

～音による剣道の補助的な審判システム～

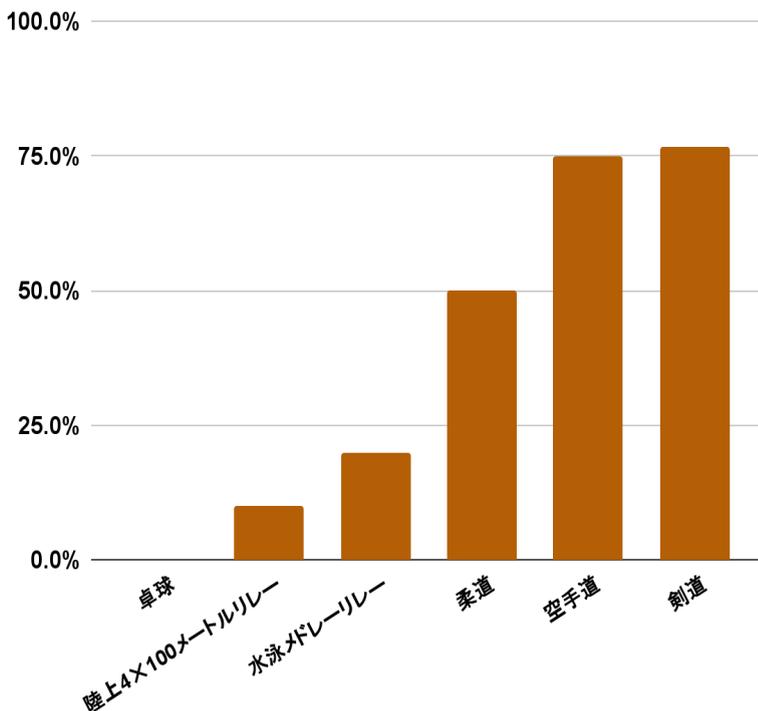
群馬県立高崎高等学校 鎌形悠 西村祥 黒木智仁

1.要旨

本研究では音による判断で剣道の補助的な審判システムの開発をした。剣道は審判員によって主観的に1本の判断が行われるので、他の競技でも採用されているような審判システムがあれば、より正確な判断ができて選手もの不満なども減らすことができるのではないかと考えた。しかし、剣道の1本の判断基準は厳密に定められているわけではなく、100%の判断は下すことはできないので「補助的」なシステムという立ち位置で研究を行った。ここでは竹刀が打突部位に当たっているかの判断を主な目的としている。

2.はじめに

まず私達はネット上で、国体の剣道の判定に対して多くの不満が見られたことに疑問を持ち、国体の過去30年間の卓球、陸上4×100メートルリレー、水泳メドレーリレー、柔道、剣道、空手道の開催県と優勝及び準優勝県の一致率を調べた。



このグラフからわかるように卓球、陸上、水泳のように点数や記録によって勝敗のきまる競技は一致率が低いのに対して柔道、空手道、剣道のような武道系の競技では一致率が高いことがわかる。私達はこれらの結果は武道系の競技の勝敗に人による判定が大きく関わっていることが要因の一つとなっていると判断した。

3.先行研究

先行研究では鋭指向性マイクとデータレコーダーを用いて有効な打突音を収録し、メンとドウの周波数構造を分析し明確な差があると結論づけていた。

剣道競技における有効打突判定システムの検討

4.研究1

①目的

剣道の打突部位である「メン」「コテ」「ドウ」の音における周波数の違いを、グラフを用いて明確にすることを目的とする。

②方法

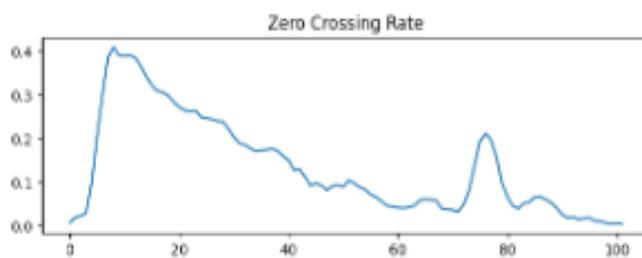
高崎高校の剣道場にて、「メン」「コテ」「ドウ」の打突音をそれぞれ1つずつ録音した。録音データは Google Colaboratory 上で分析し、特にゼロ交差率(Zero Crossing Rate)に着目してグラフを作成した。

ゼロ交差率とは、音声波形の振幅が正から負、あるいは負から正へと変化する回数を単位時間あたりで示す指標であり、周波数の高さに関係している。

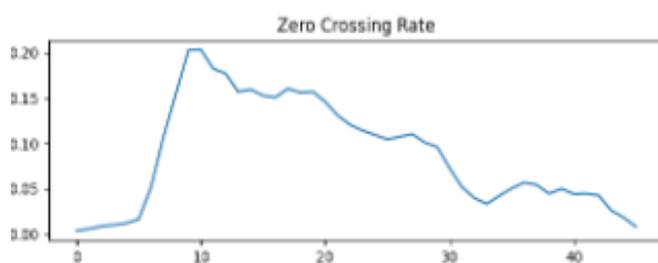
振幅が正であるときは空気圧が基準より高く、負であるときは空気圧が基準より低いことを意味する。

③結果

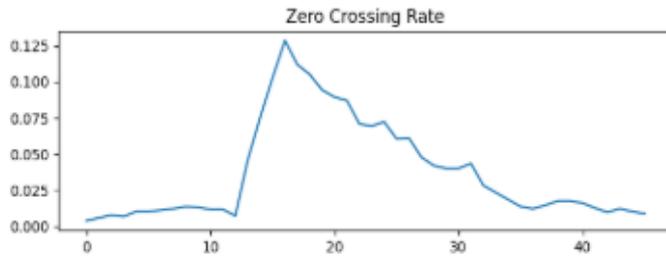
ドウ↓



メン↓



コテ↓



ドウの音

最大ゼロ交差率: 約 0.40

→波形が短い周期で振動しており、高周波成分を多く含むことが示唆される。

メンの音

最大ゼロ交差率: 約 0.20

→ドウよりは低周波だが、コテよりは高周波であることがわかる。

コテの音

最大ゼロ交差率: 約 0.125

→最も低い周波数成分を持つと考えられる。

④考察

剣道の打突音には、打突部位ごとに周波数の構造に違いがあることが確認された。

この違いを活用することで、試合中の音声から打突音だけを抽出し、竹刀がどの部位に当たったかを判定できる可能性がある。

特に、ドウの音は他の打突音に比べて周波数的に顕著な特徴が見られたため、今後はこの「ドウの音」に着目してさらに詳細な分析を行うこととした。

5.研究2

①目的

特徴を捉えやすいドウの音に絞って、ドウの音とそうでない音の特徴の違いをモデルに学習させて判断させる。ここでは、竹刀がドウに当たっているかの判断によって審判の補助を行うことにした。

②方法

研究1に基づけばドウの音の周波数には十分な特徴が存在すると言えるので、周波数に関する以下の4つの特徴量を用いて機械学習を行う。

特徴量名	内容説明	音声信号のどんな性質を表すか
Centroid (スペクトル重心)	周波数成分の「重心」のような値。音の明るさや高低感の指標。	音の「明るさ」や「高い音が低い音か」の傾向
MFCC (メル周波数ケプストラム係数)	人間の聴覚特性に基づく周波数成分の特徴量。音声の「質感」や「声の特徴」を抽出。	音声の内容や話者の特徴など、詳細な音の性質
ZCR (ゼロ交差率)	音声波形がゼロを横切る回数。音のざらつきや騒音の指標。	音の粗さや雑音の多さ、無声音・有声音の判別など
Flux (スペクトルフラックス)	隣接するフレーム間のスペクトルの変化量。音の動きや変化の指標。	音の変化の激しさ、動的な変化 (例: 打撃音や声の変化)

chatgptに作るコードについて詳細に要求して、googlecolabratoryに入力することで機械学習を行った。

⇩実際に用いたコード

(i)

```
import os
import librosa
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

<説明>

まず、必要なライブラリを導入した。

os: ファイル・フォルダの操作に使う。

librosa: 音声処理ライブラリ。特徴量抽出に使う。

numpy: 数値計算用ライブラリ。

sklearn.model_selection: データを学習用とテスト用に分ける。

sklearn.ensemble: ランダムフォレストモデル(学習をもとに分類する)。

sklearn.metrics: 分類結果の評価指標を出力する。

(ii)

```
# 特徴量抽出関数 (Centroid + MFCC + ZCR + Flux)
def extract_features(file_path):
    y, sr = librosa.load(file_path, sr=None)

    # スペクトルセントロイド
    centroid = librosa.feature.spectral_centroid(y=y, sr=sr)
```

```

# MFCC
mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=13)

# ZCR (Zero Crossing Rate)
zcr = librosa.feature.zero_crossing_rate(y)

# スペクトルフラックス (近似: 短時間フレームごとの差分の2乗)
stft = np.abs(librosa.stft(y))
flux = np.sqrt(np.sum(np.diff(stft, axis=1)**2, axis=0,
keepdims=True))

# 特徴量をベクトルにまとめる
feature_vector = np.hstack((
    np.mean(centroid), np.std(centroid),
    np.mean(mfcc, axis=1), np.std(mfcc, axis=1),
    np.mean(zcr), np.std(zcr),
    np.mean(flux), np.std(flux)
))

return feature_vector

```

<説明>

表で示した4つの特徴量を指定する。

(iii)

データ読み込み関数

```

def load_dataset(base_path):
    X, y = [], []
    label_map = {'ドウ': 1, '他': 0}
    for label_name in label_map:
        folder = os.path.join(base_path, label_name)
        for fname in os.listdir(folder):
            if fname.endswith('.wav'):
                path = os.path.join(folder, fname)
                features = extract_features(path)
                X.append(features)
                y.append(label_map[label_name])
    return np.array(X), np.array(y)

# パス指定 (ドライブに応じて変更)
base_path = '/content/drive/MyDrive/dataset/train sound'

# データ読み込み
X, y = load_dataset(base_path)

```

<説明>

base pathに示されたフォルダ内のドウのフォルダと他のフォルダから音声ファイルを取り出して読み込み、ドウの音を「1」他の音を「0」として分類ラベルを付ける。そしてそれぞれの音声ファイル(wav形式)に対して(ii)で指定した4つの特徴量を抽出する。

※basepathに示されているのはgoogle drive内のフォルダで、その中には「ドウ」と「他」の2つのフォルダが含まれている。そして、そのフォルダにはそれぞれの音声ファイルが保存されている。

(iv)

クラス分布確認

```
print("クラス分布:", dict(zip(*np.unique(y, return_counts=True))))
```

分割 (stratifyで偏り回避)

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.25, random_state=42, stratify=y
)
```

<説明>

それぞれのフォルダのデータ数を示し、バランスに偏りが無いことを確認する。ドウ/他の割合を保ったままデータ数のうちの25%をテスト用に、75%を学習用に分割する。

(v)

モデル学習

```
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

評価出力

```
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(
    y_test, y_pred, labels=[0, 1], target_names=["他", "ドウ"],
    zero_division=0
))
```

<説明>

ランダムフォレストによって機械学習を行い、テスト用のデータに対して予測を行う。その結果として正解数、不正解数を可視化する。

③結果

Classification Report:				
	precision	recall	f1-score	support
他	0.84	0.94	0.89	17
ドウ	0.80	0.57	0.67	7
accuracy			0.83	24
macro avg	0.82	0.76	0.78	24
weighted avg	0.83	0.83	0.82	24

出力された表によると、、、

precision

ドウ...80% ドウの音と判定したうち**80%**はドウの音であった。

他.....84% 他の音と判断したうち**84%**は他の音であった。

recall

ドウ...57% ドウの音のうち**57%**はドウの音と判断した。

他.....94% 他の音のうち**94%**は他の音と判断した。

④考察

全体の正確な分類ができたかの割合は83%とある程度精度は高い。一方で、ドウの音を検出できる割合が57%と低いのでドウの音を見逃しやすいといえる。

データ数の少なさやデータに含まれるノイズが主な原因で正確性を欠いていると考えられるので改善するには、多数のデータと高性能な機材などが必要である。補助システムとして一部有効性を示すことはできたが実用化には程遠いといえる。

6.まとめ

はじめは、研究の目的である剣道の補助的な審判システムの開発に向けて、剣道の試合映像を読み込んで機械学習したモデルに判断させるという研究を行っていたが、技術と知識の不足により断念し、音による研究をするを選び、この研究に至った。今回の研究結果からもわかるように、音による補助的な審判システムは一部有効性が認められる。そのため、多くのデータの収集やノイズ等の排除を行うことで、より正確に機械学習をしたり、今回は断念した視覚的なアプローチと組み合わせたりすれば、補助的な審判システムとしての実用化に近づくことはできるのではないかと考えられる。

7.終わりに

①謝辞

群馬県立高崎高等学校の先生方をはじめとして多くの方々に指導助言をしていただきました。深く感謝致します。

②参考文献

1) 国民体育大会結果【第1回(1946)～第65回(2010)】

▶https://www.japan-sports.or.jp/Portals/0/images/archives/20220617_kokutai.pdf

2) 下重大地ら (H29) 平成 29 年度電気関係学会東北支部連合大会より
「剣道競技における有効打突判定システムの検討」

▶https://www.jstage.jst.go.jp/article/tsjc/2017/0/2017_231/pdf/-char/ja