Bezüglich verschiedener feature extraction Methoden in der Spracherkennung

ein Survey

Von Hauke Bünning

Proseminar: Verarbeitung gesprochener Sprache

Sommersemester 2020

Abstract:

Ich habe mich im Zuge meines Vortrages bereits mit bottleneck feature (BNF) extraction beschäftigt. Im Folgenden werde ich auf aktuelle Technologien eingehen und diese hinsichtlich ihrer praktischen Anwendbarkeit mit der BNF extraction vergleichen. Dabei werden die Vor- und Nachteile der einzelnen Technologien herausgearbeitet.

1. Einleitung:

Sprachsignale und Sprache im Allgemeinen haben eine hohe Komplexität, auch wenn man bereits von Störgeräuschen, Akzenten und schwierigen Lautstärkeverhältnissen absieht.

Um die Verarbeitung der Sprachsignale durch Computer etwas zu erleichtern und die Berechnungskosten in vertretbaren Größen zu halten, ist es also notwendig möglichst viele, nicht essentielle, Signalteile aus einem Audiodatensatz heraus zu filtern. Meist werden, für Berechnungen viel zu große, Inputvektoren auf eine Größe weit unter ihrer Originalgröße gebracht, wobei allerdings wichtig ist, dass alle Beispiele aus einem Datensatz gleiche feature Vektorgrößen haben, da sich sonst die Berechnung schwierig gestaltet.

Dies wird in der Realität über features extraction gemacht. Diese gehört wohl zu den wichtigsten Teilschritten der Spracherkennung überhaupt und das obwohl viele der verwendeten Algorithmen bereits viele Jahre alt sind. Es gibt eine Vielzahl verschiedener Ansätze, um an eine solche Aufgabe heranzugehen. Die wohl prominenteste davon ist Teile der menschlichen Audioverarbeitung, entweder Ohr oder Rachen-/Mundbereich, technisch nachzuvollziehen.

Über die Jahre sind dabei ein paar verschiedene Methoden entstanden. Allerdings ist für jede einzelne von ihnen eine gewisse Art der Vorbereitung nötig. Daher wird sich diese Arbeit sich weniger auf die, oftmals sehr ähnlichen, Vorbereitungsverfahren konzentrieren, sondern nur auf die feature extraction Methoden selbst.

Der primäre Fokus soll auf dem Vergleich von aktuell viel benutzten feature extraction Techniken und deren Vergleich zu Bottleneck features liegen. Dazu behandelt diese Arbeit eine kurze Zusammenfassung der Funktionalität der einzelnen Methoden und ihrer kennzeichnenden Merkmale und wird diese anschließend, unter den Gesichtspunkten Rechenkosten, Genauigkeit und der Verlässlichkeit, vergleichen. Bei diesen Gesichtspunkten handelt es sich, um die wichtigsten technischen Daten eines feature extraction Algorithmus.

2.Feature Extraction Techniken

Features extraction ist ein Prozess in der Sprach- und Signalverarbeitung, der primär darauf fokussiert ist die „wichtigsten“ Elemente auf einem Sprachsignal herauszufiltern und somit die weiterzuverarbeitenden Daten zu komprimieren, um Berechnungen weniger teuer zu machen. Dabei ist wichtig, dass trotz sehr starkem selektieren der Daten keinesfalls wichtige bzw. charakteristische Teile wegreduziert werden, da das feature sonst, unter Umständen, nutzlos ist.

Im Folgenden werde ich auf ein paar verbreitete feature extraction Techniken eingehen, speziell auf bottleneck-features, MFCC, LPC/LPCC und PLP. Bevor alle diese Techniken auf ein Sprachsignal angewendet werden können muss, in der Regel Preprocessing durchgeführt werden, was z.B. aus Frame Blocking, Windowing und einer Fast Fourier Transform (FFT) bestehen kann.

2.1 BNF

Bottleneck-features bezeichnen eine Art der feature extraction in der ein komplett verbundenes Multi-Layer-Perceptron (MLP) benutzt wird, um die raw input-features auf eine Größe von, typischerweise, 35-39 dimensionalen feature-Vektoren, zu komprimieren. Dies geschieht über ein sogenanntes mapping-demapping MLP, welches erst die input Daten komprimiert und dann wieder entpackt um sicher zu gehen, dass die features sich auf die ursprünglichen Daten zurückführen lassen. Sollte dies gegeben sein, wird der feature Vektor aus dem dritten und kleinsten Layer (Bottleneck Layer) abgelesen. Darauf folgt meist noch ein Dekorrelationsverfahren, z.B. per Heteroscedastic Linear Discriminant Analysis (HLDA), womit die Vektoreinträge auf ihre Abhängigkeit voneinander geprüft werden.

2.2 LPC/LPCC

Linear Predictive Coding (LPC) ist ein Verfahren der Sprachsignalverarbeitung, bei dem ein virtuelles Modell des Stimmtraktes des Menschen erstellt wird. Die Stimmbänder und der Kehlkopf werden durch einen Schwingungsgenerator dargestellt, welcher am Ende einer Röhre sitzt, die wiederum den nachgelagerten Vokaltrakt nachbildet. Die Schwingungen, welche durch unseren Schwingungsgenerator erzeugt werden, sind in Lautstärke und Tonhöhe veränderbar. Diese, mehr oder weniger primitive Nachbildung, ist für Vokale zwar größtenteils ausreichend, würde allerdings für die Darstellung von Nasallauten einige Abzweigungen von der Röhre benötigen, was wiederum die Berechnungen um ein Vielfaches verkomplizieren würde. Linear predicition cepstral coefficients (LPCC) bauen auf LPC features auf, indem sie diese als Input nehmen und anschließend eine LPC parameter conversion durchführen.

2.3 MFCC

Bei der Benutzung von Mel frequency cepstral coefficients (MFCC) wird versucht das menschliche Gehör zu replizieren, unter der Annahme, dass das selbige ein zuverlässiger Spracherkenner ist. Das Verfahren funktioniert, indem man das Frequenzspektrum eines Signals in mehrere Filter (critical bands) aufteilt, welche unter 1000Hz linear und darüber logarithmisch verlaufen. Hierzu ist wichtig zu sagen, dass der Mensch deutlich besser darin ist feinste Frequenzunterschiede in niedrigen Frequenzen herauszuhören, als in sehr hohen, weshalb es mehr Filter in niedrigen Frequenzen gibt als in hohen. Diese sind anhand der natürlichen Frequenzgruppen in der Auswertung von Signalen beim Menschen verteilt. Alle Filter überschneiden sich mit den benachbarten, was dafür sorgt, dass phonetisch wichtige Teile eines Signals erhalten bleiben. Anschließend wird ein Verfahren namens discrete cosine transform (DCT) auf den Daten angewandt.

2.4 PLP

Perpetual linear prediction (PLP) basiert auf der barkscale, welche wiederum auf den ersten 24 critical bands basiert. Daher ist eine Ähnlichkeit zum MFCC Verfahren nicht abzustreiten, da sie beide die Mel scale benutzen, um mit Hilfe von critical bands die feature extraction durchführen. Dies heißt aber natürlich auch, dass PLP ebenfalls versucht das menschliche Gehör zu simulieren. Allerdings ist PLP nicht nur eng an der Mel scale orientiert. Stattdessen wird, wie der Name schon sagt, eine linear prediction genutzt. Das dient unter anderem dem spectrum smoothing, bei dem es darum geht, Störgeräusche sowie nicht zu erklärende Frequenzextreme aus dem Signal herauszubekommen. So ist PLP eine Kombination aus spektral Analyse und linear prediction. Durch die Benutzung von verschiedenen Teilen des Speech processing, kommt es hier allerdings auch eine erhöhte Anzahl an Verfahren, die angewendet werden müssen, um die fertigen features zu erhalten. In diesem Fall müssen nach der Inverse discrete Fourier transform (IDFT) noch sowohl eine linear prediction analysis, als auch eine cepstral analysis durchgeführt werden.

3. Vergleich

3.1 Rechenkosten

Die Kosten einer Berechnung beziehen sich in praktisch jedem Fall auf die Menge an Rechenressourcen, welche aufgewendet werden müssen, um eine Berechnung durchzuführen. Hierbei ist auf Sprachverarbeitung, vor allem im Sinne von Echtzeitspracherkennern, wichtig, dass die Zeit der Berechnung auch in Echtzeit vorgenommen werden kann, um den tatsächlichen Nutzen auch zu erfüllen.

Im Kontrast zu den anderen feature extraction Methoden muss bei der BNF extraction dazu gesagt werden, dass zusätzlich zum pretraining des HMMs noch das Training des MLP für die feature extraction kommt. Welches auch durch pretraining deutlich effizienter wird [15]. Weiterhin sind BNFs relativ rechenaufwändig, da sowohl Training als auch die Menge an Preprocessingschritten deutlich höher ist als bei den anderen. Trotzdem gab es in diesem Gebiet auch ein paar Fortschritte, da zum Beispiel die HLDA Matrix kleiner wurde [16]. Darauf aufbauend lohnt es sich allerdings BNFs und cepstral features zusammen zu benutzen, da diese komplimentäre Informationen liefern können [17].

LPC und LPCC muss man hier für tatsächlich auseinander halten, da LPCs eine relativ hohe Berechnungsgeschwindigkeit haben [22]. LPCCs hingegen nehmen noch zusätzliche Berechnungen vor, welche die Rechenzeit verlängern.

Bezüglich der cepstral features selbst: MFCCs sind an schon seit ihrer Einführung eine der „state-of-the-art“ Technologien [20]. Dies liegt unter anderem an der relativ geringen Anzahl an Schritten die zur Gewinnung der MFCC features nötig sind, die in Zusammenarbeit mit der hohen Erfolgsquote geringe Rechenkosten relativ zum Ergebnis erzeugen.

Es ist davon auszugehe, dass PLPs wahrscheinlich aus der Reihe der feature extraction Methoden die langsamste ist, da sie die Technologie mit den meisten Zwischenschritten ist. Allerdings sind dazu keine validen Forschungsergebnisse öffentlich verfügbar.

3.2 Genauigkeit

Die Studie der Quelle [17] kommt zu dem Schluss, dass ein BNF System, welches 63 Stunden trainiert wurde auf eine Word Error Rate (WER) von knapp 24% kommt, wenn der Outputvektor 39 Dimensionen hat. Generell scheint die WER laut selbiger Quelle immer um 25% zu liegen für BNF Systeme. Hierfür scheint ein MLP mit 3 Hidden Layern ideal zu sein[16]. Dies gilt allerdings nur ab einer bestimmten feature Vektor Größe. Gleichzeitig wird eigentlich immer ein PLP System als Vergleich angegeben, welches in nahezu allen durchgeführten Tests bis auf wenige Prozent an die WER des BNF Systems herankommt, dieses aber nicht übertrifft. Was BNF Systeme angeht, variieren allerdings die WERs minimal. Zum Beispiel spricht [14] von ~30% Sentence Error Rate (SER). Weiterhin wurde gezeigt, dass BNFs in Zuverlässigkeit mit cepstral features mithalten können und diese sogar bei Zeiten übertreffen[16].

MFCCs produzieren akkurate features für Spracherkennung und speziell für Sprecheridentifikationsaufgaben sind MFCC Systeme sehr gut[21]. Allerdings fällt die Effizienz auf höheren Frequenzen deutlich, da die Mel scale diese nur noch mit geringerer Intensität repräsentiert.

LPC gilt eigentlich fast seit seiner Erfindung als zuverlässig[2]. Besonders ist die Effizienz der feature extraction hervor zu heben[22]. Es ist in seiner Rolle als schnelles und genaues Tool zur feature extraction immer noch relevant und findet immer noch verbreitete Anwendung.

PLP ist relativ spezialisiert auf speaker-independant Information, in dieser Hinsicht verbessert es den Ansatz von LPC etwas, da auch Bark Filter(vgl. 2.4) in die Verarbeitung mit einfließen[23]. Dies vereint praktisch Eigenschaften der Mel scale und der linear prediction.

3.3 Verlässlichkeit

Bei Verlässlichkeit beziehe ich mich primär auf die Noise Resistance und die generelle Verlässlichkeit.

BNFs sind an sich relativ zuverlässig darin Muster in Sprache zu erkennen. Allerdings bedarf es dafür sehr vorsichtigem (Pre-)Training(25). Dies liegt dem Fakt zu Grunde, dass auch geringe Abweichungen in den Gewichten innerhalb von Neural networks drastisch unterschiedliche Ergebnisse folgen können.

Einer der bekanntesten Schwächen von MFCCs ist wohl, dass sie auf Grund der Mel scale mit hohen Frequenzen und viel Hintergrund-/Störgeräuschen schlecht umgehen. Abgesehen davon ist das System sehr zuverlässig und relativ leicht effizient zu implementieren[24].

LPCs haben insgesamt eine vergleichbar hohe Widerstandsfähigkeit gegen Störgeräusche, ganz im Gegenteil zu PLP Verfahren[23].

4. Conclusion

In dieser Arbeit wurden einige aktuelle feature extraction Techniken mit einander und mit der BNF extraction verglichen. Dabei wurde zunächst eine kurze Übersicht über die einzelnen Technologien vermittelt, anschließend wurden vermehrt die Gesichtspunkte Rechenkosten, Genauigkeit und Verlässlichkeit betrachtet.

Letztlich ist zu sagen, dass verhältnismäßig alte Systeme wie MFCCs oder speziell LPC immer noch zu den effizientesten und zuverlässigsten gehören. Auch wenn Neural Network Ansätze immer besser werden. Ich fände es interessant zu sehen, inwiefern das bald geschieht und wie dann Lösungen entstehen, die Systemen ablösen, welche zum Teil bereits seit über 30 Jahren state-of-the-art sind.

Zusätzlich wäre es gut, wenn z.B. für die Rechenkosten kontextunabhängige Studien vorhanden wäre, da es so sehr schwierig ist die verschiedenen Methoden objektiv zu vergleichen.

Mein Recherchieren für diese Arbeit verlief mittelmäßig, da es zwar eine Unmenge an Quellen in jede erdenkliche Richtung gibt, aber es schwierig sein kann wirklich relevante Informationen in Quellen zu finden, welche auch verständlich sind.

5. Quellen(Bitte entschuldigen sie, dass einige der Quellen noch nur links sind, ich finde es aktuell leicht damit zu arbeiten, werde aber natürlich bis zu Endabgabe das noch in Ordnung bringen):

1. https://link.springer.com/content/pdf/bbm%3A978-3-319-17163-0%2F1.pdf
2. https://web.archive.org/web/20110624005456/http://otolith.com/otolith/olt/lpc.html
3. https://arxiv.org/ftp/arxiv/papers/1305/1305.1145.pdf
4. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.433.7519&rep=rep1&type=pdf
5. https://d1wqtxts1xzle7.cloudfront.net/40023802/Feature_Extraction_Methods_LPC__PLP_and_MFCC.pdf?1447603451=&response-content-disposition=inline%3B+filename%3DFeature_Extraction_Methods_LPC_PLP_and_M.pdf&Expires=1598452595&Signature=LHOKPcX2AbNoI-GOiWju6EqnPH6lLALrnX9Tawo-bT3Md~bL9o6dgCcatMTkuDrw2Ru26o156mNejs0c3SjLIjSjHdzZGKqW6robF1BfK3aY2WT7Rk9S8-VVG1FmOAk3fEpkeRGWCMyrdhZj8Gm4cpU0mbaV-5~r~onDqYlFKVEuiY~f4fBUWO3ziegtuBM4dKi7xfL82U797vM4YW16KAO5peeXSkZLmxsZbh-SYOP23lS1VMi0FwsyFP6mdPQlhgpA8kmZ1Y~2mDQcxzbug49bxyVBZTXAg3B4GBQ-b0zMMVjirNPrNRl1X4tzzAdVi0FMHD4mdY9vI0vq0zzkaw__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA
6. https://www.cs.toronto.edu/~gdahl/papers/deepSpeechReviewSPM2012.pdf

7. Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition – Namrata Dave 2013
8. https://www.researchgate.net/publication/3532051_RASTA-PLP_speech_analysis_technique
9. http://www.ee.iisc.ac.in/people/faculty/prasantg/downloads/PLP_modeling_speech_Mar29_2019.pdf
10. Kumar P, Chandra M. Speaker identification using Gaussian mixture models. MIT Inter-national Journal of Electronics and Communication Engineering. 2011;1(1):27-30
11. https://ccrma.stanford.edu/~jos/bbt/
12. https://www.researchgate.net/publication/221487754
13. https://mitpublications.org/yellow_images/1301463239_logo_journal6.pdf
14. https://www.isca-speech.org/archive/archive_papers/interspeech_2011/i11_0237.pdf
15. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.570.4694&rep=rep1&type=pdf
16. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4518713
17. F. Gr´ezl et al.,"Probabilistic and bottle-neck features forLVCSR of meetings," inICASSP'07, Hononulu, 2007
18. https://doi.org/10.1371/journal.pone.0100795
19. http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/
20. http://ismir2000.ismir.net/papers/logan_paper.pdf
21. Ravikumar KM, Reddy BA, Rajagopal R, Nagaraj HC. Automatic detection of syllablerepetition in read speech for objective assessment of stuttered Disfluencies. In: Proceed-ings of World Academy Science, Engineering and Technology. 2008. pp. 270-273
22. Othman AM, Riadh MH. Speech recognition using scaly neural networks. World academyof science. Engineering and Technology. 2008;38:253-258
23. Hermansky H. Perceptual linear predictive (PLP) analysis of speech. The Journal of theAcoustical Society of America. 1990;87(4):1738-1752
24. Narang S, Gupta MD. Speech feature extraction techniques: A review. International Jour-nal of Computer Science and Mobile Computing. 2015;4(3):107-114
25. https://www.isca-speech.org/archive/archive_papers/interspeech_2011/i11_0237.pdf