

# Proposal

Niels Lange      Beate Zywietz

18. Dezember 2019

## 1 Fragestellung

Welche Emotionsklassen können Decision Tree und Support Vector Machine jeweils am besten unterschieden und welche am schlechtesten?

## 2 Verwandte Arbeiten

- Emotion recognition using a hierarchical binary decision tree approach (Chi-Chun Lee, Emily Mower, Carlos Busso, Sungbok Lee, Shrikanth Narayanan)
- Emotion recognition in speech signal using emotion-extracting binary decision trees (Jarosław Cichosz<sup>1</sup>, Krzysztof Ślot)
- Investigation of combining SVM and decision tree for emotion classification (Thao Nguyen, I. Bass, Mingkun Li, I.K. Sethi...)

## 3 Daten/Toolkits

- IEMOCAP (bis jetzt primär Klasse 0 (anger), 2 (excited), 4 (frustration), 6 (neutral) und 8 (sad))
- Scikit learn (DT und SVM)
- Eventuell später zweiter Korpus, um Ergebnisse zu vergleichen

## 4 Baseline

### 4.1 Decision Tree

test size = 0.3  
criterion = 'gini'  
min samples split = 100

```
Accuracy: 0.42534381139489197
      precision    recall  f1-score   support

    0.0         0.48      0.44      0.46         310
    2.0         0.29      0.26      0.28         310
    4.0         0.41      0.46      0.43         592
    6.0         0.42      0.43      0.42         518
    8.0         0.55      0.51      0.53         306

 accuracy          0.43         2036
  macro avg         0.43         0.42      0.42         2036
 weighted avg         0.43         0.43      0.42         2036

[[135  50  95  23   7]
 [ 55  82 105  58  10]
 [ 74  75 271 142  30]
 [ 15  61 137 221  84]
 [  4  11  50  84 157]]
pred: (array([0., 2., 4., 6., 8.]), array([283, 279, 658, 528, 288]))
test: (array([0., 2., 4., 6., 8.]), array([310, 310, 592, 518, 306]))
train: (array([0., 2., 4., 6., 8.]), array([ 793,  731, 1257, 1190,  778]))
```

### 4.2 Support Vector Machine

test size = 0.3  
gamma = scale  
C = 250.0

```
      precision    recall  f1-score   support

    0.0         0.62      0.31      0.41         310
    2.0         0.40      0.10      0.16         310
    4.0         0.39      0.61      0.48         592
    6.0         0.43      0.42      0.42         518
    8.0         0.46      0.58      0.51         306

 accuracy          0.43         2036
  macro avg         0.46         0.40      0.40         2036
 weighted avg         0.45         0.43      0.41         2036

[[ 95  13 154  26  22]
 [ 39  31 159  65  16]
 [ 16  14 359 135  68]
 [  3  17 179 215 104]
 [  1   2  66  60 177]]
pred: (array([0., 2., 4., 6., 8.]), array([154,  77, 917, 501, 387]))
test: (array([0., 2., 4., 6., 8.]), array([310, 310, 592, 518, 306]))
train: (array([0., 2., 4., 6., 8.]), array([ 793,  731, 1257, 1190,  778]))
```

## **5 Zeitplan/Nächste Schritte**

### **5.1 Ferien - 1. Woche**

- Themenverwandte Paper auswerten und gegebenenfalls weitere Literatur suchen
- Einen weiteren Korpus finden, um Ergebnisse zu vergleichen und festzustellen, welche Schwierigkeiten auf die Qualität der Trainingsdaten zurückzuführen sind
- Korpus auf Baseline-Systemen testen

### **5.2 2. Woche**

- Einstellungen der Baseline-Systeme anpassen
- Performance von DT und SVM für verschiedene Klassen vergleichen und dokumentieren

### **5.3 3. Woche**

- Weitere Dokumentation
- Ausarbeitung des Projektberichts

### **5.4 4. Woche - Ende**

- Ausarbeitung des Projektberichts