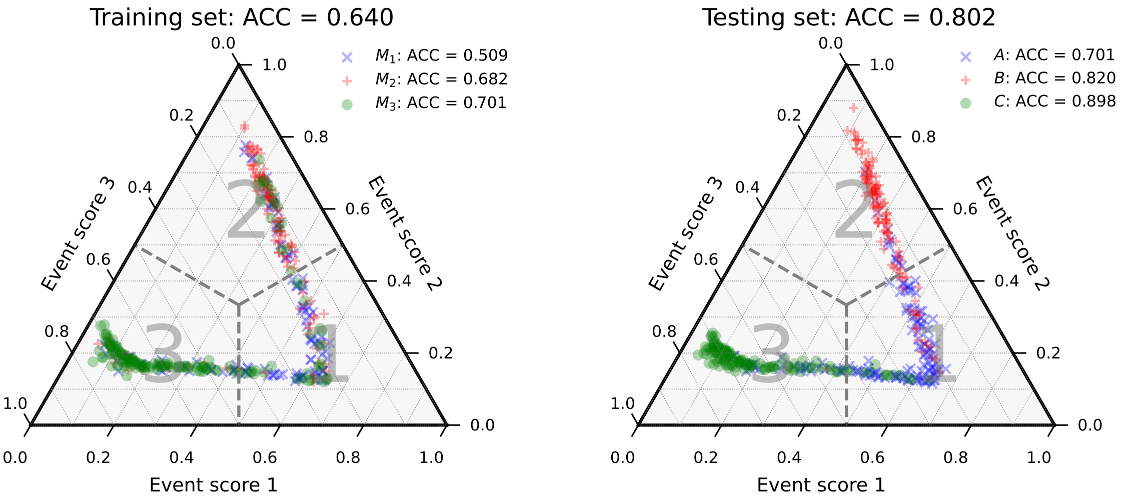


Figure 3: Distributions of pure samples in the mixed datasets. The classes A, B, and C are sampled from the normal distribution with mean values $0, 1, -1$, respectively.

If the output vector suggests that the first type has the highest probability for dominated cases, it can be directly interpreted as belonging to the class A. The same logic applies to classes B and C.

The event scores are the components of the output vector. Since the sum of event scores is always 1, we use the ternary plots to represent the event score distributions. In these plots, the coordinate of a point should be read following the direction of the ticks on each axis.

Figure 4 shows the scatter plots of output vectors. The event score distributions are well-separated for each class, indicating successful training. The connections from regions 2 to 1 to 3, due to the mean value of class B, A, and C, are $1, 0$, and -1, respectively.



(a) Training data

(b) Testing data

Figure 4: Ternary plots of event score distributions. The grey dashed lines separate the regions corresponding to different classification results. The total accuracy (ACC) and the accuracy for each type are displayed.

Table 2 presents another type of the dominated case. Although the pure sample A has the largest fraction in each mixed dataset, the pure class is still dominant in different mixed datasets. Figure 5 illustrates the distributions of the pure classes in these datasets.

Table 2: Fractions of pure samples in mixed datasets.

Dataset	A	B	C
M_1	0.80	0.10	0.10
M_2	0.70	0.25	0.05
M_3	0.60	0.20	0.20

Figure 6 shows scatter plots of output vectors. Although the training accuracy is not good, the testing results demonstrate excellent performance.

From these results, we can conclude the ternary CWoLa works for dominated cases.

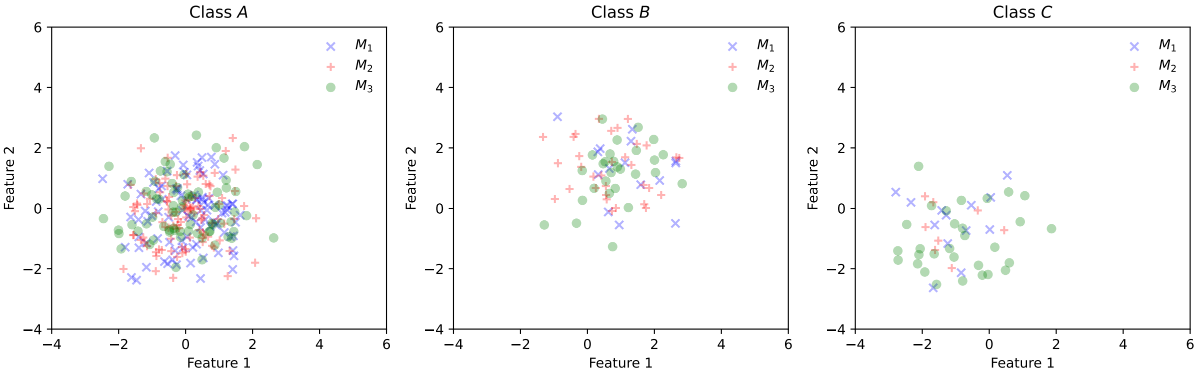
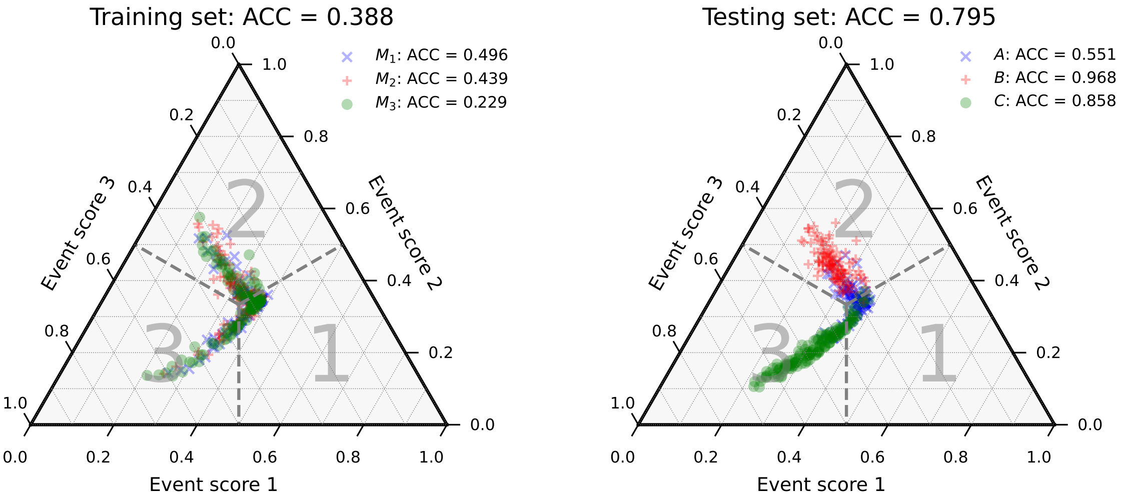


Figure 5: Distributions of pure samples in the mixed datasets. The classes A, B, and C are sampled from normal distributions with mean values 0, 1, and -1, respectively.



(a) Training data

(b) Testing data

Figure 6: Ternary plots of event score distributions. The grey dashed lines separate the regions corresponding to different classification results. The total accuracy (ACC) and the accuracy for each type are displayed.

2.3 Ambiguous case

Table 3 lists the fractions of pure samples in mixed datasets. The ambiguous cases mean that the pure sample is not dominant in different mixed datasets. For instance, in table 3, pure samples B and C both dominate in M_3. Figure 7 illustrates the distributions of the pure classes within these datasets.

Table 3: Fractions of pure samples in mixed datasets.

Dataset	A	B	C
M_1	0.80	0.10	0.10
M_2	0.70	0.20	0.10
M_3	0.60	0.25	0.15

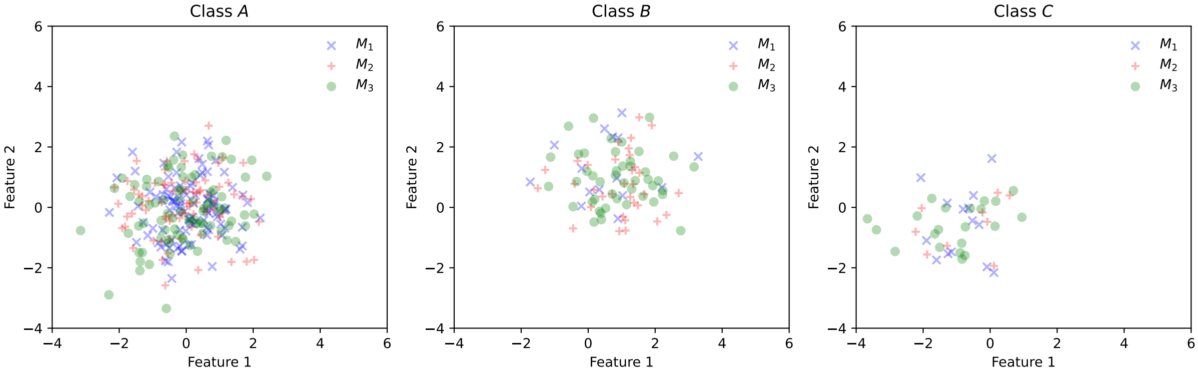
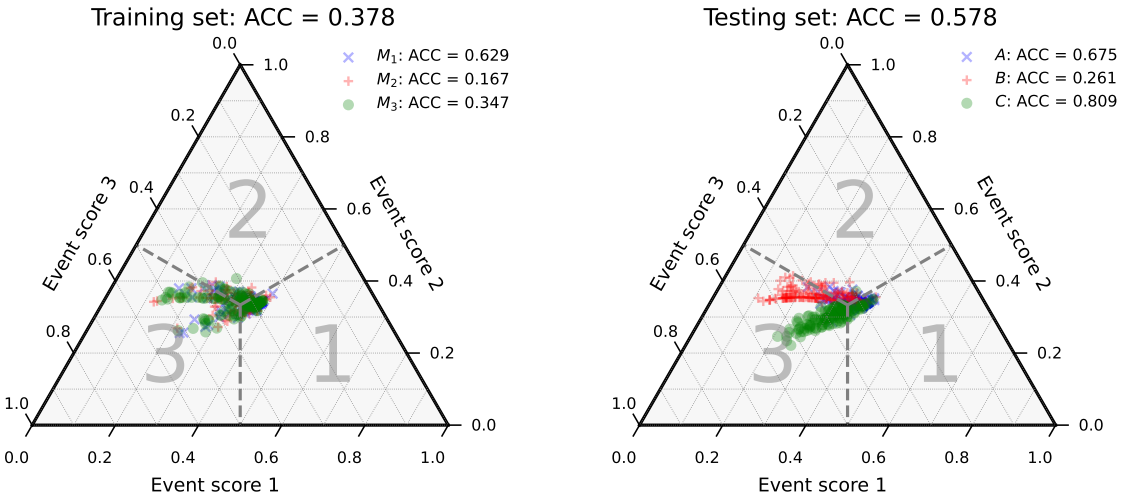


Figure 7: Distributions of pure samples in the mixed datasets. The classes A, B, and C are sampled from the normal distribution with mean values $0, 1$, and -1, respectively.

For ambiguous cases, we apply the same logic to interpret the output vector as in dominated cases. However, the results could be problematic.

Figure 8 shows scatter plots of the output vectors. While the testing accuracy is very low, different types of pure samples exhibit distinct distributions. This suggests that we need another method to interpret the output vector.

Note that the accuracy for class B is very low, while the accuracy for class C is significantly higher. This suggests that when the classifier receives an event from either class B or C, it tends to assign a high event score of type 3. This behavior is consistent with the fact that the pure samples B and C both are dominated in M_3. We need to refine our classification method or better interpret the output vectors to improve performance.



(a) Training data

(b) Testing data

Figure 8: Ternary plots of event score distributions. The grey dashed lines separate the regions corresponding to different classification results. The total accuracy (ACC) and the accuracy for each type are displayed.

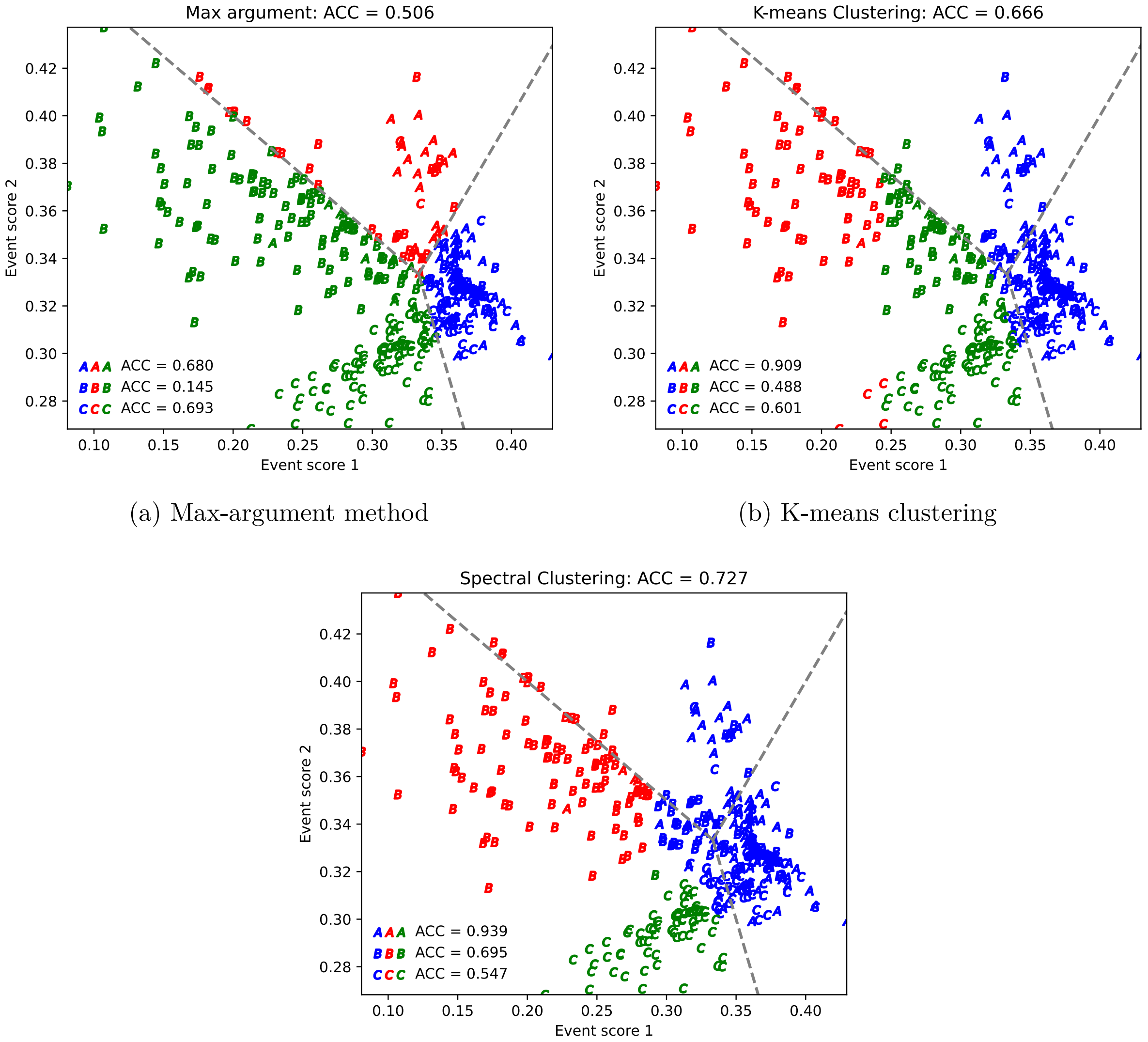
3 Output vector interpretation

In section 2, the predicted label is determined from the output vector using the following method: if the first class has the highest probability, the sample is assigned to class A. The same logic applies to classes B and C. We refer to this approach as the “max argument” method.

While this interpretation works well for dominant cases, it performs poorly for ambiguous cases. However, for such cases, the output vector distributions still differ between pure classes, suggesting that we need a better prediction method. The clustering algorithms might provide an alternative way to determine the type of a given sample.

We consider the ambiguous cases described in section 2.3 and test different clustering methods as alternative prediction strategies.

Figure 9 presents the prediction results for various methods, where colors represent the predicted labels. Both clustering approaches show improvements in overall accuracy. Table 4 summarizes the mean and standard deviation of accuracies for each prediction method. Notably, both clustering methods significantly improve the accuracy for pure class B, contributing to the observed increase in total accuracy.



(a) Max-argument method

(b) K-means clustering

(c) Spectral clustering

Figure 9: Event score distribution with prediction results. The grey dashed lines indicate regions corresponding to different predicted labels using the max argument method. The total accuracy and per-class accuracies are displayed.

Table 4: Accuracy of different prediction methods. The means and standard deviations are computed over ten training runs.

Prediction Method	A	B	C	Total
Max argument	0.742 ± 0.035	0.164 ± 0.100	0.707 ± 0.091	0.538 ± 0.027
K-means clustering	0.868 ± 0.028	0.534 ± 0.077	0.595 ± 0.050	0.666 ± 0.041
Spectral clustering	0.823 ± 0.114	0.625 ± 0.096	0.656 ± 0.173	0.702 ± 0.061