

Dokumentation Erkennung Klatschen, Klopfen, Husten

Tobias Tannert

April 9, 2014

In dieser Dokumentation werden die wichtigsten Dateien, Tools und Vorgehensweisen der Arbeit kurz erklärt. Die Dokumentation kann (hoffentlich) bei der Einarbeitung in das MATLAB System helfen.

1 Detektorschwelle

Optimierte Parameter des Detektors in Tabelle 1. Über die Höhe der Schwelle FacVar bzw. FacVarHysteresese kann das Verhalten des Detektors angepasst werden (s. ROC).

Parameter	Wert	Bedeutung
FacMinEvent	0.1	Faktor der minimalen Dauer eines Ereignisses
AdaptionTime	2	Aktualisierungszeit der Rauschparameter (in s)
FacVar	20	Faktor für Schwelle bei Stille
FacVarHysteresese	20	Faktor für Schwelle bei Geräusch
q	8	Schwelle für Ausreißer bei Rauschparameterberechnung
Alpha	0.2	Tiefpass-Faktor für Aktualisierung der Rauschparameter
FacPeakConsecutiveEvents	0.7	Höhe des Anstiegs eines nachfolgenden Ereignisses

Table 1: optimierte Parameter der Detektion

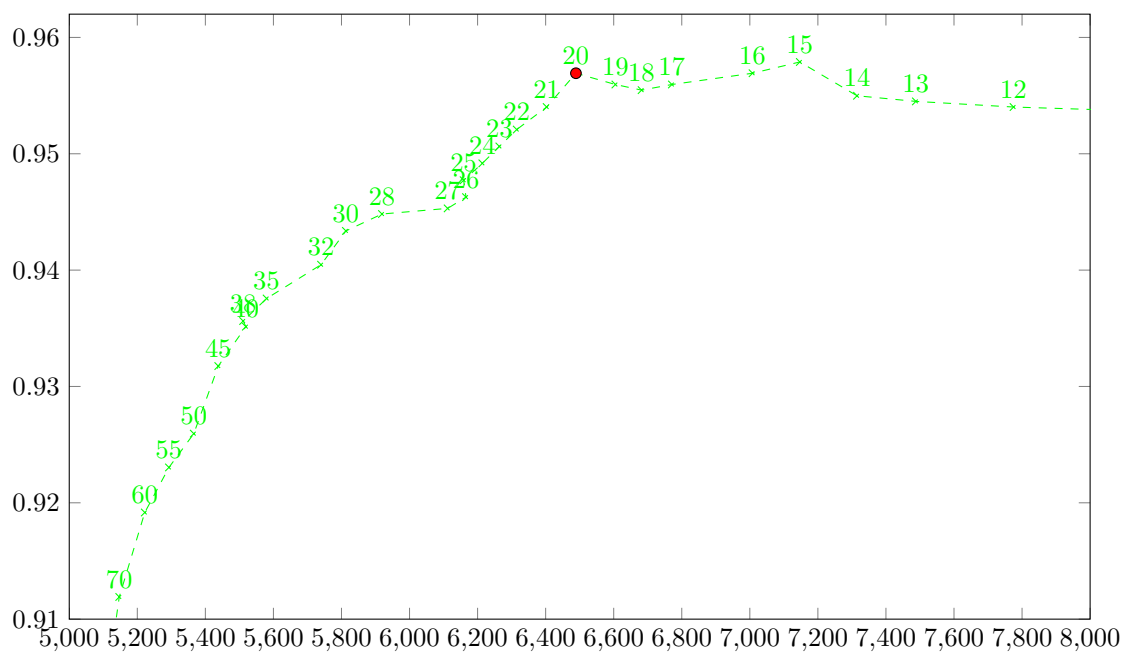


Figure 1: Abhängigkeit von FacVar = FacVarHysteresis

FacVar	P_D	N_{FA}
12	95,40	7775
13	95,45	7488
14	95,50	7315
15	95,79	7147
16	95,69	7008
17	95,60	6771
18	95,55	6682
19	95,60	6605
20	95,69	6489
21	95,40	6403
22	95,21	6315
23	95,06	6264
24	94,92	6214
25	94,77	6159
26	94,63	6166
27	94,53	6112
28	94,48	5919
30	94,34	5813
32	94,05	5740
35	93,76	5580
38	93,56	5511
40	93,51	5519
45	93,18	5438
50	92,59	5366
55	92,30	5293
60	91,92	5222
70	91,19	5147

Table 2: Abhängigkeit von FacVar = FacVarHysterese

2 Testdatenbank

Im Ordner Testdatenbank. Achtung, Klassenlabel bei database1, database2 in unterschiedlicher Zeile (s. → dect_events). Beachten bei Featureextraktion.

- **database1.mat**
Datenbank des ursprünglichen Systems. Mit 3 Klassen. Erstellt ohne adaptive Schwelle.
- **database2.mat**
Verwendet für erweiterten Klassifikator. Mit 4 Klassen. Ereignisse extrahiert mit adaptiver Schwelle und optimierten Detektorparametern.

3 AudioEditor

MATLAB Tool AudioEditor von der MATLAB Central. Modifiziert zum Anzeigen von Energieverlauf, Detektorschwelle, uvm. Möglichkeit zum Markieren von Ereignissen in der Tonspur.

Aufruf: AudioEditor(Fs, y, pow, thresh, dect, reject, plot_title, N, aux)

Argument	Bedeutung
Fs	Abtastfrequenz
y	Audiosignal
pow	Energieverlauf
thresh	Schwelle
dect	Blöcke detektierter Ereignisse
reject	Blöcke verworfener Detektionen
plot_title	String für Fenstertitel
N	Blocklänge
aux	Hilfssignale

Table 3: Argumente von AudioEditor

Bei Verwendung von Cell-Arrays für y, pow, thresh, dect, reject, aux werden mehrere Spuren gleichzeitig angezeigt. Z.B. geeignet zum Vergleich mehrerer Detektoren. Wird verwendet und aufgerufen aus der Datei read_wav_file.m.

Mit den bunten Buttons werden die verschiedenen Spuren ein-/ausgeblendet. Die Buttons 1,2 markieren am aktuell markierten Ausschnitt (rote Cursorbalken mit gehaltener Maustaste) ein Ereignis der Priorität 1 bzw. 2. Der Button S speichert die markierten Ereignisse in einer Datei.

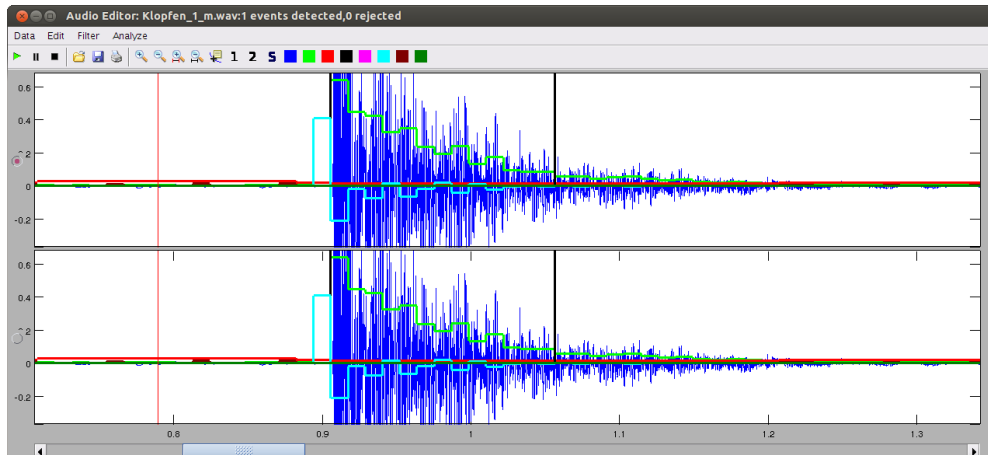


Figure 2: AudioEditor

4 wichtige MATLAB Dateien

4.1 read_wav_file

Wendet Detektion auf eine Audiodatei mit allen Parametersätzen (\rightarrow ParamExtraction) an. Optional Auswertung für alle Parametersätze, grafische Darstellung der Datei im AudioEditor oder Extraktion der Abtastwerte der detektierten Ereignisse.

4.2 read_wav_dir

Anwendung der Detektion auf ganzes Verzeichnis führt für jede Datei read_wav_file aus.

4.3 extraction_parameter_sweep

Führt Optimierungsschritt mit verschiedenen Parametersätzen durch. Enthält Erzeugung der Parametersätze paramExtraction. Führt mit den Parametersätzen read_wav_dir aus. Ergebnisse werden gespeichert in Variable extraction_info mit Auswertung der Detektion.

4.4 detection_rate_evaluation

Wertet extraction_info aus. Errechnung von P_D und N_{FA} für jeden Parametersatz. Darstellung in ROC. Dabei wird die ROC figure modifiziert.

blauer Button: Parametersatz des ausgewählten Punktes wird im Variableneditor angezeigt

roter Button: einige Parameter des ausgewählten Punktes werden in Konsole angezeigt

grüner Button: die Daten der ausgewählten Serie aus `extraction_info` werden zu `extraction_info.temp` hinzugefügt. Zum Sammeln guter Serien.

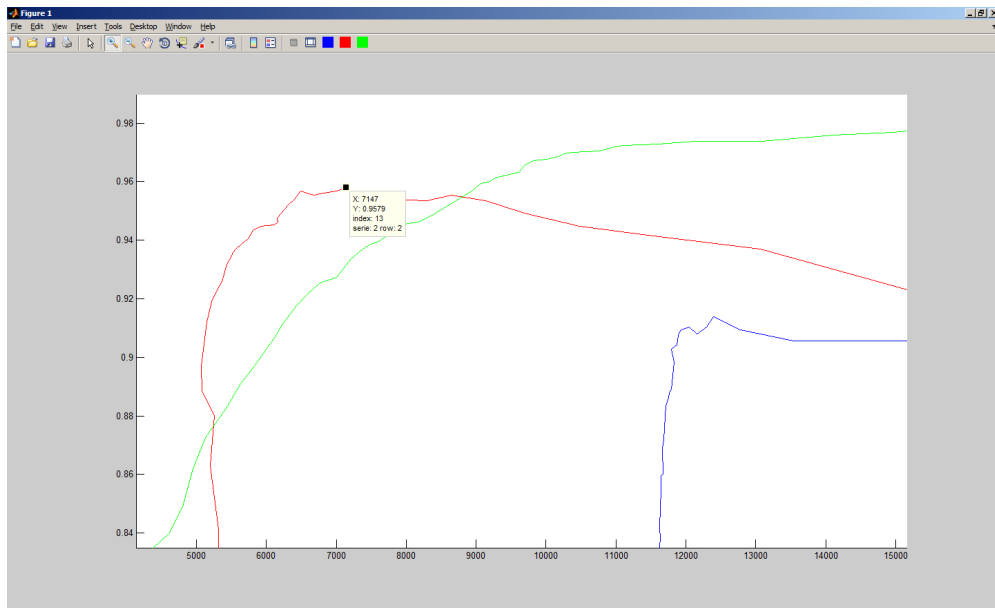


Figure 3: ROC figure

4.5 eval_extraction_info

Bearbeiten der Testdatenbank (`dect_events`). `dect_events` enthält alle vom Detektor gefundenen Ereignisse. `eval_extraction_info` entfernt Falschalarme und sorgt für eine Gleichgewichtung der Klassen in `dect_events`.

4.6 classification_script

Verwendet zum Optimieren von Parameterextraktion und Klassifikation (aus der Arbeit von C. Reuss). Arbeitet auf der Datenbank `databaseX.mat`, welche die Abtastwerte der Ereignisse `dect_events` enthält. Über Schalter werden die gewünschten Schritte eingestellt. Erzeugt die XML-Dateien für das Echtzeitsystem.

5 wichtige MATLAB Datenstrukturen

5.1 paramExtraction

Einstellungen für Detektor. Erzeugung der Datenstruktur s. `extraction_parameter_sweep.m`. Kann zu Array von Parametersätzen erweitert werden.

wichtige Einstellungen für Detektion:

Parameter	Bedeutung
<code>fileParam.checking</code>	Detektionen anhand der zero-crossing-rate prüfen und ggf. zurückweisen
<code>fileParam.AudioEditor</code>	untersuchte Tonspur im AudioEditor grafisch darstellen
<code>fileParam.evaluate</code>	Detektionen auswerten. z.B. Anzahl true positive, false positive, ...
<code>fileParam.getEventsAudio</code>	Tonspur detektierter Ereignisse extrahieren für Testdatenbank
<code>derivativeDetection</code>	Flankendetektor verwenden (\rightarrow <code>derivativeParam</code>), sonst bisheriger Detektor (\rightarrow <code>detection</code>)
<code>useFilterFFT</code>	Filter verwenden (\rightarrow <code>filterParam</code>)
<code>Distortion</code>	Störung überlagern, Rauschen oder andere Datei (\rightarrow <code>distortionParam</code>)

Table 4: wichtige Parameter von `param_extraction`

5.2 extraction_info

Auswertung der Detektion. Besteht aus zwei Arrays. Das Array `extraction_info{2,1}` enthält die Parametersätze. Das Array `extraction_info{1,1}` enthält die Detektionsergebnisse für die entsprechenden Parametersätze, welche im Folgenden erklärt werden. Die Ergebnisse sind für jede Datei aufgelistet. Spalte1 enthält den Dateinamen, Spalte2 das Klassenlabel, Spalte3 die Detektionsergebnisse für die Datei mit dem entsprechenden Parametersatz. Das Array der Detektionsergebnisse ist folgendermaßen aufgebaut:

1. Länge der Datei, Vielfaches von NEvent
2. Anzahl detektierter Ereignisse
3. Anzahl tatsächlicher Ereignisse Prio1
4. Anzahl tatsächlicher Ereignisse Prio2
5. Anzahl true positive (korrekt detektiert) Prio1
6. Anzahl true positive (korrekt detektiert) Prio2

7. Anzahl false positive (Falschalarm), wenn nur Prio1 betrachtet
8. Anzahl false positive (Falschalarm), wenn nur Prio2 betrachtet
9. Anzahl false negative (verpasst) Prio1
10. Anzahl false negative (verpasst) Prio2

Anhand dieser Ergebnisse wird von `detection_rate_evaluation` die gesamte Detektionsrate P_D und Anzahl der Falschalarme N_{FA} ermittelt.

5.3 dect_events

Enthält die Abtastwerte der detektierten Ereignisse (1. Zeile). Zeile2: =1, wenn es sich um Falschalarm handelt. Zeile3: Index der Datei, aus der das Ereignis kommt. Zeile4: Classlabel. `Database1.mat` enthält nur Abtastwerte und Classlabel.

6 typische Abläufe

6.1 Datei im AudioEditor betrachten

Parametersatz/-sätze `paramExtraction` erstellen z.B. mithilfe `extraction_parameter_sweep`. Dabei `fileParam.AudioEditor` setzen. `read_wav_file` ausführen:
`read_wav_file(paramExtraction, Dateipfad);`
 Für jeden Parametersatz werden die individuellen Verläufe von Detektorschwelle etc. in einer eigenen Spur eingezeichnet. So lässt sich das Verhalten der verschiedenen Detektoren an der gleichen Datei vergleichen.

6.2 Optimierungsprozess für Detektor

- `extraction_parameter_sweep` ausführen. In diesem Skript entsprechende Werte/Schritte für die Parameter wählen. Dabei in `paramExtraction` `fileParam.evaluate` setzen. Ergebnis ist die Variable `extraction_info`.
- `detection_rate_evaluation` ausführen. P_D und N_{FA} werden errechnet, die ROC wird geplottet. Mit den Zusatzfunktionen im Plot können einzelne Parametersätze angezeigt und untersucht werden.
- Anhand der Ergebnisse `extraction_parameter_sweep` modifizieren. Erneut ausführen
 ...

6.3 Datenbank erstellen

Einen einzelnen optimalen Parametersatz `paramExtraction` aufstellen. Dabei `fileParam.getEventsAudio` setzen. Die extrahierten Ereignisse werden in `dect_events` gespeichert. Mittels `eval_extraction_info` kann die Datenbank aussortiert werden. Speichern von `dect_events` unter `databaseX.mat`.

6.4 Klassifikator bewerten, XML-Dateien erstellen

siehe `classification_script`