Stage de recherche
Master 2 Acoustique, Traitement du signal,
Informatique, Appliqués à la Musique
Année universitaire 2016-2017

Extraction de données audio d'un corpus de cris d'enfants, avec applications à la psychologie du développement du langage

Mémoire de stage - Stage non confidentiel

Andrea Vaglio

Ircam - Équipe Perception et Design Sonores

Encadrants : Jean-Julien Aucouturier (CNRS/IRCAM), en collaboration avec Kazuo Okanoya (Université de Tokyo, Japon)

1er Mars - 30 Juillet 2017

Abstract

In this internship, we apply machine learning techniques to extract automatic descriptions from a large corpus of audio recordings of human baby cries. The main goal of the internship is to generate accurate acoustic metadata for each of the individual cry in the corpus and see how these acoustic characteristics co-vary with annotated context for the cries (for example if the infant is hungry or feels alone). The secondary goal of the internship is to provide the scientific community with open-source Python code to allow other researchers to test further scientific hypotheses about the development of the babies' linguistic abilities.

After implementating a expiration/inspiration segmentation algorithm, we extracted a range of static and dynamic descriptors from expired parts of the cries. We then studied the statistical variation of these characteristics with the context. In order to validate the perceptual relevance of some of the extracted descriptors extracted, we also conducted a free-sort psychological experiment on a sample of N=11 female participants.

Keywords : Baby cries, audio data mining, machine learning, cognitive science, baby communication, audio pattern recognition, acoustic metadata.

Résumé

Ce rapport de stage de fin d'étude propose l'application d'un ensemble de techniques d'apprentissage automatique afin de générer une description automatique d'un corpus d'enregistrements audio de cris de bébés. Le but du stage est de se doter des technologies permettant d'extraire, pour chaque cri, un ensemble de métadonnées acoustiques pertinentes et d'observer comment ces caractéristiques acoustiques co-varient avec les contextes de production du cri (par exemple, si l'enfant a faim ou se sent seul). Le but secondaire du stage est de fournir à la communauté scientifique un code Python open-source afin de permettre à d'autres chercheurs de tester d'autres hypothèses à propos du développement des aptitudes linguistiques chez l'enfant.

Après avoir conçu un algorithme de segmentation des cris en phases expiration/inspiration, nous avons extrait des parties criées (expirations) un ensemble de descripteurs statiques et dynamiques. Nous avons ensuite étudié statistiquement la variation de ces caractéristiques avec le contexte du cri. Afin de valider la pertinence perceptive des descripteurs statiques extraits, nous avons également réalisé une expérience psychologique, avec un paradigme de tri libre, sur un échantillon de N=11 mamans.

Mots clés : Cris de bébés, exploration de données audio, apprentissage automatique, science cognitive, communication de l'enfant, reconnaissance de motifs audio, métadonnée acoustique.

REMERCIEMENTS

Je tiens tout d'abord à remercier Jean-Julien Aucouturier pour m'avoir accueilli dans son équipe et pour son accompagnement tout au long du stage.

Par ailleurs un grand merci à l'équipe "Perception et Design Sonores" et particulièrement à Pablo Arias et Louise Goupil pour leur aide précieuse.

Je remercie également toutes les mamans qui ont bien voulu donner de leur temps pour passer des expériences.

Merci enfin à l'ensemble des permanents, doctorants, stagiaires de l'IRCAM, pour la bienveillance et la bonne humeur qui règne au sein du laboratoire.

TERMINOLOGIE

ATIAM : Acoustique, Traitement du signal, Informatique, Appliqués à la Musique
PDS : Perception et Design Sonores
IRCAM : Institut de Recherche et Coordination Acoustique Musique
CREAM : Cracking the Emotional Code of Music
MIR : Music Information Retrieval
MFCC : Mel-Frequency Cepstral Coefficients
GMM : Gaussian Mixture Model
HMM : Hidden Markov Model
SVM : Support Vector Machine
SWIPE : Sawtooth Waveform Inspired Pitch Estimator
CLARA : Clustering for LARge Applications
CLARANS : Clustering LARge Applications based on RANdomized Search
ANOVA : ANalysis Of VAriance
LME : Linear Mixed-Effects models
HNR : Harmonics-to-Noise Ratio
NHR : Noise-to-Harmonics Ratio
DDL : Degrés De Liberté
FWER : FamilyWise Error Rate
DTW : Dynamic Time Warping
GLME : General Linear Mixed-Effects models

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION

Présentation générale

Ce document a pour but de présenter les travaux effectués dans le cadre de mon stage de fin d'études Acoustique, Traitement du signal, Informatique, Appliqués à la Musique (ATIAM) de l'université Pierre et Marie Curie. Ce stage s'est déroulé au sein de l'équipe Perception et Design Sonore (PDS) dans les locaux de l'Institut de Recherche et Coordination Acoustique Musique (IRCAM). Le but du stage est d'étudier les caractéristiques acoustiques des cris de jeunes enfants (de la naissance à 1 an), en fonction du contexte du cri. Le code implémenté tout au long du stage est disponible, sous forme de notebook commenté (mais sans les ressources), sur github [9].

Cadre du stage

L'IRCAM [3] est un centre français associé au centre Pompidou de recherche scientifique, d'innovation technologique et de création musicale, fondé par Pierre Boulez en 1969 et dirigé depuis 2006 par Frank Madlener. C'est aujourd'hui l'un des plus grands centres de recherche publique au monde se consacrant à la création musicale et à la recherche scientifique. L'IRCAM est de nature polymorphe : c'est un institut de recherche, de production, mais aussi un lieu de formation, de consultation et d'expérimentation. L'institut s'investit dans trois grands champs de recherche : musicale, sonore et scientifique. L'IRCAM a pour mission fondamentale de susciter une interaction féconde entre recherche scientifique, développement technologique et création musicale contemporaine. Cette articulation constitue, depuis sa fondation il y a quarante ans, le principal axe structurant l'ensemble de ses activités. L'un de ses enjeux majeurs est de contribuer, par les apports des sciences et techniques, au renouvellement de l'expression artistique.

Les travaux qui y sont menés sont le fruit de la recherche de plusieurs équipes : Espaces Acoustiques et Cognitifs, Perception et Design Sonores, Analyse et Synthèse des Sons, Systèmes et Signaux Sonores, Audio/Acoustique, instruMents, Représentations Musicales, Analyse des Pratiques Musicales et Interaction Son Musique Mouvement.

Le projet de l'équipe Perception et Design Sonores, au sein de laquelle le stage a été effectué, porte sur la perception et la cognition des sons en combinant des connaissances en psychoacoustique, en traitement/synthèse du signal, en psychologie et en neurosciences cognitives. Les travaux concernent aussi bien la caractérisation perceptive des sons que les mécanismes cognitifs mis en jeu pour les identifier. L'équipe a étendu l'objet de ses recherches, initialement basé sur les sons environnementaux, aux imitations vocales, aux sons musicaux et à la voix afin de mieux comprendre les processus cognitifs mis en jeu, d'une part, par l'identification d'une source sonore, et d'autre part, par le traitement émotionnel des sons. Ce stage s'inscrit plus particulièrement dans le projet Cracking the Emotional Code of Music (CREAM) [8] qui se donne pour objectif de produire les technologies et les connaissances permettant de caractériser comment les signaux sonores (musique ou parole) activent les mécanismes cérébraux émotionnels et sociaux.

Figure 0.1 – L'Institut de Recherche et Coordination Acoustique Musique

Contexte et travaux pré-existants dans l'équipe

Le thème de ce stage résulte de l'intérêt du projet CREAM pour l'expression des émotions et des intentions communicatives dans le son de la voix. Le travail proposé consiste à analyser les caractéristiques acoustiques et perceptives d'un grand corpus d'enregistrements de cris de bébés, pré-existant au stage et collecté en collaboration avec le laboratoire de Biolinguistique de Kazuo Okanoya (université de Tokyo), collaborateurs de l'encadrant Jean-Julien Aucouturier depuis 2008 [7]. Le corpus a été collecté auprès de 31 familles japonaises volontaires, de 2008 à 2010, à qui l'équipe a prêté de petits enregistreurs audio portables (de type ZOOM H1) et demandé d'enregistrer de courts épisodes de cris de leur nouveau-né, et ce, le plus régulièrement possible, de la naissance à un an d'âge. Après chaque enregistrement, la mère de l'enfant doit annoter ce qu'elle estime être le contexte du cri, en choisissant entre une série de "besoins" pré-identifiés : celui d'être nourri ("hungry"), d'être changé ("pee"), de s'endormir ("sleepy"), et d'interagir avec un parent ("lonely"). Le corpus résultant de cette campagne de collecte de données comprend un total de plus de 1700 enregistrements audio, pour un total de plus de 14 heures de cris de bébés, tous annotés avec leur contexte et l'âge du bébé les ayant produits.

L'enjeu scientifique de l'étude d'un tel corpus est de mieux comprendre les régularités acoustiques des cris de bébés, et en particulier comment celles-ci interagissent avec les caractéristiques physiques de l'enfant (âge et sexe) et le contexte du cri. Les cris des jeunes enfants, tout comme ceux des nouveaux-nés animaux non-humains, sont considérés en biologie comme des signaux communiquant une détresse aux adultes pourvoyeurs de soins qui peuvent alors éliminer ou amoindrir les conditions aversives qui ont donné lieu au cri [42]. Or, après 18 mois de développement en moyenne, l'enfant humain prononce son premier mot [50]. Ce qui se passe entre le premier cri et le premier mot, en particulier l'évolution acoustique du cri dans la première année de vie, a donc le potentiel important de révéler les mécanismes sous-tendant le développement du langage articulé et de la cognition sociale.

Un premier travail réalisé dans l'équipe avant le démarrage du stage a consisté à développer un algorithme de segmentation automatique des enregistrements, afin d'en isoler les portions de cri expirées et inspirées du fond sonore (les enregistrements étant réalisés dans un environnement domestique relativement calme, mais non-contrôlé). Les caractéristiques des expirations et des

inspirations étant très différentes [55], il est, en effet, nécessaire de procéder à une segmentation. L'algorithme développé, basé sur une architecture de reconnaissance de formes avec apprentissage d'une partie du corpus annotée à la main, a donné lieu à publication dans la revue JASA en 2011 [12]. Le présent stage étend cette première étape de l'analyse du corpus en réimplémentant cet algorithme dans le langage Python (l'original étant en Matlab), en y incorporant quelques innovations algorithmiques récentes, et en répliquant (voire en fait, nous le verrons, en améliorant) ses bons résultats sur le corpus. Ce travail est présenté dans le Chapitre 1 de ce mémoire.

Une fois segmentés, les cris peuvent ensuite être analysés pour en extraire leurs caractéristiques acoustiques statiques (durée, hauteur moyenne, etc.), et ces caractéristiques peuvent être confrontées statistiquement à l'âge et au contexte du cri. Ce travail avait seulement été ébauché avant le stage, en particulier à cause de difficultés d'analyse statistique d'un corpus présentant de nombreuses valeurs manquantes. Le présent stage réalise cette analyse de façon approfondie, en utilisant le formalisme statistique émergent des modèles linéaires à effets mixtes. Ce travail est présenté dans le Chapitre 2 de ce mémoire.

Un troisième travail pré-existant dans l'équipe consiste en l'analyse d'un certain nombre de profils prosodiques dans les phases d'expirations criées. Nonaka *et al* font, en effet, l'hypothèse de la nature de marqueur linguistique des contours de hauteurs observés dans les phases d'expirations [52]. Ce travail, réalisé jusqu'à présent d'une façon nécessitant une importante intervention de catégorisation manuelle, n'a été mené que par intermittence depuis 2012 par l'équipe japonaise, et seuls certains résultats préliminaires ont donné lieu à présentations en congrès [53]. Le présent stage étend considérablement cette dernière étape en développant un algorithme de clustering automatique des profils prosodiques du corpus, en le validant de façon expérimentale auprès d'un échantillon de mamans, et en l'utilisant pour analyser systématiquement l'utilisation de ces profils en fonction de l'âge et du contexte du cri. Cette contribution, sans doute la plus substantielle du stage, est présentée dans le Chapitre 3 de ce mémoire.

État de l'art

L'analyse informatisée d'enregistrements de cris de bébés constitue un domaine de recherche relativement bien exploré depuis une vingtaine d'années, et où convergent les intérêts des communautés académiques du traitement du signal audio et reconnaissance de formes [54, 56], de la psychologie cognitive et du développement [45, 60], de l'éthologie [63], de la pathologie pédiatrique [42, 21] et même de l'évolution de la musique [70].

Deux types d'approche dominent l'état de l'art. D'une part, un certain nombre de travaux, principalement issus de la communauté informatique et d'ingénierie biomédicale, traite le sujet comme un problème de plus pour les technologies de la classification automatique : il s'agit pour ceux-ci de construire des algorithmes de reconnaissance de formes capables, après apprentissage, de catégoriser de grands corpus de cris en fonction de leur contexte d'émission ou des caractéristiques de l'enfant. On retrouve dans ces travaux toute la sophistication rencontrée habituellement dans des communautés comme la Music Information Retrieval (MIR). Ntalampiras [54], par exemple, utilise des caractéristiques acoustiques comme les Mel Frequency Cepstral Coefficients (MFCCs), des représentations à base d'ondelettes ou de modulations spectro-temporelles, et des algorithmes d'apprentissage comme les Gaussian Mixture models (GMMs), Hidden markov models (HMMs) et les Support Vector Models (SVMs), pour obtenir 95% de précision de classification cross-validée sur 5 classes. Une application souvent proposée pour ces travaux est celle du diagnostic automatique dans un cadre clinique : Alaie [10] par exemple, propose une architecture à base d'arbres de décision pour catégoriser les cris à la naissance en 2 classes, saines et pathologiques (pour une revue médicale de l'utilisation diagnostique du cri de bébé, voir aussi [42]). Une difficulté avec ces travaux est qu'ils ont un potentiel de découverte scientifique limité : même si on montre qu'un algorithme sait classifier des cris joyeux ou tristes, affamés ou douloureux, sains ou malades, on ne sait souvent pas comment interpréter les projections non-linéaires très optimisées de caractéristiques acoustiques sensées les représenter, ni si ces caractéristiques sont en fait utilisées sciemment par les enfants, ni même perçus par les parents. (pour une discussion approfondie de ces difficultés, voir par exemple [11]).

L'autre partie de l'état de l'art considère les cris non pas comme un signal à catégoriser de façon automatique et relativement agnostique, mais plutôt comme un comportement dont on veut fournir

une caractérisation acoustique précise, afin d'en comprendre les bases cognitives ou physiologiques. Il peut s'agir, par exemple, de caractériser les paramètres acoustiques de cris observés dans certaines pathologies (fréquence fondamentale f0 moyenne plus basse dans le cadre de la trisomie 21, variations de f0 plus élevées dans le cadre de lésions cérébrales [42]) ou bien, comme on le fait ici, dans certains contextes de cri (cri de douleur lors de la première vaccination [21]; cri de faim lorsque le parent prépare à manger à la maison [63]). Contrairement aux approches d'apprentissage automatique, ces travaux sont souvent caractérisés par des corpus d'enregistrements relativement réduits en taille ou en nombre d'enfants. Par exemple, Scheiner [63] analyse un nombre conséquent de cris (16,322), mais provenant de seulement 7 enfants, et n'ayant qu'un seul enregistrement par enfant dans les 2 premiers mois de vie. Une autre caractéristique de ces travaux est d'utiliser souvent des outils pré-existants comme le logiciel d'analyse phonétique PRAAT [4] ou le logiciel de biolinguistique Sound Analysis Pro [5]. Ces outils, s'ils ont le mérite de démocratiser l'analyse acoustique auprès de communautés scientifiques (par exemple psychologiques ou médicales) qui n'en auraient pas autrement l'expertise, restreint la plupart des analyses à un petit nombre de caractéristiques (fréquence fondamentale (f0), durée, intensité, et degré d'harmonicité), souvent résumées par leurs principales statistiques descriptives (moyenne, écart-type, maximum, minimum). Ces caractérisations, par contre, sont souvent utilisées pour faire des arguments scientifiques importants : Scheiner [63], par exemple, montre que le développement du contrôle nerveux sur la zone subglottale à partir du 3e mois se traduit par une augmentation de l'harmonicité de la voix et une diminution des variations de fréquence fondamentale ; Mampe [45] montre que les cris nouveaux-né français et allemands ont des caractéristiques de hauteur et d'intensité commune avec la langue maternelle des parents, démontrant ainsi une forme d'apprentissage acoustique intra-utérin.

Approche proposée

Le présent travail, même s'il utilise en partie des algorithmes d'apprentissage (pour la phase de segmentation, puis pour le clustering de profils prosodiques), s'inscrit résolument dans la lignée de cette deuxième catégorie de travaux, et (contrairement au précédent travail de Mael Derio à l'IRCAM sur les cris de bébés) ne se donne pas du tout l'objectif de catégoriser automatiquement les cris du corpus. Au travers d'algorithmes existants (comme PRAAT par exemple) et à développer (segmentation expirations/inspirations et clustering de profils prosodiques), l'ambition est d'aider à mieux comprendre les signaux communicatifs de l'enfant, sans objectif immédiat d'application technologique ou clinique. Il se distingue de l'état de l'art de deux façons importantes :

D'une part, avec ses 31 bébés et plus de 1700 cris analysés, il se situe dans la limite haute, voire encore inédite, de l'état de l'art. Cette opportunité nous permet de prétendre, peut-être, à une plus grande représentativité des résultats ou une plus grande puissance statistique de découverte de patterns qui auraient pu passer inaperçus dans la littérature. À l'inverse, elle nous oblige également à un effort de sophistication des outils statistiques (des modèles linéaires à effets mixtes permettant de traiter le relativement grand nombre d'individus comme facteur aléatoire), et d'optimisation algorithmique (l'algorithme CLARA/CLARANS pour le clustering des profils prosodiques de dizaines de milliers de cris).

D'autre part, seul un petit nombre de contributions récentes s'est intéressé aux profils prosodiques à l'intérieur des cris (intonations montantes, descendantes, etc.) et, le cas échéant, l'ont fait de façon manuelle ou quasi manuelle. Scheiner [63], par exemple, définit 11 types de cris expirés ("babble","laugh", "moan", etc.), identifiés de façon manuelle (*"an exclusively mathematical procedure, such as cluster analysis, was not helpful to establish call types"*), mais pour lesquelles ils établissent des limites paramétriques leur permettant ensuite d'annoter le corpus de façon automatique. D'une façon encore plus qualitative, Chittora [22], observant un petit corpus de spectrogrammes de cris, identifie de façon heuristique 10 modes de cris, dont il évalue ensuite la répartition de façon qualitative dans plusieurs pathologies infantiles (voir figure 0.2).

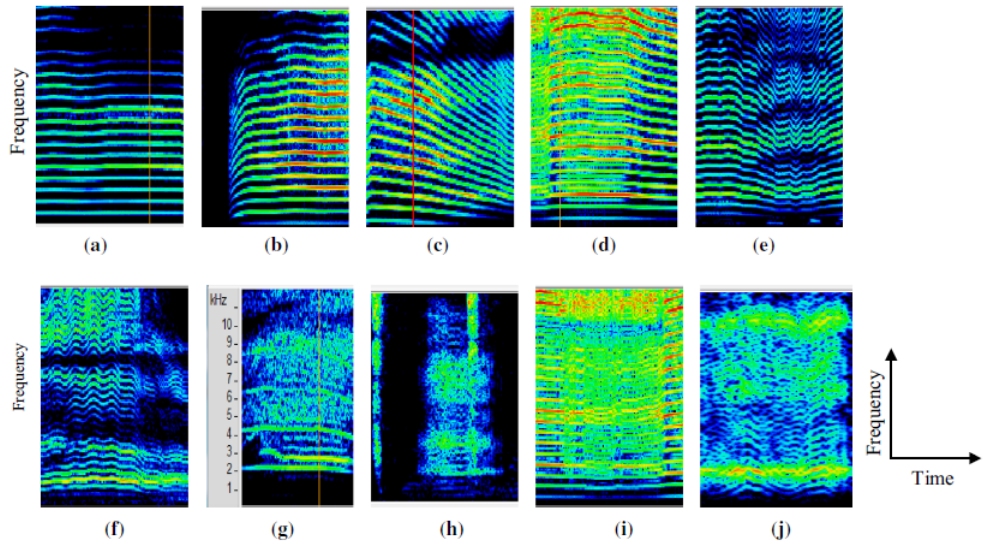


Figure 0.2 – 10 modes de cris identifiés de façon heuristique par Chittora [22]. Avec ici : **a** f0 plat, **b** f0 croissant, **c** f0 décroissant, **d** présence de sous harmoniques, **e** roulement glottal, **f** vibration, **g** hyperphonation, **h** Phonation inspiratoire, **i** Dysphonation, **j** faible vibration

De façon encore plus simplifiée, Wermke [70], suivi par Mampe [45] semble postuler seulement 4 types de cris (falling, rising, symetric et plateau - voir Figure 0.3), dont il est affirmé d'autorité qu'ils "prédominent dans le répertoire de cri des nouveaux-nés sains". Si le choix heuristique de certains types de prosodies présente l'intérêt d'un certain contrôle théorique (des intonations choisies peuvent être par exemple comparées à des intonations similaires dans le répertoire adulte), il présente également le risque d'un manque de représentativité (Mampe [45], par exemple, fait le choix de ne considérer que les cris en U inversé, et pas les autres, présentés comme rarissimes; cependant Chittora [22], sur les 10 modes considérés, n'identifie pas de cris en U inversé), et d'une certaine hétérogénéité des critères de regroupement rendant l'interprétation difficile. Par exemple, parmi les 10 modes identifiés par Chittora [22], 6 sont décrits par des considérations mélodiques (f0 plate, montante, descendante, au début, à la fin, etc.), 1 de façon statistique (f0 trop haute) et 3 de façon timbrales (présence de subharmoniques ou de bruit large-bande). Le présent travail diffère de cet état de l'art en ce qu'il propose un algorithme automatique de clustering des profils prosodiques prototypiques rencontrés dans le corpus. Cette méthode, outre le fait qu'elle n'a jamais été proposée dans le contexte des cris de bébés, peut être considéréecomme plus objective et plus facile à interpréter par la suite. Cependant, afin d'en justifier l'utilisation scientifique, il importera de valider perceptivement les résultats de cet algorithme avant de les utiliser pour analyser le corpus: il ne faudrait pas, en effet, retomber dans le travers de l'approche de catégorisation automatique en étudiant la répartition dans le corpus de profils prosodiques qui seraient, en fait, indistinguables à l'oreille par les bébés ou les parents.

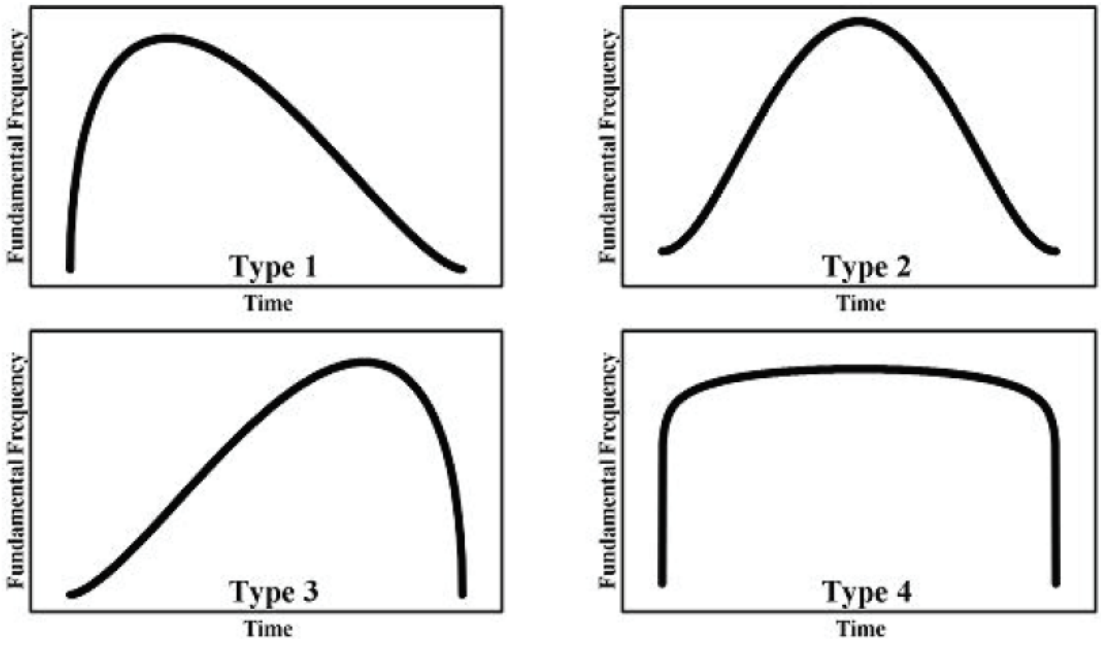


Figure 0.3 – 4 modes de cris postulés par Wermke [70]. Avec ici : **Type 1** falling, **Type 2** symetric, **Type 3** rising, **Type 4** plateau

Objectifs

À la lumière de cette revue de la littérature et du contexte pré-existant dans l'équipe, nous résumons ici les objectifs du stage :

Objectifs techniques :

- Ré-implémentation Python et amélioration d'un algorithme de segmentation de phases de cris expirées et inspirées
- Extraction de caractéristiques acoustiques statiques, bien comprises de la communauté, et test de leur association statistique avec les caractéristiques des bébés et du contexte d'émission, dans le formalisme des modèles linéaires à effets mixtes
- Développement d'un nouvel algorithme de clustering de profils prosodiques, découverte des profils les plus représentés sur le corpus et test de leur association statistique avec les caractéristiques des bébés et des contextes

Objectifs scientifiques :

- Caractériser les changements acoustiques des cris en fonction de l'âge et du sexe des bébés, et du contexte des cris
- Validation expérimentale de l'algorithme de clustering de profils prosodiques
- Etudier l'éventuelle association statistique des profils prosodiques avec le contexte du cri.

Objectifs de dissémination :

- Développer les outils nécessaires pour publier ce corpus et ses annotations (segments, caractéristiques acoustiques, clusters prosodiques) sous une forme "augmentée", avec un code Python open-source capable de régénérer les annotations automatiquement, afin de les rendre utilisables par l'ensemble de la communauté scientifique.
- Éventuelle préparation d'un manuscrit (type JASA) sur l'algorithme de clustering prosodique et sa validation expérimentale

Plan du rapport

Le rapport se présente en trois chapitres distinctes. Tout d'abord, nous détaillerons, dans le chapitre 1, l'algorithme de segmentation mis en place, les améliorations apportées et les résultats de segmentation obtenus. Puis, dans le chapitre 2, nous étudierons l'association statistique d'un ensemble de caractéristiques statiques pertinentes avec les caractéristiques des bébés et des contextes. Enfin, dans le chapitre 3, nous présenterons un algorithme de clustering automatique de profils prosodiques, l'expérience de tri libre proposée à un échantillon de mamans et les résultats obtenus puis nous étudierons l'utilisation de ces profils en fonction de l'âge et du contexte du cri.

La conclusion permettra de mettre en avant le travail accompli et les perspectives d'évolutions futures qu'amène ce travail.

Chapitre 1

SEGMENTATION AUTOMATIQUE DES SONS D'EXPIRATION ET D'INSPIRATION

Le corpus servant de base à ce travail est constitué d'enregistrements audio effectués sur 31 enfants japonais. Les expirations et inspirations ont été annotées manuellement (par une collaboratrice japonaise, Yulri Nonaka, avec le logiciel Sound Analysis Pro) pour trois des enfants (numéros 044, 050 et 051) dans trois contextes de cri différents (hungry, pee, sleepy). Ces annotations permettront la constitution d'une base d'entraînement qui servira à segmenter automatiquement les 28 autres enfants. Les trois classes considérées sont les suivantes : les expirations voisées (un cri est dit voisé si sa production s'accompagne de la vibration quasi-périodique des cordes vocales), les inspirations (voisées ou non) et les silences/fond sonore. Les annotations manuelles identifient les points de départ et de fin d'une phase d'expiration et d'inspiration. Pour faciliter la lecture du document, on parlera dans la suite du rapport des classes 'EX', 'IN' et 'SI'.

Le modèle présenté ici est un modèle de Markov dont la séquence des états est directement observable dans la base d'entraînement, et les probabilités d'émission sont modélisées par apprentissage par un modèle de mélange de gaussienne (GMM) ou en convertissant la décision de classification d'une machine à vecteur-support (SVM).

On présentera tout d'abord l'algorithme de segmentation automatique section 1.1. Le choix des paramètres de simulations est ensuite discuté section 1.2. Finalement, après présentation des différents algorithmes testés et du formalisme de la cross-validation section 1.3, les résultats seront analysés et discutés en section 1.4.

1.1 Explication de la méthode

L'algorithme de segmentation est divisé en deux parties indépendantes. Tout d'abord, une décision est prise indépendamment pour chaque trame temporelle. Puis la décision est corrigée en prenant en compte la probabilité de transition entre les différentes classes. Cette deuxième étape permet de poser des contraintes, indirectement, sur la durée passée dans chaque état, et de corriger ainsi des séries d'annotations correspondant à des étapes 'EX' ou 'IN' trop rapides ou trop longues, qui ne seraient pas physiologiquement plausibles.

1.1.1 Extraction des MFCCs

La première étape nécessaire pour la classification de chaque trame est le choix de descripteurs pertinents pour la voix. On a extrait pour chacune des trames un ensemble de MFCC [61]. Les coefficients cepstraux sont calculés par une transformée de Fourier appliquée au logarithme du module de la transformée de Fourier du signal. Les coefficients cepstraux c_n sont donnés par :

$$c_n = \frac{1}{2\pi} \int_{\omega=-\pi}^{\omega=\pi} \log S(\omega) \exp j\omega n \mathrm{d}\omega \tag{1.1}$$

Dans le cas des MFCCs, les bandes de fréquence de ce spectre sont espacées logarithmiquement selon l'échelle de Mel [61]. Cette transformation non-linéaire de l'énergie permet d'approximer la résolution fréquentielle non-linéaire du système auditif humain. Ce descripteur est connu comme une représentation compacte du "timbre" sonore [44, 66], qui semble être un bon niveau de description pour séparer différents types de sons vocaux, expirés ou inspirés.

1.1.2 Classification par GMM ou SVM

Nous avons considéré deux types de modèles pour effectuer la première étape de classification :

- Un modele discriminatif : un SVM multi-classe
- Un algorithme de maximum a posteriori basé sur des modèles géneratifs : des GMMs

Ces techniques sont dites de type supervisé car utilisant un ensemble de données déjà étiqueté (ici les classes 'EX','IN' et 'SI').

Un GMM est une somme de M densités gaussiennes simples pondérées telle que :

$$p(x_i) = \sum_{m=1}^{M} \pi_m \mathcal{N}(x_i, \mu_m, \Sigma_m) \tag{1.2}$$

avec x_i vecteur d'entraînement, ici des MFCCs, $\mathcal{N}$ fonction de densité de probabilité gaussienne de moyenne μ_m et de matrice de covariance Σ_m, et π_m coefficient de mélange. Les paramètres des GMMs sont estimés de façon classique, par la procédure d'estimation-maximisation [61]. On entraîne un GMM pour chacune des classes, on considère les classes comme étant équiprobables. Chaque GMM produit alors, pour un vecteur inconnu x_i, une estimation de la vraisemblance $p(x_i)$, le GMM donnant la vraisemblance la plus forte est alors choisi comme classe (équivalent dans ce cas, d'une distribution uniforme des classes, à un classifieur basé sur la règle du maximum a posteriori).

Un SVM [67] évalue une fonction de décision qui permet de définir l'appartenance de chaque trame à une classe. Il permet la séparation de données non linéairement séparables en transformant l'espace des données d'entraînement en un espace de plus grande dimension dans lequel il existe un hyperplan séparateur linéaire qui maximise la marge entre les classes (afin de garantir la généralisation du problème à des nouvelles données). La marge est la distance entre la fonction de séparation et les échantillons les plus proches de chaque classe.

L'entraînement du SVM est rendu possible sans spécifier la fonction de transformation θ, qui est souvent inconnue, mais seulement le produit vectoriel $K(x_i, x_j) = \theta(x_i)^T \theta(x_j)^T$, que l'on appelle fonction noyau. Les fonctions noyaux permettent de transformer un produit scalaire dans un espace de grande dimension, ce qui est coûteux en temps de calcul, en une évaluation ponctuelle d'une fonction. Cette technique est appelée le "kernel trick".

Le SVM étant un classifieur binaire, il existe plusieurs méthodes pour l'étendre au cas multi-classe. On utilisera dans notre cas la méthode *one-versus-one* [17].

1.1.3 Modèle de Markov

Un modèle de Markov (ou chaine de Markov) [36] est défini par :

- Un ensemble d'état : $\{s_1, s_2, s_3, ..., s_N\}$
- Une matrice de probabilité de transition M tel que $M = (a_{i,j})_{1 \leq i,j \leq N}$ avec $a_{i,j} = P(s_i, s_j)$
- Un ensemble de probabilité initiale $\{\pi_i = P(s_i), i \in [1, N]\}$

Une chaine de Markov possède la propriété de Markov : l'information utile pour la prédiction du futur est entièrement contenue dans l'état présent du processus et n'est pas dépendante des états antérieurs. De manière plus formelle : $P(s_{ik+1}|s_{ik}, ..., s_1) = P(s_{ik+1}|s_{ik})$ avec ici s_{ik} état à l'instant $t = k$.

La séquence des états définit l'enchaînement des états produit par le modèle. La probabilité initiale définie la probabilité du système d'être dans l'un des états à l'instant $t = 0$.

Dans notre case, contrairement à un modèle de Markov caché, la séquence des états est directement observable dans la base d'entraînement : un état correspond à une des trois classes ('EX', 'IN' et 'SI') obtenues en sortie du classifieur. La matrice de probabilité de transition est donc obtenue directement en comptant les occurrences des transitions entre les états dans la base d'apprentissage déjà annotée. On considère les probabilités initiales comme étant équiprobables.

1.1.4 Correction par algorithme de Viterbi

L'utilisation de l'algorithme de Viterbi [69] permet, avec la connaissance préalable de la probabilité d'émission de chaque état à chaque instant et de la matrice de probabilité de transition entre états, de déterminer l'enchaînement d'états le plus probable à partir d'une séquence d'états observée.

Les probabilités d'émissions sont estimées à l'étape précédente par GMMs ou en appliquant une technique de Platt scaling [59] en sortie du SVM. L'utilisation de la technique de Platt scaling [59] permet de transformer la décision de classification du SVM en une densité de probabilité a posteriori pour chacune des classes. On a alors :

$$\mathrm{P}(classe|x) = 1/[1 + \exp(Af(x) + B)] \tag{1.3}$$

avec $f(x)$ fonction de décision du SVM et x vecteur d'entraînement. Les paramètres A et B sont des constantes à estimer. L'algorithme de Viterbi est présenté dans l'Annexe B.

La figure 1.1 nous permet de comparer les annotations manuelles d'un cri d'enfant de 30 secondes, la décision de classification trame-à-trame par SVM et l'enchaînement des classes obtenu après correction de la décision de classification prise précédemment par algorithme de Viterbi. On observe que la plupart des erreurs commises par le classifieur trame-à-trame sont peu plausible d'un point de vue physiologique (transitions multiples et rapides entre classes) et sont correctement corrigées par l'algorithme de Viterbi.

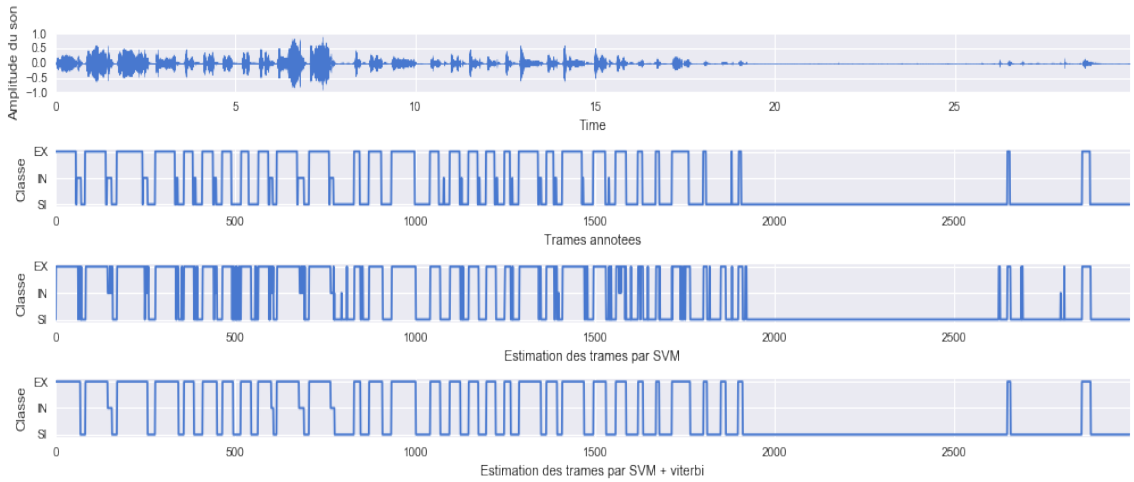


Figure 1.1 – Comparaison entre un signal audio d'un cri d'enfant de 30s, les annotations manuelles de ce dernier, la classification du signal trame par trame par SVM entraînée sur l'ensemble des trames annotées et la classification obtenue après correction par algorithme de Viterbi

1.2 Implémentation et paramètres de simulations

Nous avons utilisé plusieurs bibliothèques Python pour l'implémentation de l'algorithme :

- `librosa` : une bibliothèque pour la fouille de données musicales (MIR) qui permet notamment l'extraction des MFCCs [47],
- `scikit-learn` : une bibliothèque de machine-learning [57],
- `matplotlib` : une bibliothèque pour l'affichage [35],
- `scipy` et `numpy` : des bibliothèques de calcul scientifique [37],
- `pandas` : une bibliothèque pour la gestion de grands tableaux de données [48]

L'algorithme de Viterbi a quant à lui été implémenté manuellement en s'inspirant de l'implémentation matlab de Dan Ellis [26].

Pour l'extraction des MFCCs, nous avons choisi une fenêtre de Hanning de 20 ms avec un pas de 10ms. Nous avons également normalisé ces derniers, la normalisation de la base d'entraînement étant indispensable pour le bon fonctionnement du SVM. Dans le cas des GMMs, on comparera les résultats avec et sans normalisation.

La fonction noyau choisie pour la classification par SVM est le noyau gaussien, défini tel que :

$$K(x_i, x_j) = \exp(-\gamma||(x_i - x_j)||^2), \quad \gamma > 0 \tag{1.4}$$

avec x_i et x_j vecteurs d'entraînement, ici des MFCCs, et γ paramètre du noyau. Les hyper-paramètres C [23] (paramètre de pénalisation) et γ du SVM sont choisis par recherche exhaustive sur un sous-ensemble de l'espace des paramètres sur laquelle on applique une procédure de validation croisée à 10 échantillons (10-fold CV). Le paramètre C est un paramètre de régularisation permettant de prévenir du phénomène de sur-apprentissage (un cas dit de sur-apprentissage, ou *overfitting* est un cas où les performances du système sont très bonnes sur la base d'apprentissage, mais mauvaises sur de nouveaux vecteurs. Cela signifie que le système a bien appris, mais qu'il est incapable de généraliser son fonctionnement). Dans notre cas, on choisira $C = 1$ et $\gamma = 0.1$.

Les paramètres A et B de la fonction de Platt scaling sont estimés par une méthode de maximum de vraisemblance sur le même ensemble d'entraînement. Une validation croisée à 5 échantillons de la base d'entraînement [41] est également utilisée pour l'estimation de la sortie du modèle probabilistique afin d'éviter l'overfitting. Contrairement aux hyper-paramètres C et γ qui sont fixés, le modèle probabilistique est appris pour chaque base d'entraînement.

Dès les tests préliminaires, il est apparu que l'utilisation de matrice de covariance diagonale pour les GMMs donne de meilleurs résultats que l'utilisation de matrices de covariance complètes. Cela s'explique par le peu de données d'entraînement disponibles pour la classe 'SI'. En effet, dans le cas de base d'entraînement trop peu fournie l'utilisation de matrices complètes peut entraîner, pour le GMM, l'apparition d'un phénomène d'overfitting. On utilisera 20 itérations de l'algorithme k-means pour l'initialisation et 50 itérations de l'algorithme EM. On comparera les résultats en considérant 5 gaussiennes par GMMs ou 20.

1.3 Calcul de l'accuracy par cross-validation

On cherche donc à estimer les performances de dix algorithmes différents :

1. Classification par SVM multi-classe, avec et sans Viterbi, MFCCs normalisés;
2. Classification par GMMs de 5 composantes, avec et sans Viterbi, MFCCs normalisés ou non.
3. Classification par GMMs de 20 composantes, avec et sans Viterbi, MFCCs normalisés ou non.

Pour l'estimation des performances des algorithmes, on utilise différentes bases de données, chacune issue d'un seul enfant et d'un seul contexte. Les différents fichiers sont mélangés avant extraction aléatoire d'un ensemble de trames consécutives. Notons qu'il est nécessaire de conserver des trames qui se suivent dans le temps pour pouvoir appliquer l'algorithme de Viterbi.

On extrait tout d'abord 40 000 trames consécutives pour 6 couples (enfant, contexte) différents sur lesquels on applique l'ensemble des algorithmes. Puis, on extrait 30 000 trames consécutives sur 8 couples différents qu'on applique sur les meilleurs algorithmes donnés par l'étape précédente. On ignore le contexte 'pee' pour l'enfant 51 par manque de données.

Pour estimer la fiabilité des différents algorithmes, on utilise la méthode de validation croisée à 10 échantillons. La méthode est la suivante :

1. Division de la base de données en 10 sous-ensembles
2. Entraînement sur les 9 sous-ensembles et test sur le dernier
3. Répétition de la précédente étape sur les 9 sous-ensembles restants
4. Calcul de la précision moyenne

La validation croisée permet d'éviter de sur-estimer la précision dans des cas d'overfitting, en donnant une meilleure estimation de la précision du modèle généralisée à une base de données indépendante de la base d'entraînement (dans le cas d'un problème réel, une base de donnée inconnue).

1.4 Résultats

Number	Context	SVM norm	SVM+vtb norm	GMM5	GMM5+vtb	GMM5 norm	GMM5+vtb norm	GMM20	GMM20+vtb	GMM20 norm	GMM20+vtb norm
44	Hung	0.8735	0.889075	0.810275	0.836075	0.79815	0.853325	0.79995	0.839025	0.80765	0.870325
44	Pee	0.869875	0.89385	0.810075	0.83785	0.8099	0.862225	0.819375	0.85675	0.822775	0.8808
44	Sleepy	0.8821	0.898825	0.80965	0.797975	0.8049	0.868525	0.8126	0.832475	0.805675	0.870475
50	Hung	0.895875	0.9136	0.837975	0.878275	0.827425	0.891675	0.858525	0.89795	0.843725	0.897525
50	Pee	0.884625	0.901425	0.80565	0.880475	0.759375	0.872075	0.857725	0.913025	0.819675	0.906925
50	Sleepy	0.926418	0.946158	0.879593	0.910999	0.86809	0.928946	0.895008	0.915646	0.896888	0.939552

Table 1.1 – Résultats de ten-fold CV avec plusieurs algorithmes sur 6 bases de données de 40000 trames (norm : MFCCs normalisés, vtb : avec Viterbi)

Number	Context	SVM+vtb norm	GMM5+vtb norm	GMM20+vtb norm
44	Hung	0.874967	0.8507	0.849533
44	Pee	0.893367	0.8493	0.869267
44	Sleepy	0.892	0.8546	0.859767
50	Hung	0.8726	0.8431	0.860933
50	Pee	0.906333	0.872467	0.909567
50	Sleepy	0.945627	0.91916	0.919514
51	Hung	0.876133	0.843167	0.8372
51	Pee	nan	nan	nan
51	Sleepy	0.8613	0.8289	0.842133

Table 1.2 – Resultats de ten-fold CV avec plusieurs algorithmes sur 8 bases de données de 30000 trames (norm : MFCCs normalisés, vtb : avec Viterbi).

Le tableau 1.1 présente les résultats pour 6 bases de données et l'ensemble des algorithmes. Il est assez clair que les algorithmes donnant des meilleurs résultats sont ceux corrigés par l'algorithme de Viterbi avec une accuracy moyenne de 88.2% contre 83.9%. On remarque également que l'utilisation de plus de gaussiennes dans le cas de GMMs change peu les résultats, avec un léger bonus pour des modèles plus complexes ($M = 84.33\%$ pour tous les algorithmes basé sur un GMM5 contre $M = 86.08\%$ pour tous les algorithmes basé sur un GMM20). Toujours dans le cas des GMMs, la normalisation des MFCCs a peu d'impact ($M = 84.97\%$ sans normalisation contre $M = 84.88\%$ avec normalisation). Globalement, la classification par SVM corrigé avec l'algorithme de Viterbi présente les meilleurs résultats avec une moyenne d'accuracy de 90.71%.

Dans la deuxième étape, on se restreint à l'utilisation d'algorithmes utilisant des MFCCs normalisés et corrigés par Viterbi. Les résultats pour 8 bases de données sont présentés dans le tableau 1.2. Le meilleur algorithme est, de nouveau, l'algorithme SVM+Viterbi norm. En effet, ce dernier permet d'atteindre une moyenne d'accuracy de 89.02%, contre 85.76% pour GMM5+vtb norm et 86.84% pour GMM20+vtb norm.

Il est intéressant de constater que, contrairement aux résultats de JJ.Aucouturier dans [12], l'algorithme SVM+Viterbi donne de meilleurs résultats que les algorithmes GMMs+Viterbi. On constate en effet une augmentation de l'accuracy moyenne de SVM+Viterbi qui passe de 85.62% à 89.02%. Les résultats précédemment publiés pour les algorithmes GMM+Viterbi, quant à eux, sont équivalents à ceux trouvés dans notre étude (86.3% pour 5GMM et 86.4% pour 20GMM). L'amélioration des résultats obtenue ici pour les modèles SVMs, par rapport à ceux publiés précédemment, peut vraisemblablement s'expliquer par l'utilisation d'une 5-fold cross validation pour l'estimation du modèle probabilistique en sortie du SVM, ce qui n'était pas le cas dans [12]. En effet, les propriétés fortement discriminantes du SVM impliquent des probabilités hautement contrastées entre les classes en sortie du Platt scaling en l'absence de cross-validation, ce qui peut empêcher la correction par l'algorithme de Viterbi même en cas de probabilité de transition peu probable entre deux états. A contrario, la classification par GMMs donne des densités de probabilité d'appartenance à une classe moins contrastées, le but de ce type de modèle n'étant pas de maximiser la discrimination entre les classes, mais simplement la vraisemblance d'appartenance des données pour la distribution de chaque classe. L'utilisation de la cross-validation permet donc de rendre moins contrastées les différences de probabilités entre les classes dans le cas du SVM et peut expliquer cette amélioration.

Cette première partie a donc permis de sélectionner le modèle SVM+Viterbi pour faire l'apprentissage de l'ensemble des trames annotées disponibles (ensemble des bébés et ensemble des contextes), et segmenter le reste, non annoté, du corpus. On peut observer la comparaison entre les annotations manuelles du fichier et la classification obtenue par le modèle final dans la Figure 1.1. L'accuracy sur la base d'entraînement du modèle final est de 89.02%.

Chapitre 2

EXTRACTION DES CARACTÉRISTIQUES STATIQUES DES CRIS, ET ÉTUDE STATISTIQUE DE LEUR VARIATION AVEC LE CONTEXTE

Dans cette second partie, on génère, à partir des différents segments estimés par l'algorithme précédemment appliqué, un ensemble de caractéristiques statiques pertinentes (durée des expirations/inspirations et hauteur moyenne par exemple) afin de regarder comment ces caractéristiques acoustiques peuvent être utilisées pour produire les contextes annotés des cris. On veut donc décrire les invariants acoustiques statiques entre les différents contextes de cris, en modélisant leur dérive au cours du temps et les différences individuelles. On utilise pour cela plusieurs méthodes statistiques comme l'analyse de variance (ANOVA) [27] ou les modèles linéaires à effets mixtes (LME) [68].

On présentera tout d'abord les différentes caractéristiques statiques choisies dans la section 2.1. On introduira ensuite, section 2.2, le formalisme relatif au test ANOVA, au LME et à la correction de Bonferroni. Puis, après présentation des choix des paramètres de simulation (section 2.3), les résultats seront analysés et discutés en section 2.4.

2.1 Choix des caractéristiques statiques

La première caractéristique choisie est la durée moyenne des segments. Les caractéristiques des expirations et des inspirations étant très différentes [55], on considérera la durée moyenne des inspirations et des expirations de manière séparée. Comme l'ensemble des segments estimés étant constitué très majoritairement d'expirations, on extrait seulement les autres caractéristiques par rapport aux cris expirés.

La deuxième caractéristique choisie est la hauteur moyenne (pitch) des expirations. Elle pourra être mesurée de deux manières différentes dont on comparera les résultats par la suite : avec l'algorithme SWIPE et avec le logiciel PRAAT.

L'algorithme Sawtooth Waveform Inspired Pitch Estimator (SWIPE), présenté par Camacho dans [19], estime le pitch comme la fréquence fondamental d'une forme d'onde en dents de scie dont le spectre correspond le mieux au signal d'entrée. PRAAT [4] est un logiciel libre scientifique gratuit conçu pour la manipulation, le traitement et la synthèse de sons vocaux (phonétique), qui estime le pitch par une méthode d'autocorrélation [18].

Le logiciel PRAAT nous permet également d'extraire un certain nombre d'autres caractéristiques relatives à la voix que l'on peut classer en quatre groupes :

- les caractéristiques relatives à la hauteur (le maximum du pitch, le minimum etc)
- des mesures du rapport signal-sur-bruit comme le Harmonics-to-Noise Ratio (HNR) et le Noise-to-Harmonics Ratio (NHR)
- un ensemble de diverses mesures du shimmer (vshimmer-abs, shimmer-loc-dB, shimmer-apq3, shimmer-apq5, shimmer-apq11)
- un ensemble de diverses mesures du jitter (jitter-loc-abs, jitter-rap, jitter-ppq5)

Le calcul des différentes mesures de shimmer, de jitter et de rapport signal sur bruit est détaillés [65]. Ces trois caractéristiques sont relatives à la rugosité et l'enrouement présent dans la voix. Le jitter et le shimmer sont des mesures des variations cycle par cycle respectivement de la fréquence fondamentale et de l'amplitude du signal. Le HNR, quant à lui, est le rapport entre le contenu harmonique et le contenu bruité du signal vocal. Le bruit aléatoire contenu dans le signal est la conséquence de la fermeture irrégulière ou asymétrique du conduit vocal. Un meilleur contrôle du conduit vocal entraîne alors une augmentation du HNR et une diminution du shimmer et du jitter. Ce qui se traduit par une diminution de l'enrouement de la voix.

2.2 Explication des différents tests statistiques effectués

Un test statistique est une démarche consistant à rejeter ou à ne pas rejeter une hypothèse statistique, appelée hypothèse nulle, en fonction d'un jeu de données. L'hypothèse nulle est une hypothèse postulant l'égalité entre des paramètres statistiques (comme par exemple la moyenne) de deux échantillons dont elle fait l'hypothèse qu'ils sont pris sur des populations équivalentes. On calcule alors en fonction d'un modèle statistique approprié, une p-valeur qui correspond à la probabilité d'obtenir avec ce modèle, une différence au moins égale à celle observée. Le test est alors dit significatif si la p-valeur est inférieure à un seuil α choisi. Nous rejetons l'hypothèse nulle (cad qu'il est peu probable que les deux populations d'origine ait la même statistique) si le test est déclaré significatif. Nous ne rejetons pas l'hypothèse nulle (cad qu'il est peu probable que les deux populations d'origine soient différentes) si le test est non significatif.

Les différents tests statistiques présentés dans cette section sont implémentés sous le langage R [64], un langage de programmation spécialisé dans l'analyse de données et les statistiques. On utilisera alors la librairie `rpy2` [2] qui permet l'intégration de R dans Python. Les fonction LME et ANOVA sont respectivement disponibles dans la librairie `lme4` et `stats` de R.

2.2.1 ANOVA

L'ANOVA est un test statistique permettant de vérifier que plusieurs échantillons sont issus d'une même population. L'ANOVA [27] teste l'effet d'une, ou plusieurs, variables prédictives catégoriques (appelées effets fixes ou facteurs) sur une variable continue dépendante (dans notre cas nos différentes caractéristiques statiques). Les variables catégorielles peuvent prendre un nombre fini de valeurs appelées niveaux. Nous supposons, dans ce test, que les populations suivent des distributions normales, qu'elles ont une variance égale et que les observations sont indépendantes. L'hypothèse nulle correspond au cas où les distributions suivent la même loi normale (cad qu'elles ont les mêmes moyennes). L'hypothèse alternative est qu'il existe au moins une distribution dont la moyenne s'écarte des autres moyennes.

Afin de simplifier le formalisme, on se place dans le cas d'une ANOVA à un seul facteur de variabilité (ou *one way*). On considère donc I échantillons Y_i d'effectifs n_i, issus des I populations qui suivent I lois normales $N(\mu_i, \sigma^2)$ de même variance. Chaque individu s'écrit $y_{i,j}$, avec $i \in [1, I]$ et $j \in [1, n_i]$. L'effectif total est $N = \sum_{i=1}^{I} n_i$. Le modèle est alors décrit par l'équation :

$$y_{i,j} = \mu + y_i + \epsilon_{i,j} \tag{2.1}$$

avec μ constante, y_i est la moyenne de la distribution i et $\epsilon_{i,j}$ erreur pour l'individu j de la population i.

La moyenne par échantillon et totale s'écrivent alors :

$$\begin{aligned} \overline{y_i} &= \frac{1}{n_i}\sum_{j=i}^{n_i} y_{i,j} \sim N(\mu_i, \frac{\sigma^2}{n_i}) \\ \overline{y} &= \frac{1}{N}\sum_{i=1}^{I}\sum_{j=i}^{n_i} y_{i,j} \sim N(\mu, \frac{\sigma^2}{N}) \text{ avec } \mu = \frac{1}{N}\sum_{i=1}^{I}(n_i\mu_i) \end{aligned} \tag{2.2}$$

La somme des carrés des écarts (égale à la variance à un facteur multiplicatif prés) peut être calculée simplement par la formule :

$$SCE_{total} = SCE_{facteur} + SCE_{residu} \tag{2.3}$$

avec ici, SCE_{total} variance totale, $SCE_{facteur}$ variance expliquée par le modèle (ou variabilité inter-classe), SCE_{residu} variance non expliquée par le modèle (ou variabilité intra-classe) qui sont définies tel que :

$$\begin{aligned} SCE_{facteur} &= \sum_{i=1}^{I}(\overline{y_i} - \overline{y})^2 \\ SCE_{residu} &= \sum_{i=1}^{I}\sum_{j=1}^{n_i}(y_{i,j} - \overline{y_i})^2 \end{aligned} \tag{2.4}$$

Par ailleurs, les DDL (degrés de liberté) sont aussi calculés. Ceux-ci représentent le nombre de variables aléatoires qui ne peuvent être déterminées ou fixées par une équation (notamment les équations des tests statistiques).

$$\begin{aligned} DDL_{facteur} &= \sum_{i=1}^{I-1} 1 = I - 1 \\ DDL_{residu} &= \sum_{i=1}^{I-1}(n_i - 1) = N - I \end{aligned} \tag{2.5}$$

Par hypothèse, la variable observée y_i suit une loi normale. La loi du χ^2 à k degrés de liberté étant définie comme étant la somme de k variables normales au carré, les sommes des carrés des écarts suivent les lois suivantes :

$$\begin{aligned} SCE_{facteur} &\sim \chi^2(DDL_{facteur}) \\ SCE_{residu} &\sim \chi^2(DDL_{residu}) \end{aligned} \tag{2.6}$$

Les variances s'obtiennent en faisant le rapport de la somme des carrés des écarts sur le nombre de degrés de liberté. La Loi de Fisher étant définie comme le rapport de deux lois du χ^2, le rapport des variances suit donc une Loi de Fisher :

$$F_{obs} = \frac{S^2_{facteur}}{S^2_{residu}} = \frac{\frac{SCE_{facteur}}{I-1}}{\frac{SCE_{residu}}{N-I}} \sim F(I-1, N-I) \tag{2.7}$$

Ce ratio représente donc le rapport entre la variance expliquée par le modèle et la variance qui n'est pas expliquée par le modèle. Dans le cas où les populations sont très différentes, on a un F_{obs} important. Si la valeur de F_{obs} est supérieure au seuil de rejet, on rejette l'hypothèse nulle : on conclut qu'il existe une différence statistiquement significative entre les distributions.

On définit alors la p-valeur p, probabilité d'avoir un F aussi grand sous l'hypothèse nulle H_0, tel que :

$$p = P_{H_0}(F > F_{obs}) \tag{2.8}$$

La valeur p montre explicitement si F_{obs} est supérieur ou non au seuil de rejet, en pratique, On considère que si $p < 0.05$ la différence est statistiquement significatif.

On se trouve ici dans le cas particulier d'un jeu de donnée à mesures répétées. On possède, en effet, plusieurs observations pour chaque bébé. On utilise alors, dans cas, une ANOVA à mesures répétées [31].

Le test ANOVA présente l'avantage d'être peu complexe. Cependant, il nécessite l'utilisation d'un design équilibré. Un design représente l'ensemble des combinaisons des différents facteurs étudiés. Un design équilibré est alors un design possédant au moins une observation (et le même nombre) par combinaison de niveau de facteurs. Cela signifie, par exemple, que pour le bébé numéro 11, il faut avoir autant d'observations annotées "hungry" le 1er mois que "sleepy" le 3e mois. Cela veut également dire que l'ANOVA n'est pas compatible avec des designs possédant des valeurs manquantes (par exemple, si un bébé ne possède pas d'enregistrement annotés "hungry" le 4e mois). On rendra alors le design équilibré en prenant la moyenne des observations pour chaque combinaison de facteurs et en interpolant linéairement les valeurs manquantes. Cela rend malheureusement peu fiable l'analyse effectuée, le corpus possédant un grand nombre de valeurs manquantes (principalement à partir du 4e mois). Une manière de résoudre ce problème est l'utilisation du modèle LME.

2.2.2 LME

présentation des modèles linéaires

Les modèles linéaires sont souvent utilisés pour modéliser la relation entre une variable y, appelée variable dépendante, et une (ou plusieurs) variable prédictive $X_1, \ldots, X_p$ (dans notre cas des variables catégoriques). Ces variables sont également appelées effets fixes.

Dans le cas d'un modèle simple à une seule variable explicative, on a :

$$y = \beta_0 + \beta_1 X_1 \tag{2.9}$$

Dans le cadre d'un modèle linéaire simple, on peut représenter graphiquement la relation entre x et y à travers un nuage de points. L'estimation du modèle linéaire permet de tracer la droite de régression. Le paramètre β_0 (appelé *intercept*) représente l'ordonnée à l'origine et β_1 le coefficient directeur (pente) de la droite.

Si on généralise :

$$\mathbf{y} = \mathbf{X\beta} + \boldsymbol{\epsilon} \tag{2.10}$$

avec ici : $\mathbf{y}$ vecteur d'observations indépendantes de taille $n * 1$, $\boldsymbol{\beta}$ vecteur des effets fixes de taille $p * 1$, $\mathbf{X}$ est une matrice de taille $n * p$ et $\boldsymbol{\epsilon}$ vecteur des termes d'erreurs aléatoires $n * 1$, les erreurs sont supposées indépendantes et identiquement distribuées selon une loi normale de moyenne 0 et de variance $\boldsymbol{\epsilon} \sim N(0, \sigma^2\mathbf{I})$, on en déduit alors que $\mathbf{y} \sim N(\mathbf{X\beta}, \sigma^2\mathbf{I})$. La matrice $\mathbf{X}$ est déterminée à partir de la formule du modèle et de la valeur des variables catégorielles.

Les modèles linéaires mixtes

Dans le cas d'un jeu de donnée à mesures répétées, les observations ne sont pas indépendantes. L'utilisation de modèles linéaires est donc impossible. Les modèles LMEs permettent d'étendre le modèle linéaire à l'utilisation d'observations non-indépendantes.

Dans un modèle à effets mixtes au moins une des variables prédictives est une variable catégorielle representant les unités d'observations (ce qu'on appelle un effet aléatoire, dans notre cas

l'identité du bébé), les autres variables sont des effets fixes. Cela nous permet de résoudre l'absence d'indépendance des données en considérant un intercept différent pour chaque sujet. Le modèle mixte permet d'estimer ces intercepts. Notre modèle est alors ce qu'on appelle un "random intercept model". Si on considère, par exemple, un modèle simple avec y pitch moyen des expirations et l'âge comme seule variable prédictive, l'intercept correspond au pitch moyen à la naissance. L'ajout d'un effet aléatoire permet alors de prendre en compte le fait que chaque enfant possède un pitch moyen propre à la naissance.

Cependant, ce modèle considère que, quels que soient les effets des variables prédictives, il est identique pour tous les sujets. Un modèle "random slope model" permet alors de considérer, pour un effet fixe, un coefficient directeur de la droite de régression différent pour chaque sujet. On parle alors de coefficient directeur aléatoire pour l'effet. En reprenant l'exemple précèdent, cela permet de prendre en compte que le pitch moyen des expirations de chaque enfant va évoluer de manière différente avec l'âge.

De manière formelle :

$$\mathbf{y} = \mathbf{X\beta} + \mathbf{Z\Upsilon} + \boldsymbol{\epsilon} \tag{2.11}$$

avec ici : $\mathbf{y}$ vecteur des observations de taille $n * 1$, $\boldsymbol{\beta}$ vecteur des effets fixes de taille $p * 1$, $\boldsymbol{\Upsilon}$ vecteur des effets aléatoires de taille $q * 1$, $\mathbf{X}$ est une matrice de taille $n * p$, $\mathbf{Y}$ est une matrice de taille $n * p$ et $\boldsymbol{\epsilon}$ vecteur d'erreurs de taille $n * 1$.

Les matrices $\mathbf{X}$ et $\mathbf{Z}$ sont déterminées par la formule du modèle et les variables prédictives.

Nous faisons l'hypothèse que les effets aléatoires et les termes d'erreurs sont décorrelés et sont normalement distribués de moyenne nulle. On a donc :

$$cov(\boldsymbol{\epsilon}, \boldsymbol{\Upsilon}), \boldsymbol{\epsilon} \sim N(0, \mathbf{R}) \ , \boldsymbol{\Upsilon} \sim N(0, \mathbf{G}) \tag{2.12}$$

avec ici $\mathbf{G}$ et $\mathbf{R}$, matrice de covariance respective de $\boldsymbol{\Upsilon}$ et $\boldsymbol{\epsilon}$. Alors le vecteur d'observation est normalement distribué tel que : $\mathbf{y} \sim N(\mathbf{X\beta}, \mathbf{ZGZ^T} + \mathbf{R})$. On remarque que les erreurs ne sont plus requises d'être indépendantes. Les équations de Henderson [33] peuvent être utilisées pour trouver $\hat{\boldsymbol{\beta}}$ le 'best linear unbiased estimator' (BLUE) de $\boldsymbol{\beta}$, et $\hat{\boldsymbol{\Upsilon}}$ le 'best linear unbiased predictor' (BLUP) de $\boldsymbol{\Upsilon}$.

Les p-valeurs pour les modèles mixtes ne sont pas aussi simples à définir que dans le cas du test ANOVA. Si on veut tester l'effet d'un effet précis, on compare un modèle complet (avec l'effet fixe en question) contre un modèle réduit (sans l'effet fixe en question). On se retrouve alors dans le cas d'un test du χ^2, qui nous permet de déterminer une p-valeur (démontrer que la comparaison des deux modèles revient à effectuer un test du χ^2 n'est pas trivial, pour plus d'informations voir [71]).

Le test LME présente l'avantage, contrairement au test ANOVA, de ne pas nécessiter l'utilisation d'un design équilibré. Il gère donc également les valeurs manquantes. Cependant il est nettement plus complexe que le test ANOVA et nécessite strictement plus d'observations que de degrés de liberté.

2.2.3 Cas des comparaisons multiples : comparaison de Bonferonni

Le "problème des comparaisons multiples" advient lorsque l'on considère simultanément un ensemble de tests statistiques, les résultats des problème des procédures statistiques standards peuvent alors devenir faussés. En effet, si de multiples hypothèses sont testées, le risque d'observer un événement rare augmente, et en conséquence, la probabilité de rejeter une hypothèse nulle de manière incorrecte (ce qu'on nomme erreur de type I) augmente. Ce problème peut arriver, par exemple, si l'on teste l'hypothèse nulle pour le pitch moyen des expirations et en même temps pour la durée des expirations. Une des méthodes possibles pour compenser ce problème est la correction de Bonferroni (du nom du mathématicien italien Carlo Emilio Bonferroni).

La correction de Bonferroni compense cette possible sur-évaluation de la significativité statistique de chaque test en testant chaque hypothèse individuelle à un seuil de rejet de $\frac{\alpha}{m}$, où α est le seuil de rejet global voulu et m le nombre d'hypothèses. Par exemple, si un test teste $m = 20$ hypothèses

et $\alpha = 0.05$, alors la correction de Bonferroni consiste à tester chaque hypothèse individuellement au seuil $\frac{0.05}{20}$, au lieu de $\alpha = 0.05$.

De manière plus formelle $H_1, .., H_m$ famille d'hypothèses et $p_1, .., p_m$ leur p-values correspondantes, avec ici, m le nombre total d'hypothèses nulles et m_0 le nombre réel d'hypothèses. Le FamilyWise Error Rate (FWER) est la probabilité de rejeter au moins un vraie H_i c'est-à-dire de faire une erreur de type I. Alors on obtient avec l'inégalité de Boole :

$$\text{FWER} = P\big\{ \cup_{i=0}^{m_0} (p_i \leq \frac{\alpha}{m})\big\} \leq \sum_{i=0}^{m_0} \big\{P(p_i \leq \frac{\alpha}{m})\big\} = m_0(\frac{\alpha}{m}) \leq m(\frac{\alpha}{m}) = \alpha \tag{2.13}$$

On constate que le FWER sera inférieur ou égale à α donc que la probabilité de faire une erreur de type I sera inférieure ou égale à α. Cela veut donc dire que le test peut être conservatif, c'est à dire que la probabilité de faire une erreur de type II augmente. Une erreur de type II correspond au fait de ne pas rejeter une hypothèse nulle fausse. Dans le cas d'un FWER strictement inférieur à α, on perd donc en puissance statistique. Il est alors possible que le test ne détecte pas une différence significative. Il peut notamment être montré que la correction de Bonferonni est conservative quand les tests sont dépendants. Dans ce cas, on regroupe ensemble les tests que l'on considère de la même famille (par exemple, le pitch moyen et le pitch maximum des expirations ne comptent que pour un).

2.3 Choix des paramètres de simulation

2.3.1 Choix des effets testés et description des modèles mis en place

L'ANOVA à mesures répétées est constituée de trois effets fixes : le mois, le contexte et le sexe. On considère également l'effet des différentes interactions possibles entre les trois effets fixes. L'interaction du mois et du contexte seraient significative, par exemple, pour la durée moyenne des expirations si les bébés présentaient une durée moyenne plus forte pour le contexte hungry le 3e mois par rapport aux autres durée moyenne des couples (contexte, mois). On définit ici la variable sexe comme une 'Between-subject variables', c'est-à-dire une variable dont différents groupes de sujets sont utilisés pour chaque niveau (un bébé ne possédant qu'un sexe). Inversement, on définit les variables mois et contexte comme des 'within-subjects variables' comme chaque sujet est testé pour chaque niveau du facteur.

Le test LME sera constitué de trois effets fixes : le mois, le contexte et le sexe et d'un effet aléatoire : l'identité du bébé. On prendra également en compte l'effet des différentes interactions possibles entre les trois effets fixes. On considérera un coefficient directeur aléatoire pour l'effet contexte, l'effet mois et l'interaction entre le mois et le contexte. Afin de diminuer la complexité du modèle, on ne considérera pas que l'intercept et les différents coefficients directeurs (un pour chaque effet fixe) de chaque bébé sont corrélés (cette approximation est du même ordre que celle faite au chapitre 1 en passant des matrices de correlations complètes à diagonales pour l'apprentissage des GMMs).

2.3.2 Choix des différents Design

On se restreindra à l'étude de trois contextes de cris différents : hungry, pee et sleepy.

L'ANOVA nécessitant d'interpoler les valeurs manquantes du design choisi, on s'est tout d'abord restreint aux bébés possédant le minimum de valeurs manquantes. De même, on a seulement considéré les mois possédant le moins de valeurs manquantes possible. On se restreindra donc à l'étude de 13 bébés sur les 4 premiers mois avec interpolation des valeurs manquantes dans le cas de l'ANOVA.

Pour le cas du LME, on n'a pas besoin d'interpoler les valeurs manquantes. On va donc pouvoir utiliser le design précédent sans interpoler les valeurs manquantes. On va également pouvoir considérer plus de bébés et plus de mois (tant que le nombre de dégrées de liberté est strictement inférieur au nombre d'observations).

ANOVA	mean_EX	mean_maxPitch_Praat	mean_shimmer_apq3	mean_nhr	mean_hnr
month	F(3,36)=11.27, p=2.33e-05	F(3,36)=5.09, p=0.00486	F(3,36)=5.111, p=0.00476	F(3,36)=5.874, p=0.00226	F(3,36)=8.715, p=0.000177
context	0.196	0.198	0.259	0.172	0.129
month:context	0.466	0.864	0.267	0.354	0.323

LME - Design 1 : ANOVA	mean_EX	mean_hnr
month	X2(1)=10.2455, p=0.00137	X2(1)=8.9703, p=0.002744
context	0.27087	0.397449
month:context	0.52970	0.561447

LME - Design 2 : 4 months and all babies	mean_EX
month	X2(1)=9.7481, p=0.001795
context	0.033780
month:context	0.590165

LME - Design 3 : 8 months and all babies	mean_EX	mean_IN	mean_jitter_loc_abs	mean_shimmer_apq5	mean_nhr	mean_hnr
month	X2(1)=19.4310, p=1.043e-05	X2(1)=15.9655, p=0.001675	X2(1)=8.5720, p=0.003414	1.00000	X2(1)=16.8320, p=4.084e-05	X2(1)=26.4811, p=2.661e-07
context	0.2328	0.896772	0.482464	X2(2)=12.3412, p=0.00209	0.7561	0.9113
month:context	0.2072	0.944877	0.651606	0.21698	0.2589	0.2851

Figure 2.1 – Valeur de p pour différentes caractéristiques statiques selon plusieurs effets fixes dans le cas de tests ANOVA et LME selon plusieurs design d'expérience. Ici les effets statistiquement significatifs sans correction de Bonferroni sont colorés en jaune et ceux significatifs avec correction de Bonferroni sont colorés en orange

On a donc 4 designs différents pour 2 tests différents tel que :

- Design ANOVA : 3 contextes, 13 bébés, 4 premiers mois, interpolation valeurs manquantes
- Design LME 1 : 3 contextes, 13 bébés, 4 premiers mois
- Design LME 2 : 3 contextes, tous les bébés, 4 premiers mois
- Design LME 3 : 3 contextes, tous les bébés, 8 premiers mois

2.4 Résultats

Présentation des résultats

La totalité des résultats sont donnés en annexe C et D. Il est intéressant de constater que les résultats pour le pitch moyen estimé par SWIPE et par PRAAT sont très similaires. Le facteur sexe n'ayant pas d'influence sur les deux modèles, on l'exclura des deux modèles. On présente ici les modèles qui ne l'incluent pas. Afin d'alléger le rapport, on ne présentera ici que les résultats significatifs. Le tableau allégé est présenté figure 2.1. Les paramètres et valeurs des différents tests statistiques sont indiqués seulement dans le cas d'effet statistiquement significatifs après correction de Bonferroni.

On considère dans notre cas 6 groupes de caractéristiques : la durée des expirations, la durée des inspirations, les caractéristiques statiques relatives à la hauteur des expirations, les caractéristiques statiques relatives au jitter des expirations, les caractéristiques statiques relatives au shimmer des expirations, les caractéristiques statiques relatives au rapport signal/bruit des expirations. En appliquant la correction de Bonferroni, on considère donc les valeurs significative pour $p < (0.05/6) = 0.00083$.

Analyse et discussions

Tout d'abord, il est intéressant de noter que le facteur sexe n'a pas d'influence statistiquement significative sur les caractéristiques. Fuller met également en avant l'absence d'influence du sexe sur certaines de ces caractéristiques (comme le jitter, shimmer et pitch) dans [29]. Ce qui semble cohérent avec l'absence de différence entre les sexes de morphologie et de dimension de l'appareil vocal chez les bébés, la différentiation entre les deux sexes ne commençant seulement à partir de 9 ans [43].

ANOVA	mean_EX	mean_maxPitch_Praat	mean_shimmer_apq3	mean_nhr	mean_hnr
month	↑	↑	↓	↓	↑

LME - Design 1 : ANOVA	mean_EX	mean_hnr
month	↑	↓

LME - Design 2 : 4 months and all babies	mean_EX
month	↑

LME - Design 3 : 8 months and all babies	mean_EX	mean_IN	mean_jitter_loc_abs	mean_shimmer_apq5	mean_nhr	mean_hnr
month	↑	↑		x	↓	↑
context	x	x	x	↑(pee)	x	x

Figure 2.2 – Sens de variations des effets statistiquement significatifs mis en avant pour plusieurs caractéristiques statiques. Dans le cas de l'effet "mois", ↑ indique une augmentation du paramètre correspondant avec l'âge. Dans le cas de l'effet "context", ↑ (contexte) indique une augmentation de ce paramètre pour le contexte précisé entre parenthèse. Inversement, ↓ indique une diminution du paramètre avec l'âge ou pour un contexte. Une croix indique une absence de variation significative

On observe également un effet principal du mois pour plusieurs descripteurs (comme par exemple la durée des expirations) et quasiment pas d'effet du contexte dans tous les cas (hormis sur une mesure du shimmer).

Les différents tests statistiques nous permettent de dire que la variation d'une caractéristique statique en fonction d'un effet est statistiquement significative, mais pas dans quel sens elle varie. Il est donc nécessaire de visualiser un par un la caractéristique en fonction de l'effet. La figure 2.2 détaille dans quel sens varie ces caractéristiques en fonction des effets significatifs, après visualisation des différentes courbes.

On analysera principalement les résultats donnés par le Design 3 par le LME car étant plus représentatif du corpus (plus de bébés, plus de mois, pas de valeurs interpolées). De plus les designs 1 et 2 mettent en avant des résultats également montrés par le design 3. On observe tout d'abord une augmentation très significative de la durée moyenne des expirations. La figure 2.3 met en effet en avant une augmentation de la valeur moyenne de 500 ms le premier mois jusqu'à plus de 700 ms le second mois (+40%). Cette augmentation peut être la conséquence de facteurs physiologiques comme l'augmentation de la capacité pulmonaire [14]. Cette augmentation de la capacité pulmonaire pourrait également expliquer l'augmentation significative de la durée moyenne des inspirations visible figure 2.4, avec une augmentation de la valeur moyenne de 120 ms le premier mois jusqu'à 170 ms le septième (+42%). Il est intéressant de constater que ces augmentations se stabilisent autour du 4e et du 5e mois.

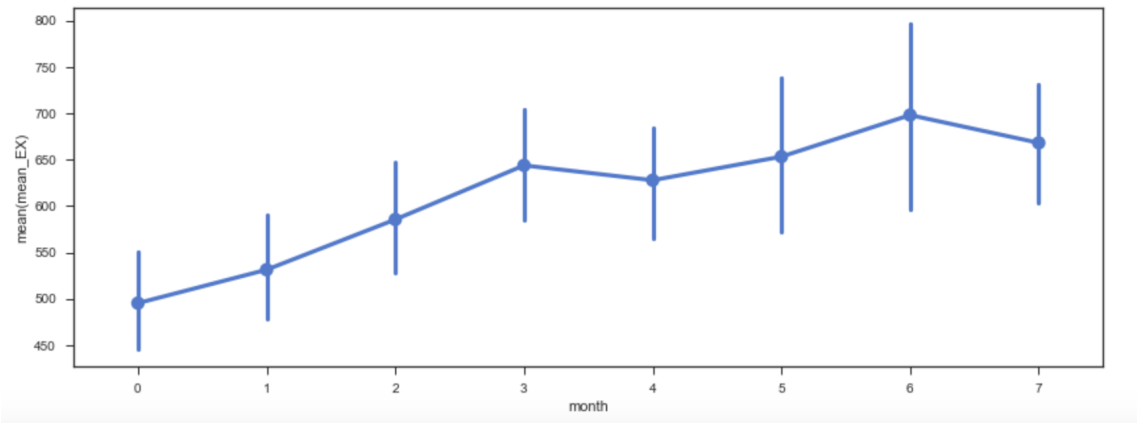


Figure 2.3 – Durée moyennes des expirations voisées (en ms), moyennée sur 31 bébés, pour différents âges (de 0 à 7 mois). Une différence statistiquement significative de la durée des expirations en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

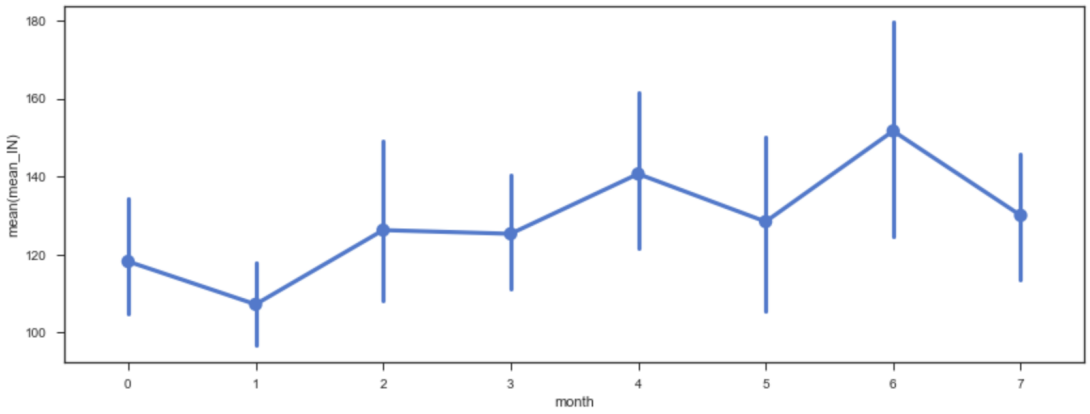


Figure 2.4 – Durée moyennes des inspirations voisées ou non (en ms), moyennée sur 31 bébés, pour différents âges (de 0 à 7 mois). Une différence statistiquement significative de la durée des expirations en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

On observe également une augmentation du rapport harmonie sur bruit (et donc inversement une diminution du rapport bruit sur harmonie, voir figure 2.7), dénotant une baisse de la rugosité et du caractère 'enrouée' de la voix du bébé [38]. On observe figure 2.5 que le HNR reste constant entre le 1er et le 3e mois, augmente fortement entre le 3e et le 4e mois et reste constant à partir du 4e mois. Or Scheiner [63] montre que le développement du contrôle nerveux sur la zone subglottale à partir du 3e mois se traduit par une augmentation de l'harmonicité de la voix. Inversement le jitter-loc-abs moyen, qui caractérise également une mesure de rugosité de la voix, est stable jusqu'au 3e mois, diminue fortement du 3e au 4e et est stable ensuite. Une diminution du jitter signifie un meilleur contrôle de la vibration des cordes vocales (inversement les voix de patients atteints de pathologies présente souvent un jitter plus élevé [65]). Un meilleur contrôle de la zone subglottale et du conduit vocal entraînerait donc une diminution de la rugosité de la voix de l'enfant à partir du 3e mois.

De plus la valeur moyenne du shimmer-apq5 est plus forte pour le contexte pee. Cependant si on observe la figure 2.8, on se rend compte que le contexte pee est fortement différent sur le 5e mois, mais que l'intervalle de confiance à 95 % dans ce cas-là est très important. Il est donc difficile de tirer des conclusions de ce résultat, possiblement dû à un outlier statistique.

L'analyse des résultats ANOVA (13 enfants, 4 mois, interpolé) met en avant le même genre de résultat. On retrouve en effet une augmentation de la durée moyenne d'expiration figure E.1 (en annexe), une augmentation du HNR entre le 3e et le 4e mois figure E.2, une diminution du NHR entre le 3e et le 4e mois figure E.3 et du shimmer-apq3 entre le 3e et le 4e mois. Une diminution du shimmer signifie, comme le jitter, un meilleur contrôle des cordes vocales accompagnant la maturation du système nerveux de l'enfant.

L'analyse de l'ANOVA permet également de mettre en avant, contrairement à l'analyse LME, une augmentation du max pitch, visible Figure 2.9. Cela pourrait exprimer que le bébé passe, dans les 4 premiers mois, d'un cri qui est purement une réaction physiologique à un cri qui est l'expression d'un état émotionnel. En effet Zeskind montre dans [72] que les cris avec un F0 maximum important ont été jugés plus aversifs sur un certain nombre d'échelles perceptuelles différentes. Le bébé serait donc plus capable d'exprimer son mal-être au fil des mois. Toutefois, le fait que ce résultat ne soit pas repris par les analyses LME peut faire penser qu'il résulte d'erreur d'interpolations de valeurs manquantes, ou qu'il ne persiste pas si on considère plus de bébés et plus de mois.

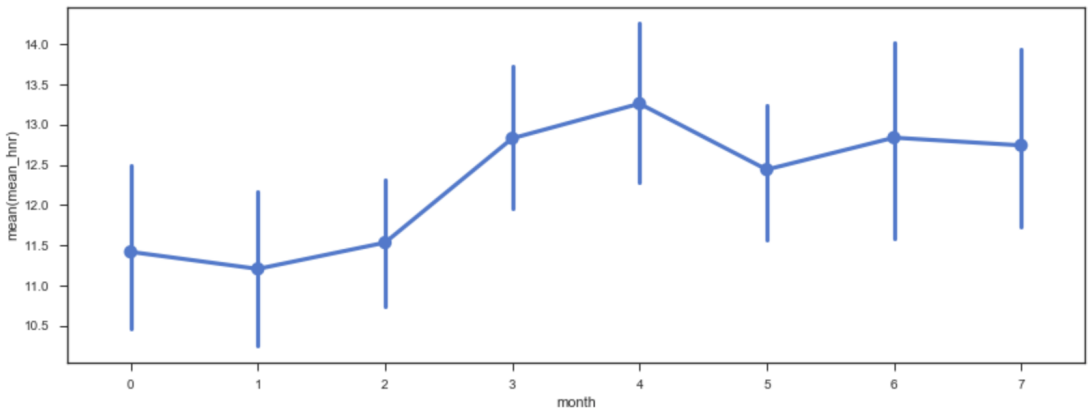


Figure 2.5 – Valeur moyenne du rapport harmonique sur bruit (en dB), moyennée sur 31 bébés, pour différents âges (de 0 à 7 mois). Une différence statistiquement significative de ce rapport en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

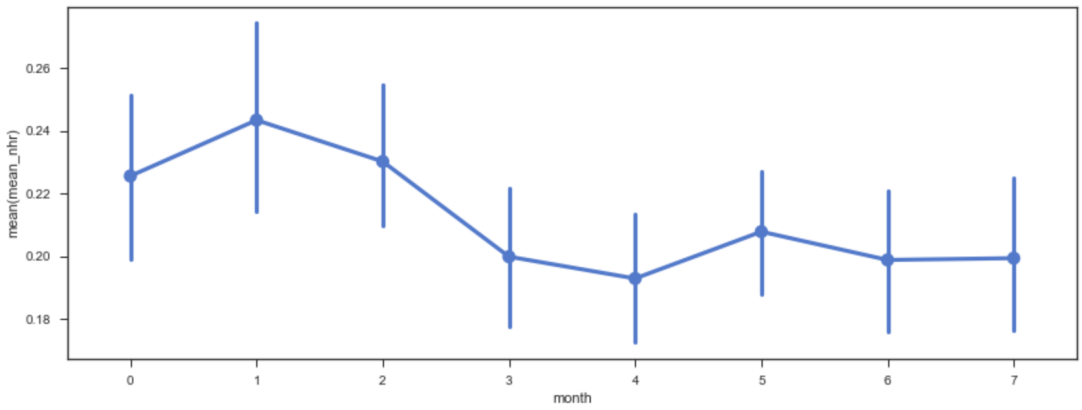


Figure 2.6 – Valeur moyenne du rapport bruit sur harmonique (en dB), moyennée sur 31 bébés, pour différents âges (de 0 à 7 mois). Une différence statistiquement significative de ce rapport en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

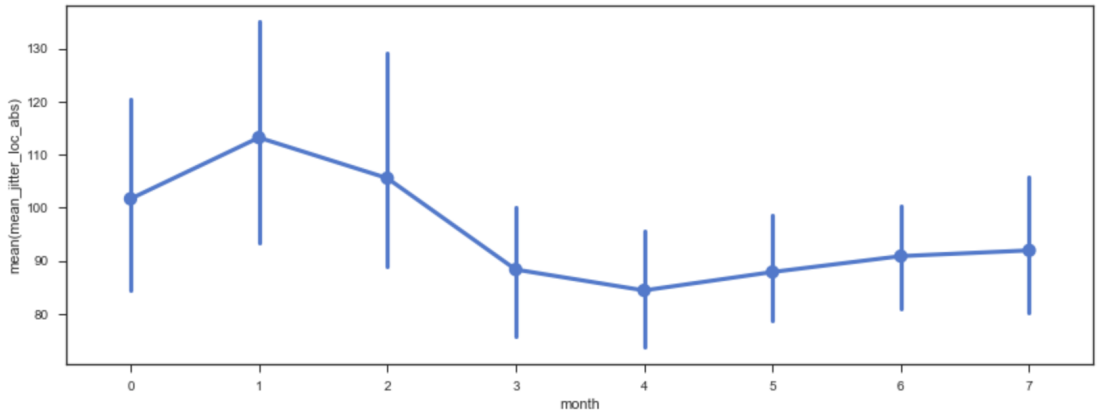


Figure 2.7 – Valeur moyenne du rapport bruit sur harmonique (en dB), moyennée sur 31 bébés, pour différents âges (de 0 à 7 mois). Une différence statistiquement significative du jitter-loc-abs (en secondes) en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

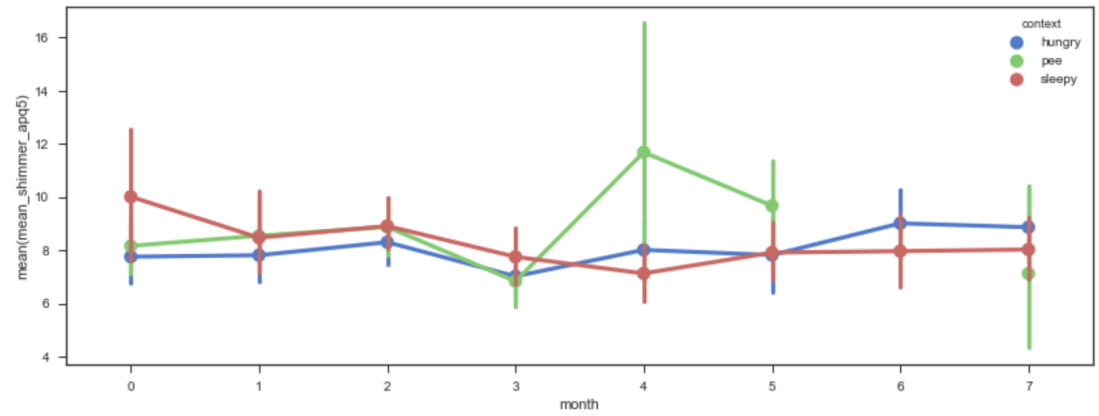


Figure 2.8 – Valeur moyenne du shimmer-apq5 (sans dimension), moyennée sur 31 bébés, pour différents contextes(hungry,pee,sleepy) et âges (de 0 à 7 mois). Une différence statistiquement significative du shimmer-apq5 en fonction du contexte est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

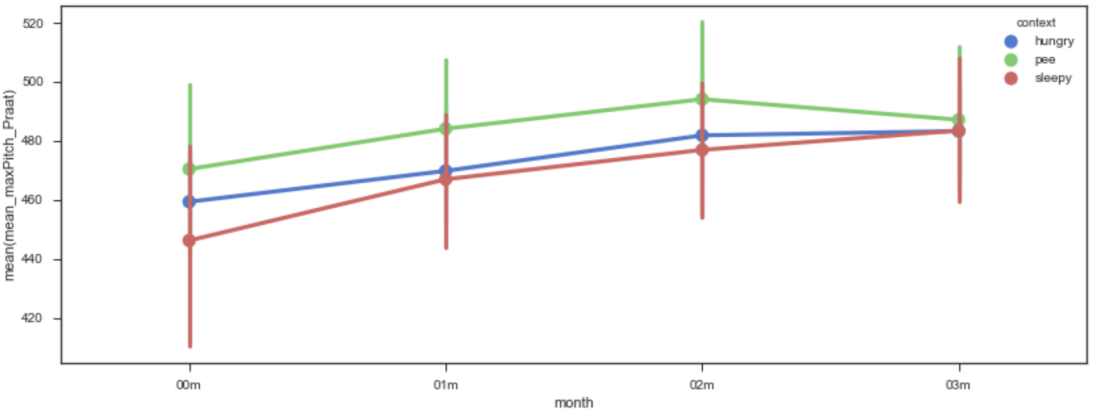


Figure 2.9 – Valeur moyenne du F0 maximum (en Hz), moyennée sur 13 bébés, pour différents âges (de 0 à 3 mois). Une différence statistiquement significative du F0 maximum en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

Chapitre 3

EXTRACTION DES CARACTÉRISTIQUES DYNAMIQUES DES CRIS, ET ÉTUDE STATISTIQUE DE LEUR VARIATION AVEC LE CONTEXTE

Cette troisième et dernière partie, s'intéresse à l'analyse d'un ensemble de contours de pitch prototypiques extraits du corpus. Nonaka fait, en effet, l'hypothèse dans [52] de la nature de marqueur linguistique des contours de hauteur observés dans les phases d'expiration. On présente ici un algorithme de clustering automatique des profils prosodiques des différents segments expirés du corpus permettant la découverte des profils les plus représentés du corpus. Ce genre d'algorithme pose plusieurs problèmes comme la nécessité de définir la notion de similarité entre séries temporelles mais également relatifs à la complexité algorithmique. On valide ensuite les résultats de l'algorithme perceptivement auprès d'un échantillon de mamans en utilisant une expérience de tri libre. Finalement, l'utilisation de ces profils prosodiques est analysée statistiquement en fonction de l'âge et du contexte du cri. On utilise pour cela un General Linear Mixed-Effects Models (GLME).

Après présentation du principe général de l'exploration des séries temporelles (section 3.1), on présentera l'algorithme de clustering automatique des profils prosodiques en section 3.2. Les différents contours de pitch prototypiques extraits seront ensuite analysés en section 3.3. Finalement, après validation perceptive des résultats (section 3.4), on étudiera la distribution des différents clusters trouvés en fonction de l'âge et du contexte.

3.1 Introduction à l'exploration des séries temporelles

Cette section propose une introduction aux différentes tâches relatives à l'exploration des séries temporelles. On peut noter que plusieurs de ces tâches sont similaires à celles que l'on retrouve dans l'apprentissage automatique. Nous décrivons brièvement les tâches qui seront utiles dans le cadre de l'extraction de contours de pitch prototypiques. Ainsi, la figure 3.1 décrit le worflow des différentes étapes utilisées par notre système générique d'extraction de savoir à partir des contours de pitch.

Les séries temporelles sont des données ordonnées dans le temps. Ainsi, on ne peut pas leur appliquer des méthodes de fouille de données classiques, mais bien des méthodes spécialement adaptées. Il est, en effet, nécessaire de respecter l'ordonnancement temporel de ce type de données.

Notion de similarité entre séries temporelles

Si on veut regrouper les séries en groupes similaires, il est tout d'abord nécessaire de définir la notion de similarité entre séries temporelles. Dans les bases de données traditionnelles, les mesures de similarité sont basées sur un appariement exact comme la distance euclidienne. Cependant, de nombreux chercheurs ont pointé les faiblesses d'une mesure de distance euclidienne dans le cas des séries temporelles [24, 39]. La plupart des tâches d'analyse des séries temporelles nécessitent, en effet, une notion plus subtile de similarité entre séries, basées sur la notion plus intuitive de

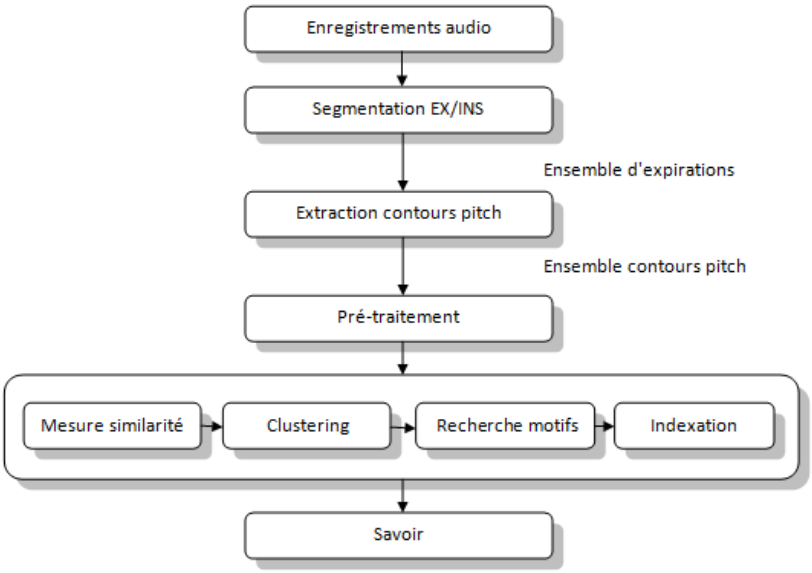


Figure 3.1 – Schéma général de l'extraction des contours de pitch prototypiques

forme. Nous entendons par forme d'une série les différentes variations relatives qu'elle effectue au cours du temps. Ce genre de mesure de similarité serait alors capable, comme la perception humaine, de s'abstraire de problèmes tels que des variations d'amplitude, de modification d'échelle, de distorsion temporelle, du bruit et les valeurs aberrantes. C'est-à-dire de reconnaître des objets similaires, même s'ils ne sont pas identiques mathématiquement.

On va donc chercher une mesure de similarité qui respecte les deux propriétés suivantes :

- Les valeurs absolues (les amplitudes) des points d'une série ne sont pas significatives. C'est la variation relative qui caractérise les séries temporelles
- Les variations des valeurs de deux séries peuvent intervenir à différents instants tout en préservant la similarité des séries

Pour respecter ces propriétés, on centrera (retrait de la valeur moyenne) les contours de pitch après leur transposition sur une échelle perceptive (voir section 3.2.1). Puis, on utilisera une mesure de similarité indépendante aux variations temporelles : le Dynamic Time Warping (voir section 3.2.2).

Notions de clustering

Le but est maintenant de regrouper les séries temporelles, à partir de la notion de similarité définie précédemment, dans différents groupes naturels de manière à ce que chaque groupe contiennent des séries similaires.

Le clustering est le processus qui permet de trouver ces groupes naturels, appelés clusters, dans un ensemble de données. Le clustering est une méthode de classification non supervisée dans le sens où, contrairement à la classification supervisée, on ne possède pas d'informations à priori sur l'appartenance à une classe des données. Il est cependant généralement nécessaire de définir le nombre de clusters voulu, donnée non connue dans la plupart des problèmes. Le choix du nombre de clusters, dans notre cas, est discuté section 3.2.4. L'objectif du clustering est alors de trouver un ensemble de clusters homogènes les plus différenciés possibles. Plus formellement, les groupes doivent typiquement maximiser la variance inter-clusters et minimiser la variance intra-clusters [16].

L'algorithme le plus utilisé est l'algorithme K-means [25], cependant, n'acceptant que des distances de type euclidienne, on utilisera plutôt un algorithme de type k-medoids. En effet, ce dernier accepte l'utilisation de n'importe quelle métrique et donc de la distance DTW.

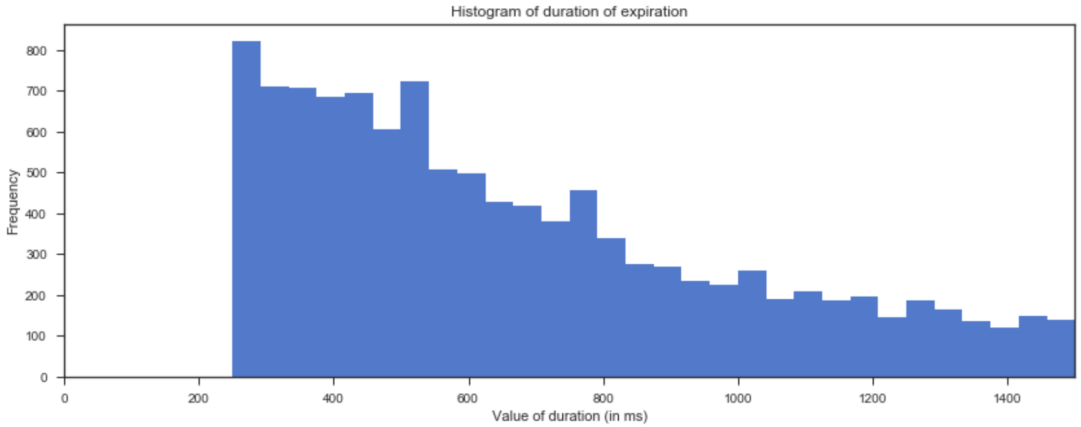


Figure 3.2 – Histogramme des durées des expirations

3.2 Explication de la méthode

3.2.1 Extraction des contours de pitch et pré-traitement

On a vu dans la section précédente que les deux algorithmes d'estimation de pitch donnent des résultats équivalents, on choisira donc d'utiliser l'algorithme SWIPE, au vu de sa plus grande facilité d'intégration en Python. On extrait pour chaque segment expiré estimé dans la section 1 un profil de pitch composé d'une estimation de la hauteur toutes les 10 ms. Pour chaque contour de pitch, on interpole linéairement les valeurs manquantes, c'est-à-dire les valeurs de pitch non estimées par SWIPE. La base de contours de pitch ainsi constituée est alors composée de plus de 60000 contours.

Une étape importante dans l'utilisation des séries temporelles obtenues à partir de mesures réelles est la réduction de bruit ainsi que la suppression des valeurs aberrantes effectuées par notre système d'extraction de connaissances (en anglais, *outliers*). Il est donc nécessaire de définir un certain nombre de critères sur l'ensemble des contours afin de séparer ce qui est du bruit de ce qui n'en est pas. De manière heuristique on décide de ne conserver, après écoute, que les contours de pitch de plus de 250 ms et de moins de 1500 ms. Les contours de moins 250 ms semblent, en effet, trop courts pour pouvoir donner lieu à un percept de forme du contour. Les contours de plus de 1500 ms sont, quant à eux, souvent des erreurs de segmentation et correspondent en réalité à un enchaînement d'expirations. On ne conserve également que les contours de pitch possédant moins de 10 % de valeurs interpolées. La base de donnée finale est constituée de plus de 25000 contours.

La figure 3.2 permet d'avoir une idée de la répartition des expirations en terme de durée. On constate que la majorité des expirations (80%) se durent moins d'une seconde.

Afin d'être cohérent avec la perception humaine, on transpose chaque contour de pitch en échelle logarithmique (cents). En effet, la perception des fréquences de l'oreille est logarithmique et non linéaire. Les écarts entre la fréquence 400 Hz et 800 Hz et la fréquence 800 Hz et 1600 Hz sont par exemple perçus de la même manière (un interval d'une octave). On définit alors la fréquence en cent, tel que :

$$\sigma = 1200 * \log_2 \frac{f}{f_{ref}} \tag{3.1}$$

avec ici σ fréquence en cent, f fréquence en Hz et f_{ref} une fréquence de référence en Hz que l'on fixe. On centre finalement, comme on s'intéresse plus aux variations relatives qu'aux valeurs absolues, chacun des contours de pitch de la base de données.

3.2.2 Dynamic time warping

Présentation générale de l'algorithme

La technique du Dynamic time warping (DTW) est une technique permettant de trouver un alignement optimal entre deux séquences données qui dépendent du temps, sous certaines restrictions. De manière intuitive, les séquences sont déformées de manière non-linéaire afin de les aligner l'une à l'autre, indépendamment de certaines transformations non linéaires du temps. Le formalisme complet du DTW et de ses contraintes est présenté par Muller dans [49], nous présentons ici une brève synthèse de la technique.

L'objectif de la DTW est de comparer deux séries temporelles, ici des contours de pitch, $\mathbf{X} = (x_1, x_2, ..., x_N)$ et $\mathbf{Y} = (y_1, y_2, ..., y_M)$ de taille respective N et M deux entiers.

Une séquence d'alignement est alors une séquence $\mathbf{P} = (p_1, p_2, ..., p_M)$ avec $p_l = (n_l, m_l) \in [1 : N] \times [1 : M]$ pour $l \in [1 : L]$ qui satisfait les trois conditions suivantes :

- Conditions aux frontières : $p_1 = (1, 1)$ et $p_L = (n, m)$
- Condition de monotonie : $n_1 \leq n_2 \leq ... \leq n_L$ et $m_1 \leq m_2 \leq ... \leq m_L$
- Condition du pas d'avancement : $p_{l+1} - p_l \in \big\{(1, 0), (0, 1), (1, 1)\big\}$ pour $l \in [1 : L - 1]$

On peut maintenant définir la matrice de coût cumulée $\mathbf{D}$, de taille $n \times m$, à partir des deux séquences $\mathbf{X}$ et $\mathbf{Y}$. On définit la distance cumulative $D(n, m)$ comme étant la distance locale $c(x_n, y_m)$, avec dans notre cas c distance euclidienne, ajoutée au minimum des distances cumulatives des éléments adjacents telle que :

$$D(n, m) = min\big\{D(n - 1, m - 1), D(n - 1, m), D(n, m - 1)\big\} + c(x_n, y_m) \tag{3.2}$$

avec ici $1 < n \leq N$ et $1 < m \leq M$.

La matrice D satisfait également les conditions aux frontières : $D(n, 1) = \sum_{k=1}^{n} c(x_k, y_1)$ pour $n \in [1 : N]$ et $D(1, m) = \sum_{k=1}^{n} c(x_1, y_k)$ pour $n \in [1 : M]$.

Finalement, on définit la distance DTW telle que $DTW(\mathbf{X}, \mathbf{Y}) = D(N, M)$. Cette distance est implémentée de manière dynamique (i.e. en utilisant les distances cumulatives calculées précédemment afin de réduire grandement la complexité de l'algorithme) en $O(NM)$ opérations. Elle nous permet de comparer deux séquences temporelles indépendamment des déformations temporelles. L'algorithme permettant de calculer le chemin d'alignement optimal est disponible annexe A. Un exemple d'alignement de deux contours de pitch extraits d'une phase crié (expiration) de bébé est visible en Figures 3.3 et 3.4.

Rajout de contraintes

Une des premières contraintes courantes est de rajouter des poids multiplicatifs additionnels (w_d, w_h, w_v), tels que :

$$D(n, m) = min \begin{cases} D(n - 1, m - 1) + w_d c(x_n, y_m) \\ D(n - 1, m) + w_h c(x_n, y_m) \\ D(n, m - 1) + w_v c(x_n, y_m) \end{cases}$$

Si on pose $(w_d, w_h, w_v) = (1, 1, 1)$, on se retrouve dans le cas classique. Ce dernier à une préférence pour la direction diagonale, car un pas diagonal (coût d'une cellule) correspond à la combinaison d'un pas horizontal et vertical (coût de deux cellules). Afin de contrebalancer cette préférence, il est souvent choisi $(w_d, w_h, w_v) = (2, 1, 1)$.

Une autre contrainte courante est d'imposer une condition sur les séquences d'alignement admissibles. Une telle contrainte permet non seulement de diminuer la complexité temporelle de l'algorithme, mais également de se prévenir d'alignements pathologiques en contraignant le chemin de la séquence d'alignement. Une contrainte connue est la bande de Sakoe-Chiba. La séquence d'alignement peut alors seulement parcourir une région donnée dans la matrice de coûts cumulés.

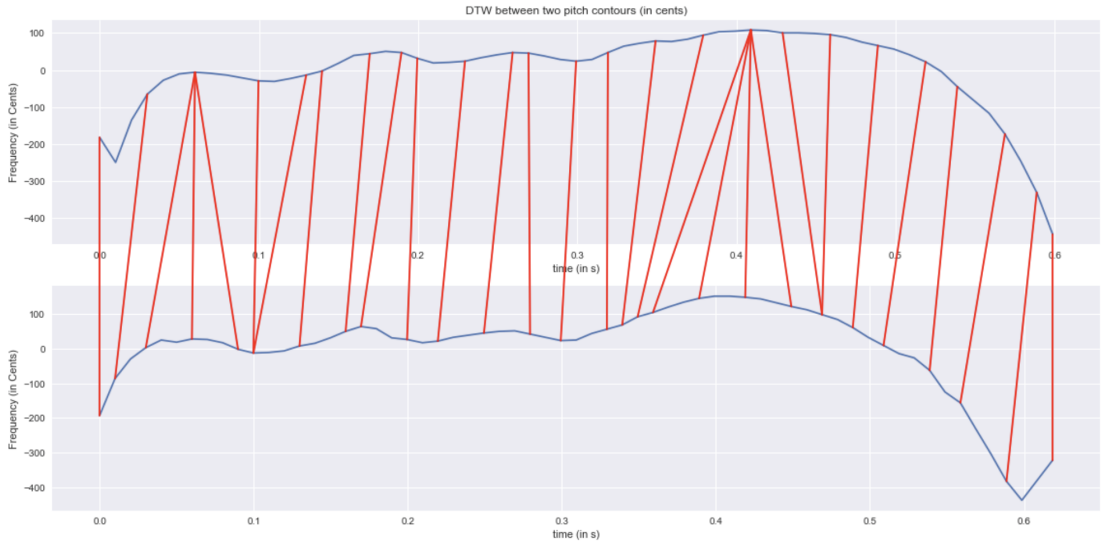


Figure 3.3 – Alignement temporel de deux contours de pitch **X** (en haut) et **Y** (en bas). Les points alignés sont indiqués par les traits rouges.

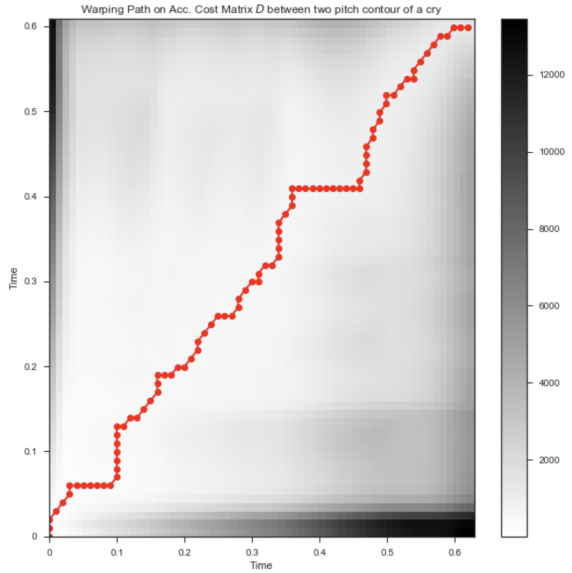


Figure 3.4 – Matrice de coûts cumulés entre deux contours de pitch **X** (axe vertical) et **Y** (axe horizontal) en utilisant la distance euclidienne comme distance locale. Les régions de faibles coûts cumulés sont indiquées en couleur claire et les régions de forts coûts cumulés sont indiquées en couleur sombre. L'alignement optimal choisi est indiqué par des points rouges

La bande de Sakoe-Chiba longe la diagonale principale de la matrice à une largeur fixe (horizontale et verticale) $T \in \mathbb{N}$. Cette contrainte implique qu'un élément x_n peut seulement être aligné à un élément y_m tel que $m \in \left[\frac{M-T}{N-T}(N-T), \frac{M-T}{N-T}(N+T)\right] \cap \left[1 : M\right]$.

Pour définir la largeur de la bande, on utilise souvent le facteur br, définit tel que :

$$T = 2 * \lfloor (br * min(N, M)) \tag{3.3}$$

De manière heuristique, on choisit dans la suite de ce travail les valeurs des poids multiplicatifs $(w_d, w_h, w_v) = (2, 1, 1)$ et $br = 0.1$.

3.2.3 Algorithmes de clustering

Les différents algorithmes présentés ici sont abordés de manière plus profonde dans [51].

K-medoid

L'algorithme k-medoid est un algorithme de clustering conceptuellement similaire à l'algorithme k-means [25], mais utilisable dans un espace métrique non nécessairement euclidien. Ces deux algorithmes partitionnent les données en groupes, appelés clusters, et essayent de minimiser la distance entre les points appartenant à chacun des clusters et un point désigné comme le centre de ce cluster. Contrairement à l'algorithme k-means, k-medoid utilise des données comme centre (les medoids). Le medoid d'un cluster est défini comme l'objet du cluster dont la distance moyenne à tous les autres objets du cluster est minimal, c'est-à-dire comme l'objet le plus représentatif du cluster (dans notre cas, le contour de pitch le plus représentatif du cluster de contours). Contrairement a k-means, l'algorithme k-medoid peut utiliser n'importe quelle distance et donc la distance DTW.

De manière plus formelle les algorithmes k-medoid organisent N objets dans K partitions (avec $K \leq N$) qui cherchent à minimiser W_K, distance intra-cluster totale, définie telle que :

$$W_K = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{kn} DTW(x_n - m_k) \tag{3.4}$$

avec x_n, où $n \in [1, N]$, point de la base d'entraînement, N taille de la base d'entraînement, DTW la distance DTW, z_{kn} coefficient valant 1 dans le cas où x_n appartient au cluster numéro k et 0 sinon et m_k medoid numéro k du partitionnement à K cluster.

PAM

Un des algorithmes de type k-medoid les plus communs est l'algorithme Partitioning Around Medoids (PAM). Un schéma de l'algorithme est proposé figure 3.5.

L'idée de l'algorithme est la suivante :

- Sélection de k medoid aléatoirement sur les n données
- Association de chacune des données à son plus proche medoid
  - Tant que la distance intra-cluster diminue
    - Pour chaque medoid m, pour chaque donnée non medoid o :
      - Échanger m et o, puis recalculer la distance intra-cluster du nouveau medoid
      - Si la distance intra-cluster augmente dans l'étape précédente, annuler l'échange

L'algorithme complet est présenté en annexe F. La complexité temporelle est $O(k(n-k)^2)$ avec ici : k nombre de clusters, n taille des données. Cette complexité le rend en pratique inutilisable pour le partitionnement de base de données de plus de 1000 objets.

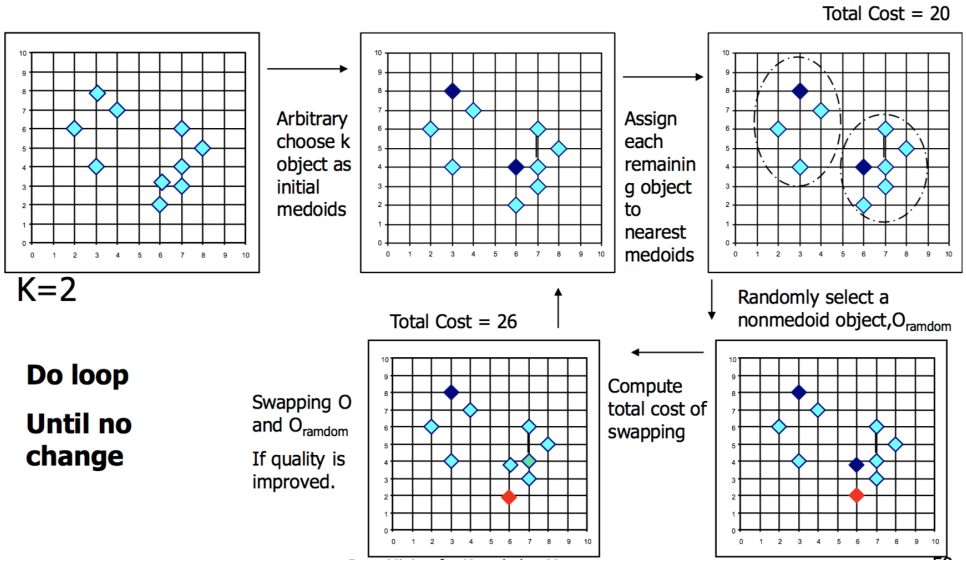


Figure 3.5 – Algorithme Partioning around Medoid (PAM)

CLARA

L'algorithme Clustering for LARge Application (CLARA) est un dérivé de l'algorithme PAM qui réduit considérablement la complexité temporelle de ce dernier. Un schéma de l'algorithme est proposé figure 3.6.

Le principe est le suivant :

- On tire aléatoirement r échantillons de taille s
- Pour chaque échantillon on applique l'algorithme PAM et on extrait k medoids optimaux
- Les k medoids optimaux que l'on conserve sont ceux qui donnent une distance intra-cluster minimale par rapport à l'ensemble des points de la base de données.

L'algorithme complet est donné en annexe 3.6. La complexité temporelle est $O(ks^2 + k(n-k))$, avec ici : s taille des échantillons, k nombre de clusters, n taille des données. La complexité est donc linéaire par rapport à la base de données, ce qui est bien mieux que PAM qui est quadratique. Cependant, dans le cas où, les meilleurs k medoids ne sont pas sélectionnés durant la phase d'échantillonnage alors l'algorithme CLARA ne sera pas capable de trouver le meilleur clustering. Cet algorithme permet donc de gagner en complexité temporelle, mais en prenant le risque de perdre en qualité de clustering.

CLARANS

L'algorithme Clustering Large Applications based upon RANdomized Search (CLARANS) a été proposé pour améliorer la qualité de CLARA. Pour comprendre le fonctionnement de l'algorithme, il est nécessaire d'introduire quelques notions de recherche de solution dans un graphe. Le processus de clustering peut, en effet, être présenté comme la recherche de solution optimale dans un graphe (présenté 3.7) où chaque nœud est une solution potentielle, c'est-à-dire un ensemble de k medoids.

De manière formelle, un nœud du graphe note $G_{n,k}$ est représenté par un ensemble d'objets $\{O_{m1}, ..., O_{mk}\}$ avec k nombre de clusters et $O_{m1}, ..., O_{mk} \in D$, D représentant l'ensemble des objets. L'ensemble des nœuds du graphe est donc tel que : $\{\{O_{m1}, ..., O_{mk}\} | O_{m1}, ..., O_{mk} \in D\}$. Si deux nœuds S_1 et S_2 sont voisins alors $|S_1 \cap S_2| = k - 1$.

L'algorithme PAM cherche le nœud qui minimise la distance intra-cluster. A chaque itération i de l'algorithme, il analyse tous les voisins du nœud actuel (la meilleure solution à l'itération i) et l'échange avec le nœud qui possède la plus forte diminution de la distance intra-cluster. Cette méthode est très peu efficace d'un point de vue computationnel.

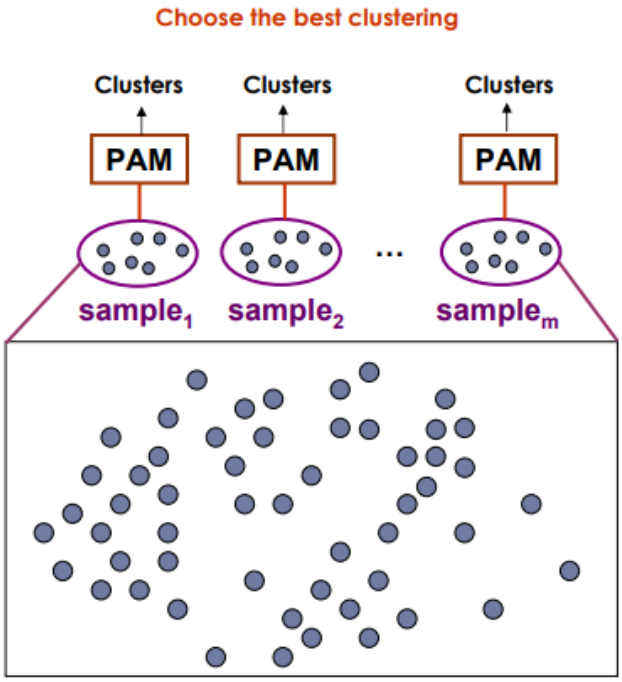


Figure 3.6 – Algorithme CLARA

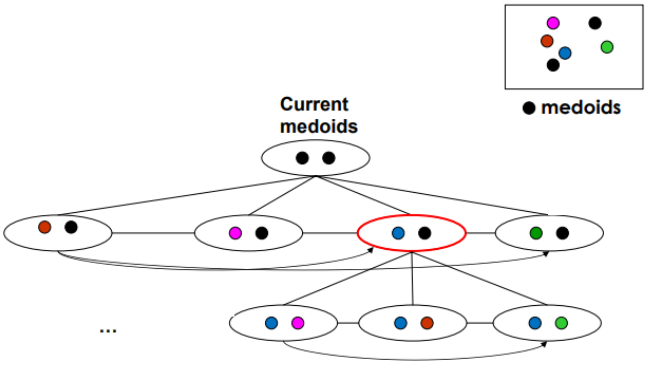


Figure 3.7 – Graphe des solutions potentielles du clustering

L'algorithme CLARA, quant à lui, diminue le nombre de nœuds en restreignant sa recherche à une sous-partie du graphe. Cette stratégie condamne la solution optimale à n'être jamais trouvée si le meilleur nœud ne fait pas partie du sous-graphe. Comme CLARA, CLARANS ne regarde pas tous les voisins du nœud actuel, mais ne se limite pas à un sous-graphe. En fait, il cherche dans le graphe original, mais regarde un échantillon du voisinage du nœud. En d'autres termes, alors que CLARA tire un échantillon de nœuds au début de chaque itération, CLARANS tire un échantillon de voisinage au début de chaque itération. La recherche n'est donc pas localisée dans un sous-graphe.

L'idée de l'algorithme est la suivante :

- On tire aléatoirement un nœud du graphe.
- Tant que la distance intra-cluster diminue
  - On tire un échantillon du voisinage de ce nœud. La taille de l'échantillon est spécifiée par l'utilisateur.
  - Si on trouve un meilleur nœud, on le définit comme le nouveau nœud actuel
- Un optimum local est alors trouvé, l'algorithme recommence alors avec un nouveau nœud aléatoire à la recherche d'un autre optimum.
- L'optimum local que l'on conserve est celui qui possède la distance intra-cluster la plus faible.

Ng montre dans [51] que l'algorithme CLARANS est plus efficace que l'algorithme PAM et l'algorithme CLARA. La complexité temporelle de CLARANS est $O(n^2)$, où n est la taille des objets. Même si l'algorithme CLARANS est quadratique, il reste cependant d'une complexité inférieure à l'algorithme PAM.

Algorithme final

On se trouve ici dans le cas d'une base de données de plus de 25000 contours de pitch. L'algorithme CLARANS est bien adapté pour les bases de quelques milliers d'objets, mais, étant de complexité quadratique, est malheureusement inutilisable pour une base de données de cette taille. On utilisera donc l'algorithme CLARA qui lui possède une complexité linéaire. Ng dans [51] montre que CLARANS à des performances similaires à PAM pour des temps de calcul moins importants. Afin de réduire le temps de calcul nécessaire au clustering de cette base de données, on utilisera CLARANS à la place de PAM dans l'algorithme CLARA. L'algorithme final utilisé est présenté figure 3.8.

On utilisera pour les algorithmes les paramètres donnés par Ng dans [51] et décrits comme donnant de bons résultats. C'est à dire :

- Pour l'algorithme CLARA : $r = 5$ nombre d'échantillons tirés aléatoirement de taille $s = 0.04 * n$ avec n taille de la base de données
- Pour l'algorithme CLARANS : $numlocal = 2$ optimums locaux et $maxneighbor = 0.0125 * k(n-k)$ taille de l'échantillon du voisinage du nœud considéré avec n taille de la base de données et k nombre de clusters

L'algorithme CLARANS étant implémenté par A.Novikov dans sa bibliotheque `pyclustering` [1], les contributions du stage sont donc au nombre de deux :

- Rajout de la métrique DTW pour l'algorithme CLARANS
- Implémentation d'une version modifiée de l'algorithme CLARA utilisant l'algorithme CLARANS au lieu de l'algorithme PAM.

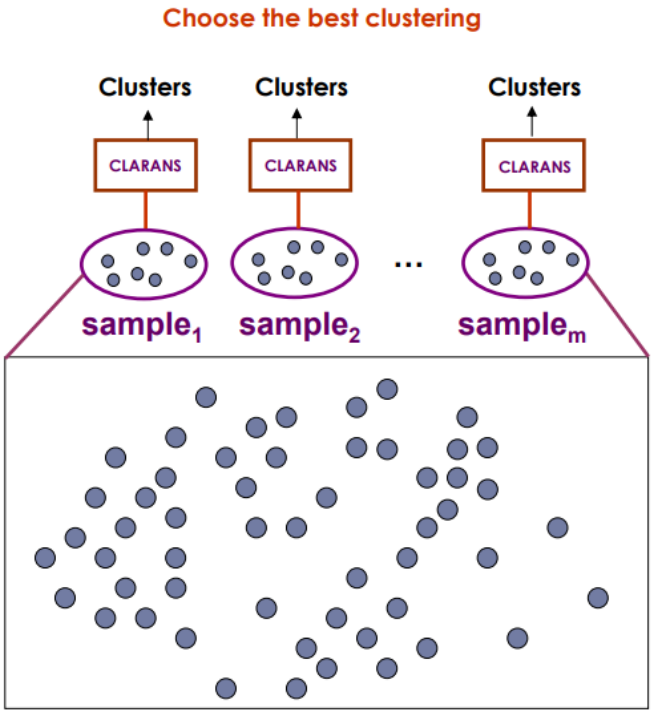


Figure 3.8 – Algorithme final

3.2.4 Choix du nombre de clusters

Une des méthodes [40] pour choisir K, nombre de clusters est d'utiliser la distance intra cluster totale. On visualise pour cela cette distance en fonction de K. La recherche d'un "coude" dans ce graphe est une indication heuristique du choix de K donné (c'est la technique du elbow point). De manière plus formelle, on cherche un K^* tel que :

$$\{W_K - W_{K+1} : K < K^*\} >> \{W_K - W_{K+1} : K^* < K\} \tag{3.5}$$

Malheureusement, la figure 3.9 ne présente pas de coude visible. On décide donc de choisir le nombre de clusters de manière heuristique. On observe alors, en faisant varier K de 2 à 8, la compacité des clusters trouvés, la taille des clusters (pourcentage des cris à l'intérieur) et la forme des medoids trouvés. Finalement, on fixe $K = 8$.

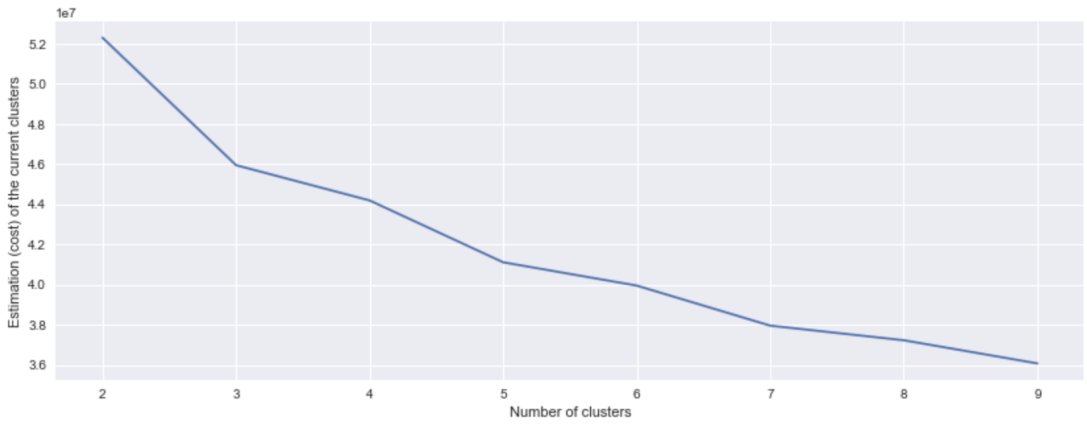


Figure 3.9 – Coût total du clustering en fonction du nombre de cluster

3.3 Résultats de clustering

Les résultats de clustering sont présentés en Figure 3.10. A l'inspection, le premier cluster apparait peu pertinent. En effet, sa taille relative et sa compacité sont faibles. De plus, le medoid extrait variant fortement par rapport aux autres medoids, il est difficile d'en extraire une tendance. On ne considérera donc pas ce cluster dans la suite du rapport, mais seulement les 7 derniers clusters.

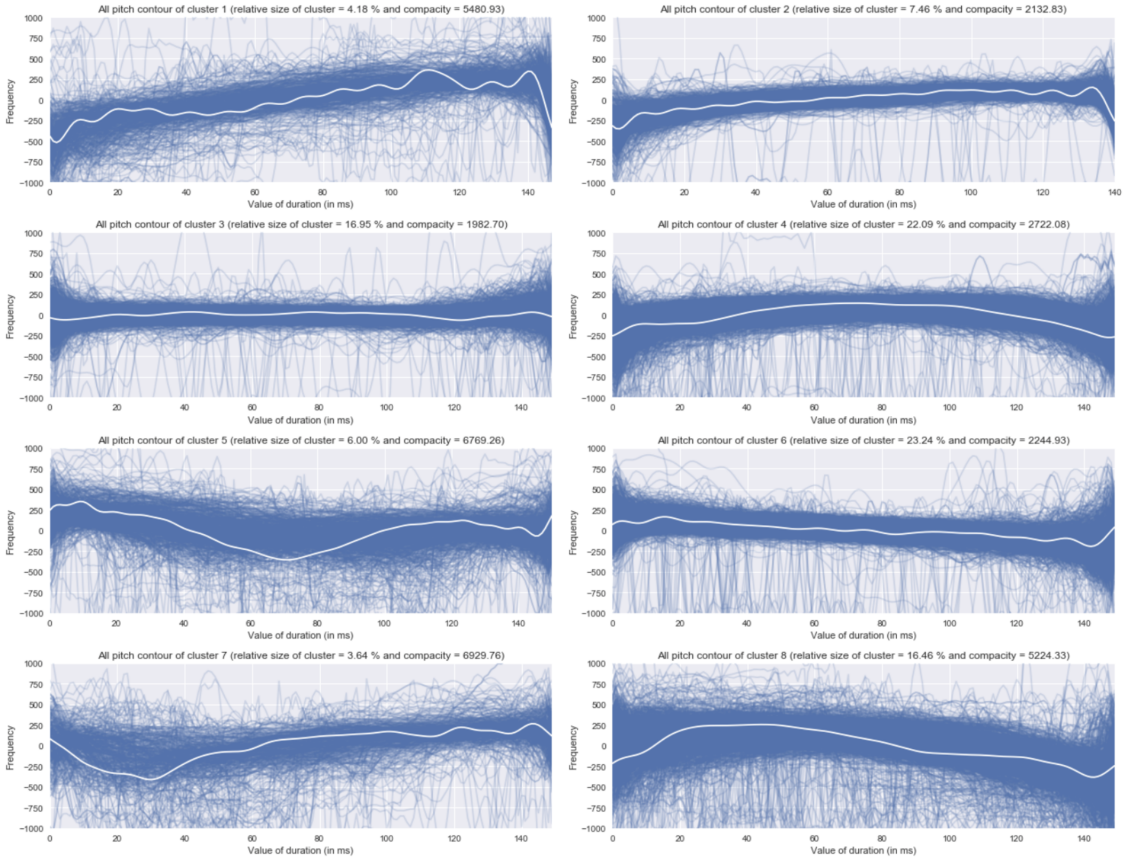


Figure 3.10 – Résultats du clustering d'une base de plus de 25 000 contours de pitch. Tous les contours de pitch sont normalisés temporellement pour la visualisation des clusters. Pour chaque cluster les données non medoid sont tracées en bleu et le medoid est tracé en blanc

En analysant les différents contours prototypiques extraits, on trouve 7 modes de cri schématisés dans la figure 3.11, que l'on peut décrire de la manière suivante (note que l'on conserve la numérotation initiale des 8 clusters, en excluant le cluster 1: les 7 modes sont donc numérotés de 2 à 8).

- Le mode 2 est un contour de pitch croissant
- Le mode 3 est un contour de pitch stable
- Le mode 4 est un contour en U inversé
- Le mode 5 est un contour en U
- Le mode 6 est un contour décroissant
- Le mode 7 est un contour fortement décroissant puis croissant
- Le mode 8 est un contour fortement croissant puis décroissant

Il est intéressant de constater que l'on trouve des formes comparables à celles trouvées dans la littérature. En effet, on peut trouver un lien avec les profils identifiés de façon manuelle par Chittora dans [22] et visibles en figure 0.2, et ceux postulés par Wermke dans [70] et visibles en figure 0.3.

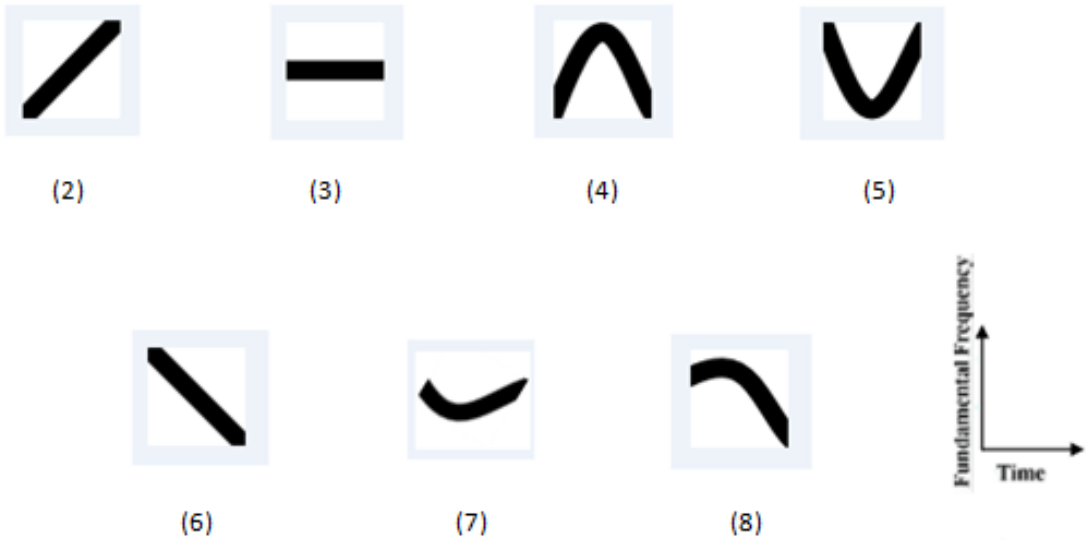


Figure 3.11 – 7 modes de cris proposés après analyse des contours de pitch prototypiques extraits par clustering. Le numéro du mode correspond au numéro du cluster dont est extrait le contour de pitch prototypique

On peut faire plusieurs rapprochements :

- Le mode 2 est très similaire au (b) (f0 croissant) de Chittora et au type 3 de Wermke (rising)
- Le mode 3 au type 4 (plateau) de Wermke et au (a) de Chittora (f0 plat)
- Le mode 4 au type 2 (symetric) de Wermke
- Le mode 6 au (c) de Chittora et au type 1 de Wermke (falling)
- Le mode 8 au type 3 de Wermke (rising)

On constate que si l'on conserve les 4 meilleurs clusters (en terme de compacité et de taille relative), c'est-à-dire les clusters (2,3,4,6), on retombe sur les 4 modes de cris postulés par Wermke (plus de 70 % des cris sur ces 4 clusters). Mampe dans [45] n'avait choisi que de considérer les cris en U inversé, prépondérants selon lui. Or, ici, les cris en U inversé sont les deuxièmes plus courants (avec 22.09 %) à quasi-égalité avec les cris décroissants (avec 23.24 %).

3.4 Validation perceptive des clusters trouvés

3.4.1 Expérience de catégorisation libre d'un ensemble de contours de pitch

Présentation de l'expérience

Cette section présente une expérience de tri libre réalisée sur un échantillon de N=11 mamans (en Juillet 2017) avec le logiciel TCL-LabX afin de valider perceptivement les résultats de l'algorithme obtenus dans le paragraphe précédent. La procédure demandée aux participant(e)s de l'expérience est de trier librement, et sans contrainte de temps, 35 courts cris de bébés d'environ 500 ms (5 pour chacun des 7 clusters défini précédemment) dans un nombre arbitraire de clusters selon leur similarité perceptive. Le même ensemble de stimuli était présenté à chacune des mères, mais, dans un ordre aléatoire. L'expérience était constituée de deux étapes.

Dans la première étape (d'une durée de 20 minutes environ), on demande aux participantes de classer les stimuli par similarité, en une série de plusieurs catégories ou classes (une catégorie pouvant ne contenir qu'un seul stimulus). Le critère de tri est de "regrouper dans un même tas les cris présentant la même forme au cours du temps". Les participantes peuvent écouter chaque son autant de fois souhaité et n'était pas limitées dans le temps. Les instructions de la première étape sont détaillées en annexe I.

Dans la deuxième étape (d'une durée de 10 minutes environ), il est demandé de décrire, pour chacune des classes crées dans l'étape précédente, ce qui caractérise, selon le participant, les cris de cette classe, et en particulier ce que le bébé pourrait exprimer avec ce genre de cris. Les instructions de la deuxième étape sont détaillées en annexe J. Les données de la deuxième étape n'ont malheureusement pas été traitées dans le cadre de ce stage et feront l'objet d'une étude ultérieure.

Les différents tris individuels ont ensuite été utilisés pour créer une matrice de co-occurrence globale qui a été soumise à un algorithme de clustering hiérarchique à partir duquel on a pu extraire sept catégories. Cela nous permet d'évaluer si les clusters générés se répartissent de manière cohérente avec ces catégories. Les différents outils nécessaires à la mise en place de l'expérience et à l'analyse de ces résultats sont brièvement présentés dans les paragraphes suivants.

Méthode du tri libre

Le principe de la méthode du tri libre [30], est de demander aux sujets d'effectuer la tâche de catégorisation sans information sur les catégories attendues a priori, ni leur nombre, et de ne demander la description des catégories qu'une fois l'ensemble des partitions réalisées.

L'expérimentateur se trouve généralement confronté à une grande quantité de paramètres susceptibles de se révéler pertinents pour l'auditeur dans l'écoute des sons. En général, on choisit de décider les paramètres acoustiques particuliers à soumettre à la perception du participant pour évaluer s'ils sont susceptibles de faire partie de ceux que l'auditeur utilise et traite lors de l'écoute du son. On choisira alors dans notre cas à se limiter à des sons d'intensité et de durée équivalentes.

Les tests de catégorisation libre permettent d'appréhender, non seulement ce que l'auditeur est capable de percevoir, mais également ce qu'il va chercher à identifier dans le signal acoustique qui lui permette de comprendre ce qu'il entend.

Présentation du logiciel TCL - LabX

TCL-LabX [6, 30] est un progiciel spécifiquement développé afin de préparer rapidement et facilement des expérimentations de tri libre et de recueillir les données finales sous une forme susceptible d'être traitée par des logiciels de statistiques. Le nombre de catégories n'est pas limité par la consigne, et chaque participant est libre à la fois du nombre et du contenu de chacune d'elles. Les stimuli sont présentés sous forme de boutons carrés sur un écran d'ordinateur. Le participant peut alors librement écouter les sons et/ou déplacer les boutons à l'écran à l'aide de la souris. L'interface de TCL-LabX est présentée figure 3.12.

Clustering hiérarchique

Le clustering hiérarchique [62] est une méthode de clustering non supervisée qui utilise une matrice de distance entre les points de la base de données comme critère de clustering. Cette méthode ne requiert pas de préciser le nombre de clusters k.

Figure 3.12 – Interface de TCL-LabX, chaque carré correspond à un court cri de 500 ms

On utilisera, dans notre cas, un algorithme de type ascendant, dont l'idée est la suivante :

- Calcul de la distance de matrice entre l'ensemble des points de la base de données
- Chaque point est considéré comme un cluster
- Tant que l'on n'a pas un seul cluster restant :
  - Fusionner les deux clusters les plus proches
  - Mettre à jour la matrice de distance

On remarque que cet algorithme nécessite de définir comment calculer une distance entre deux clusters, différentes définitions de la distance entre les clusters pouvant alors produire des résultats très différents. On peut visualiser les résultats de l'algorithme sous la forme d'un dendrogramme : un diagramme de la forme d'un arbre qui permet d'illustrer l'arrangement de groupes crées par l'algorithme. On peut alors scinder le dendrogramme, à n'importe quel endroit, pour avoir le nombre de catégories désirées. L'algorithme de clustering hiérarchique est disponible dans la librairie `scipy` [37].

3.4.2 Résultats de l'expérience

Présentation des résultats

À partir des tris individuels de chaque participante, on construit une matrice de co-occurrence globale de taille $(35, 35)$ (35 étant le nombre de sons utilisés dans l'expérience). Chaque coordonnée (i, j) de la matrice correspond au nombre de fois ou une participante a mis le son i dans le même cluster que le son j. La matrice de co-occurrence est présentée en Figure 3.13. Après transformation de cette dernière en matrice de distance, on peut la soumettre à un clustering hiérarchique et visualiser les différents dendrogrammes obtenus en faisant varier la métrique de l'algorithme. Après visualisation de ces derniers, on choisit de manière heuristique la distance dite de Ward. La distance de Ward entre deux cluster C_i et C_j est définie telle que :

$$D_w(C_i, C_j) = \sum_{x \in C_i} (x - r_i)^2 + \sum_{x \in C_j} (x - r_j)^2 + \sum_{x \in C_{ij}} (x - r_{ij})^2 \tag{3.6}$$

avec r_i centroid de C_i, r_j centroid de C_j et r_{ij} centroid du cluster obtenu après fusion des deux précédents clusters : C_{ij}. On construit finalement un dendrogramme que l'on peut observer figure 3.14.

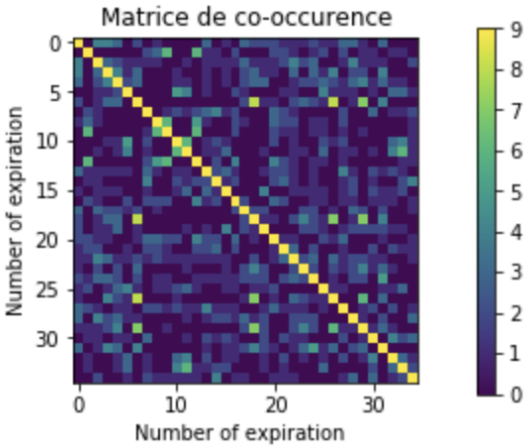


Figure 3.13 – Matrice de co-occurrence obtenue après expérience de tri libre sur 35 enfants

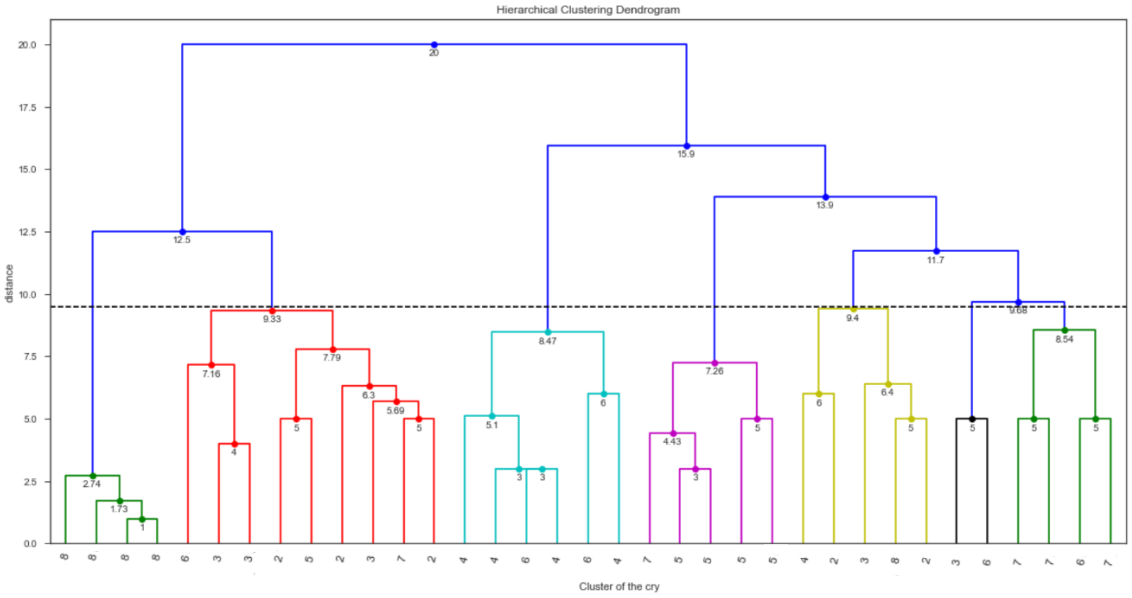


Figure 3.14 – Dendrogramme obtenu après clustering hiérarchique sur la matrice de co-occurrence construite avec les résultats de l'expérience de tri libre

Analyse des résultats

Si on coupe le dendrogramme pour avoir sept catégories, on a :

- Catégorie 1 : 8 - 8 - 8 - 8
- Catégorie 2 : 6 - 3 - 3 - 2 - 5 - 2 - 3 - 7 - 2
- Catégorie 3 : 4 - 4 - 6 - 4 - 6 - 4
- Catégorie 4 : 7 - 5 - 5 - 5 - 5
- Catégorie 5 : 4 - 2 - 3 - 8 - 2
- Catégorie 6 : 3 - 6
- Catégorie 7 : 7 - 7 - 6 - 7

Par rapport aux 4 meilleurs clusters définis dans la section précédente, on constate que le cluster 2 et le clsuter 3 semblent perceptivement relativement interchangeables (regroupés ici dans la catégorie 2 et le catégorie 5) et que le cluster 4 semble particulièrement bien défini perceptivement (identifié par les participants comme la catégorie 3). Cependant le cluster 6 qui est un bon cluster en terme de taille et de compacité semble mal défini perceptivement (égrené ici dans les catégories 2, 3, 6 et 7). De manière inverse le mode 8 qui est très peu compacte semble bien défini perceptivement.

Test de l'association statistique des catégories et des clusters trouvés par l'algorithme

On teste ici si l'association statistique des catégories trouvés dans l'expérience du tri libre et des clusters trouvés par l'algorithme de clustering est significative. On considère alors deux variables **X** et **Y**. L'hypothèse nulle H_0 de ce test est la suivante : les deux variables **X** et **Y** sont indépendantes. **X** et **Y** sont censées prendre un nombre fini de valeurs, I pour **X**, J pour **Y**. On dispose d'un échantillonnage de taille N données. Dans notre cas, X_i correspond au cluster numéro i et Y_j, à la catégorie numéro j et $N = 35$ au nombre de cris présentés lors de l'expérience de tris libres. On effectuera ici deux tests statistiques : le test du χ^2 et le test exact de Fisher.

Test du χ^2 On fait d'abord un test du χ^2 [32]. Notons Oi, j l'effectif observé de données pour lesquelles X prend la valeur i et Y la valeur j. On appelle le tableau composé des éléments Oi, j tableau de contingence. Dans notre cas l'élément Oi, j représente alors le nombre de cris extraits du cluster trouvé par l'algorithme i dans la catégorie j. La valeur attendue dans chaque cellule Ei, j, sous l'hypothèse nulle, est simplement le produit du totale de la ligne i par le total de la colonne j divisé par N. On calcule alors la distance entre les valeurs observées et les valeurs attendues s'il y avait indépendance avec la formule :

$$T = \sum_{i,j} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}} \tag{3.7}$$

Il est possible de montrer que T suit une loi du χ^2 à $(I-1)(J-1)$ degrés de liberté. Dans notre cas le test du χ^2 est normalement inutilisable, car non-valide dans le cas où la majorité de la valeur attendue dans chaque cellule est inférieure à 5. Il est ici donné à but comparatif : la valeur du χ^2 est très élevée, et même si la p-valeur n'est pas fiable, cela indique que l'association est significativement différente de ce que l'on pourrait attendre par chance.

$$\chi^2(36) = 91, p = 1.2649e^{-6} \tag{3.8}$$

Test exact de Fisher On utilise ici un test exact de Fisher [28] qui est, contrairement au test du χ^2, valable pour toutes les tables de contingence. Ce test se base sur le fait que si les variables $\mathbf{X}$ et $\mathbf{Y}$ sont indépendantes, la probabilité de présenter X_i et Y_j est alors égale à $P(X_i) * P(X_j)$. On peut ainsi calculer la probabilité de se trouver dans chaque case du tableau. On peut donc calculer la probabilité, si l'hypothèse nulle est vraie, d'observer un tableau de contingence donné.

On calcule, pour un certain nombre de tableaux de contingence simulés par la méthode de Monte Carlo (qui crée pour chaque tableau $N = 35$ observations aléatoires), dont le tableau étudié, la possibilité d'observer ce tableau de contingence si l'hypothèse nulle est vraie. On range alors les tableaux de contingence en deux catégories : ceux qui sont plus compatibles avec l'hypothèse nulle que le tableau étudié (leur probabilité est plus élevée sous l'hypothèse nulle), et ceux qui sont autant ou moins compatibles. On calcule alors la p-valeur tel que :

$$p_{mc} = \frac{NLE + 1}{N_{runs} + 1} \tag{3.9}$$

avec ici NLE le nombre de tableaux de contingence autant ou moins compatibles avec l'hypothèse nulle que le tableau étudie et N_{runs} le nombre total de simulations.

La p-valeur estimée pour 10 000 simulations est, dans notre cas, trouvée égale à 0.00009999, soit 1/(10 001). Cela signifie donc qu'aucune des tables de contingence simulées n'est aussi peu compatible que celle observée ici. On en déduit donc que l'association entre les clusters trouvés par l'algorithme et les catégories est très significative.

Il est intéressant de noter, après discussion avec les participantes, que la quasi-totalité de ces dernières a regroupé les cris en utilisant un critère d'expressivité du bébé dans la première partie de l'expérience. C'est-à-dire qu'elles ont regroupé ensemble les cris qui, selon elles, expriment les mêmes choses. Il y aurait ici indication d'un possible lien entre une classe de contour de pitch et une expression (volontaire ou involontaire) de la part de l'enfant.

3.5 Tests statistiques sur la distribution des différents clusters trouvés

On présente dans cette section une analyse statistique de l'utilisation des profils prosodiques extraits par l'algorithme, après validation perceptive, en fonction de l'âge et du contexte du cri. Dans ce cas, la variable dépendante à analyser est définie comme un vecteur constitué des numéros d'indexation des contours de pitch aux différents clusters trouvés par l'algorithme. Le formalisme relatif aux LMEs est malheureusement inutilisable ici, la variable dépendante étant une variable multinomiale. On utilise pour ce faire un General Linear Mixed-Effects Models (GLME).

3.5.1 Introduction au GLME

Les GLME [46] étendent les modèles mixtes (décrit section 2.2) en permettant à la variable dépendante $\mathbf{y}$ de ne pas suivre une distribution gaussienne. Le modèle linéaire est alors relié à la variable réponse via une fonction lien. On pose $\boldsymbol{\eta}$, définie tel que :

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\Upsilon} \tag{3.10}$$

On définie également la fonction lien $g(\cdot)$. La fonction lien permet de relier la variable dépendante $\mathbf{y}$ et le prédicateur linéaire $\boldsymbol{\eta}$. L'espérance conditionnelle de $\mathbf{y}$ est alors définie telle que :

$$g(E(\mathbf{y})) = \boldsymbol{\eta} \tag{3.11}$$

La fonction binomiale dans le cas d'une sortie binomiale est, par exemple, la fonction logistique. C'est à dire :

$$g(p) = log_e(\frac{p}{1-p}) \tag{3.12}$$

GLME (binomial) - Design 3 : 8 months and all babies	Distrib cluster 2	Distrib cluster 3	Distrib cluster 4	Distrib cluster 5	Distrib cluster 6	Distrib cluster 7	Distrib cluster 8
month	X2(1) = 7.1121, p = 0.007656	0.09064	0.7849	0.390737	0.2935	0.8687	X(1) = 5.0472, p = 0.02467
context	X2(2) = 7.4868, p = 0.023674	X2(2) = 7.2695, p = 0.02639	0.6692	X2(2) = 12.3233, p = 0.002109	0.7081	0.1865	0.25195
month:context	0.333627	0.26712	0.8605	0.566752	0.3554	0.4302	0.49062

GLME (binomial) - Design 3 : 8 months and all babies	Distrib cluster 2	Distrib cluster 3	Distrib cluster 4	Distrib cluster 5	Distrib cluster 6	Distrib cluster 7	Distrib cluster 8
sex	0.177363	0.096167	0.1210	X2(1) = 4.0406, p = 0.044417	0.93242	0.1626	0.67031
month	X2(1) = 6.7932, p = 0.009151	0.056825	0.7760	0.421205	0.29485	0.7888	X2(1) = 5.1345, p = 0.02346
sex:month	0.060047	0.345023	0.1118	1.000000	0.77093	0.4644	1.00000
context	X2(2) = 6.6631, p = 0.035738	X(2) = 6.5812, p = 0.037231	0.4677	X2(2) = 13.3995, p = 0.001231	0.70596	0.3839	0.22451
sex:context	0.998941	0.06535	X(2) = 9.8916, p = 0.007113	X2(2) = 8.1081, p = 0.017352	X2(2) = 8.6248, p = 0.0134	0.2331	0.69450
month:context	0.292032	0.252999	0.920904	0.603056	0.35587	0.4432	0.82212
sex:month:context	0.742579	X(5) = 15.1666, p = 0.009674	X(5) = 0.002958, p = 0.002958	0.095199	0.07329	0.6235	0.91173

Figure 3.15 – Valeur de p pour la distribution des clusters de contours de pitch selon plusieurs effets fixes dans le cas de tests GLME avec sortie binomiale selon plusieurs designs d'expérience. Ici les effets statistiquement significatifs sans correction de Bonferroni sont colorés en jaune et ceux significatifs avec correction de Bonferroni sont colorés en orange

3.5.2 Approximation de Begg et Gray

La variable dépendante **y** est, dans notre cas, une variable suivant une distribution multinomiale (une distribution sur 7 types de contours de pitch). Begg et Gray montrent dans [15] que l'on peut estimer un modèle de régression multinomiale par un ensemble de modèles de régression binomial simple, c'est l'approximation de Begg et Gray. Cela permet de réduire grandement la complexité temporelle d'un tel algorithme. Pour cela, on réduit, pour chaque cluster trouvé, la base de données de contours de pitch à seulement deux index possibles : 0 si le contour appartient au cluster considéré, 1 sinon. On estime alors, pour chaque cluster trouvé, un modèle mixte de régression binomiale en utilisant un GLME avec une sortie binomiale.

3.5.3 Choix des effets testés et du design

On testera les mêmes effets et aléatoires que ceux définis pour le test LME dans 2.3. De même, on considérera un coefficient directeur aléatoire pour l'effet contexte, l'effet mois et l'interaction entre les deux. On ne considérera également pas que l'intercept et les différents coefficients directeurs de chaque bébé sont corrélés. Le design choisi est le même que le design LME 3 définit dans 2.3, c'est-à-dire : 3 contextes, tous les bébés, les 8 premiers mois.

3.5.4 Analyse des résultats

Présentation des résultats

Les résultats sont donnés dans le Tableau 3.15. On présente dans ce tableau les paramètres et valeurs des différents tests seulement dans le cas d'effets statistiquement significatifs avec ou sans correction de Bonferroni (ici tel que $p < 0.05/7$). Contrairement à la section 2.4, on s'intéressera également ici aux résultats significatifs avec et sans correction de Bonferroni. En effet, il n'est pas clair dans le cas de l'approximation de Begg et Gray s'il est nécessaire, ou pas, de prendre en compte la correction de Bonferonni. De plus, Begg met en avant dans [15], que l'approximation peut être considérée comme conservative.

Le travail présenté dans ce mémoire étant à caractère exploratoire, et particulièrement dans cette section, on cherche à découvrir le plus de pistes de réflexions possibles. On analysera donc dans cette partie tous les résultats présentant une p-valeur telle que $p < 0.05$. Dans le futur, il pourra être intéressant de ne pas faire l'approximation de Begg et Gray et d'estimer un modèle mixte de régression multinomiale.

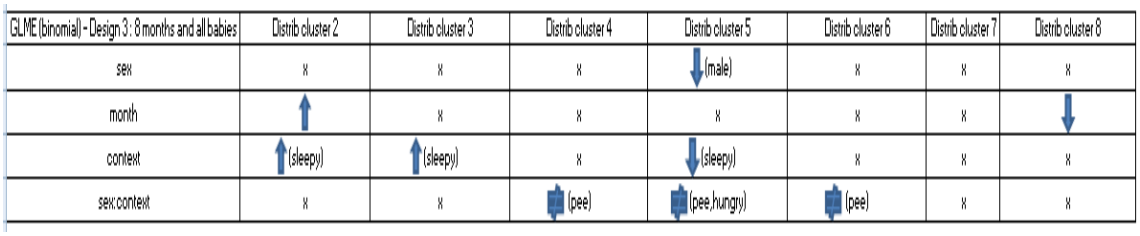


Figure 3.16 – Sens de variation des effets statistiquement significatifs mis en avant pour la distribution des contours de pitch. Dans le cas de l'effet "mois", ↑ indique une augmentation du paramètre correspondant avec l'âge. Dans le cas de l'effet "context", ↑ (contexte) indique une augmentation de ce paramètre pour le (ou les) contexte précisé entre parenthèses. De même pour ↑ (sexe). Inversement, ↓ indique une diminution du paramètre avec l'âge, pour un contexte ou pour un sexe. Dans le cas de l'effet "sex:context", ≠(contexte) indique une différence de ce paramètre entre les deux sexes pour le (ou les) contexte précisé entre parenthèse. Une croix indique une absence de variation significative

Analyse et discussions

De manière équivalente à la section 2.4, dans le cas où la variation de la distribution d'un cluster est statistiquement significative, il est nécessaire de visualiser la distribution du cluster en fonction de l'effet statistiquement significatif pour savoir comment il varie. La figure 3.16 détaille dans quel sens varie les distributions en fonction des effets significatifs, après visualisation des différentes courbes. On ne considérera pas l'effet d'interaction "sex:month:context" qui est trop compliqué à interpréter.

On observe, que contrairement à la section 2.4, le sexe du bébé a un impact sur la fréquence des clusters trouvés. Cependant, Lee énonce, dans [43] que la différentiation entre les deux sexes ne commence qu'à partir de la 9e année. Cette observation rentrerait donc en conflit avec nos observations. On observe en effet, ici, un effet principal du sexe sur la fréquence du cluster 5 (voir figure, 3.22), les enfants de sexe mâle ayant une utilisation moins fréquence des contours de pitch de ce mode. On observe également une différence de fréquence significative entre les deux sexes dans le cas du contexte pee pour le cluster 4 (voir figure 3.20), 5 (voir figure 3.22) et 6 (voir figure 3.23). La fréquence d'utilisation des différents modes de contours prosodiques dans le cas d'un contexte "pee" serait donc bien différenciée selon le sexe. De plus, on observe une différence de fréquence significative entre les deux sexes dans le cas du contexte hungry pour le cluster 5 (voir figure 3.22).

Le contexte sleepy semble avoir également un effet, indépendamment du sexe, sur la fréquence des clusters trouvés. En effet, la fréquence du cluster 2 et 3 est plus forte pour le contexte sleepy (voir respectivement la figure 3.18 et la figure 3.19). Cela nous conforte dans l'idée, évoquée section 3.4, que le cluster 2 et 3 sont similaires. De manière inverse, la fréquence du cluster 5 (voir figure 3.21) est plus faible pour ce contexte.

De même, le mois semble avoir également un effet, indépendamment du sexe, sur la fréquence des clusters trouvés. La fréquence du cluster 2 est, par exemple, croissante avec l'âge (voir figure 3.17). La fréquence du cluster 8 est, quant à elle, décroissante avec l'âge (voir figure 3.24).

Il intéressant de constater que les clusters les plus informatifs pour le contexte (ici le 2 et 3, 4 et 5) sont également des clusters qui sont bien définis perceptivement dans la section précédente. La littérature liant des contours de pitch à des contextes de production étant malheureusement assez faible, il est difficile de la relier aux dépendances statistique significatives observées ici. Hsu dans [34] formule des liens possibles entre les contours observés et les états émotionnels exprimés, mais sur une sélection de contextes différente (et bien plus complexe) de celle considérée dans notre cas. Il n'est donc pas possible de relier cette étude à la nôtre.

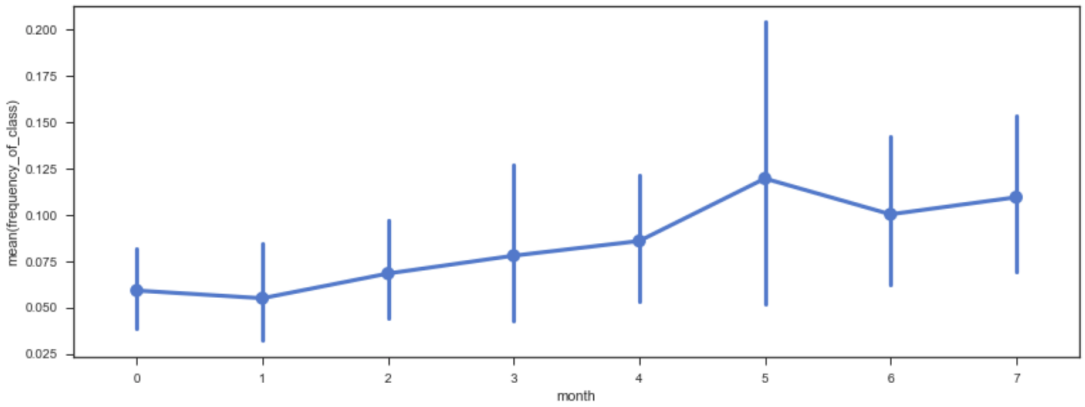


Figure 3.17 – Fréquence du cluster 2, moyennée sur 31 bébés, pour différents âges (de 0 à 7 mois). Une différence statistiquement significative de la fréquence en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

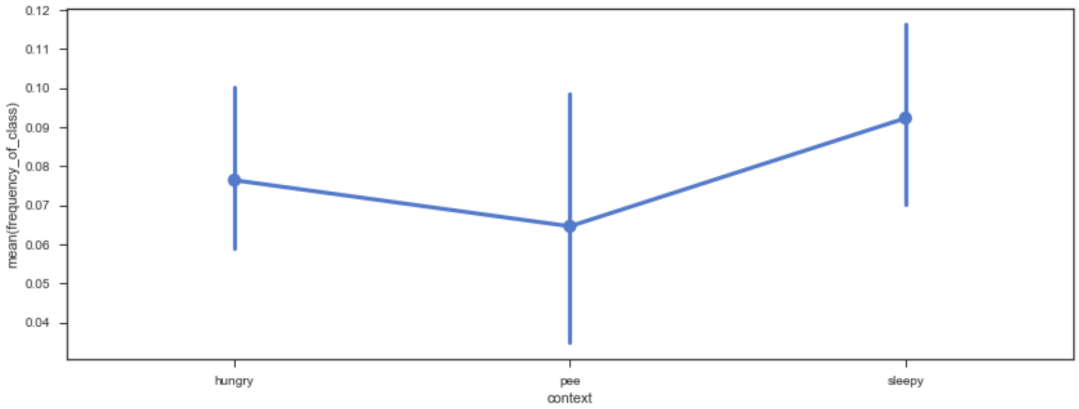


Figure 3.18 – Fréquence du cluster 2, moyennée sur 31 bébés, pour différents contexte (hungry, pee, sleepy). Une différence statistiquement significative de la fréquence en fonction du contexte est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

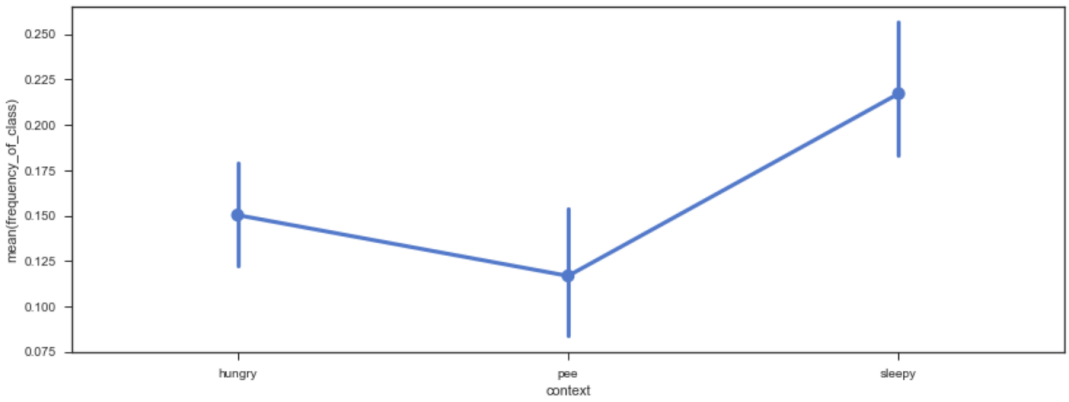


Figure 3.19 – Fréquence du cluster 3, moyennée sur 31 bébés, pour différents contexte (hungry, pee, sleepy). Une différence statistiquement significative de la fréquence en fonction du contexte est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

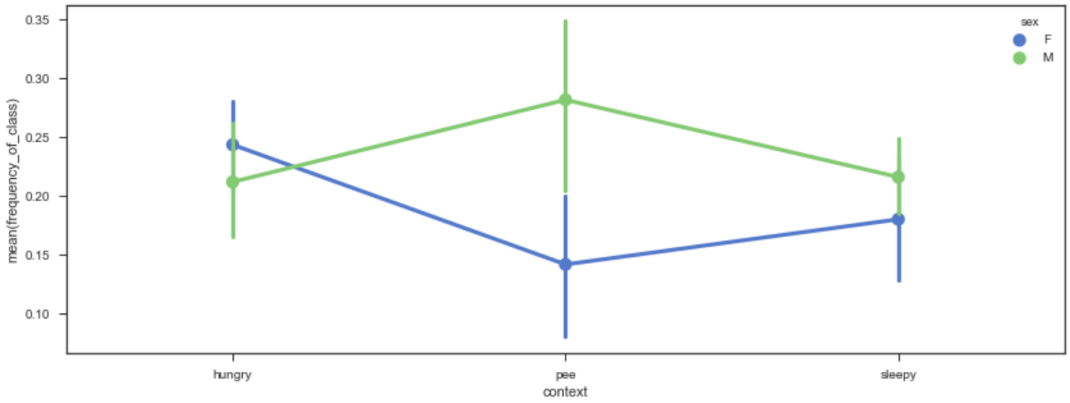


Figure 3.20 – Fréquence du cluster 4, moyennée sur 31 bébés, pour différents contexte (hungry, pee, sleepy) et différent sexe (male et femelle). Une différence statistiquement significative de la fréquence en fonction du couple (contexte, sexe) est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

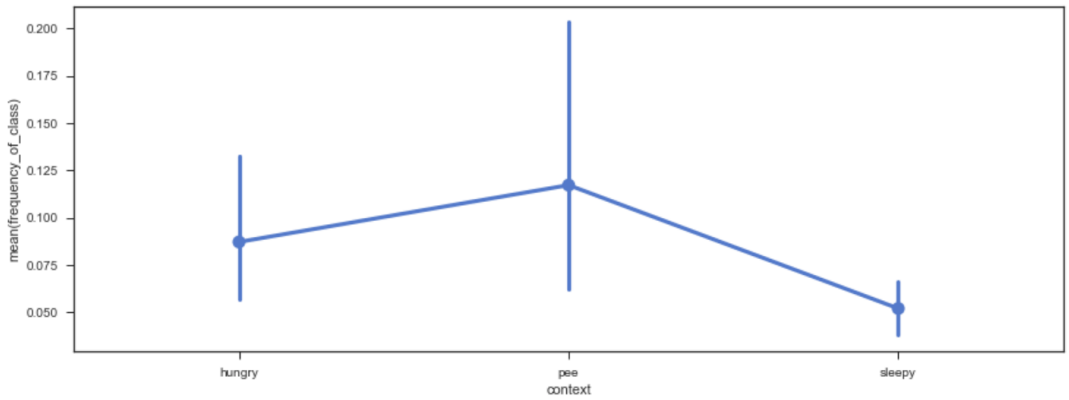


Figure 3.21 – Fréquence du cluster 5, moyennée sur 31 bébés, pour différents contexte (hungry, pee, sleepy). Une différence statistiquement significative de la fréquence en fonction du contexte est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

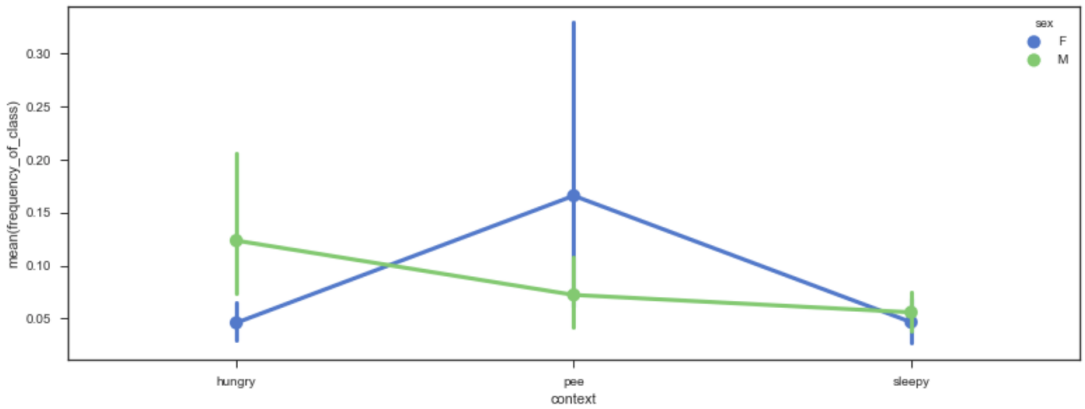


Figure 3.22 – Fréquence du cluster 5, moyennée sur 31 bébés, pour différents contexte (hungry, pee, sleepy) et différent sexe (male et femelle). Une différence statistiquement significative de la fréquence en fonction du couple (contexte, sexe) et du sexe est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

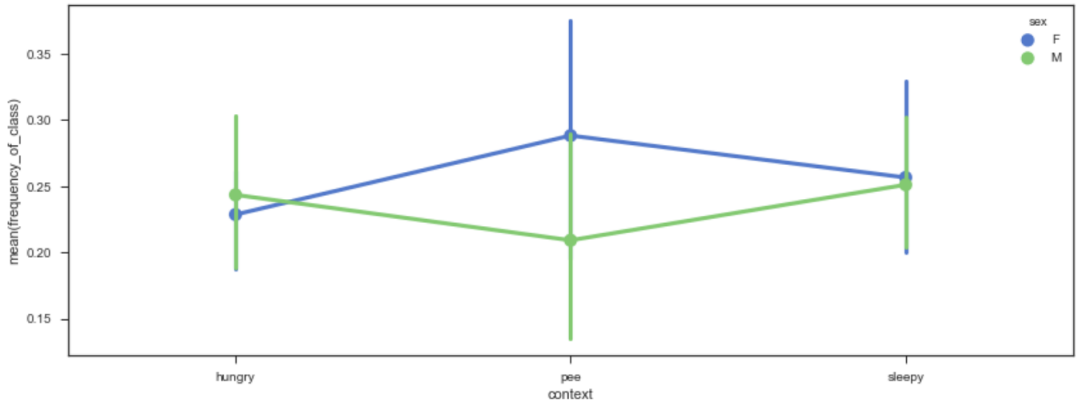


Figure 3.23 – Fréquence du cluster 6, moyennée sur 31 bébés, pour différents contexte (hungry, pee, sleepy) et différent sexe (male et femelle). Une différence statistiquement significative de la fréquence en fonction du couple (contexte, sexe) est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

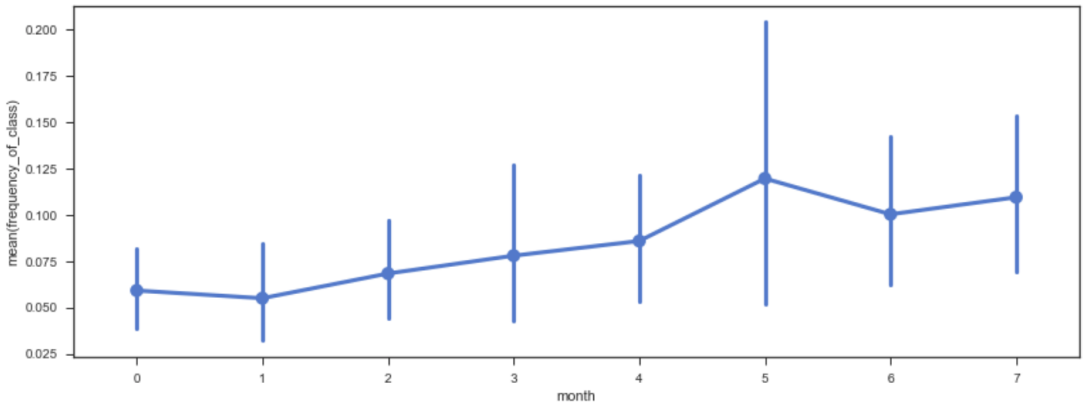


Figure 3.24 – Fréquence du cluster 8, moyennée sur 31 bébés, pour différents âges (de 0 à 7 mois). Une différence statistiquement significative de la fréquence en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

CONCLUSION ET PERSPECTIVES

Travaux accomplis

Les travaux accomplis lors de ce stage sont les suivants :

D'une part, une nouvelle implémentation d'un algorithme de segmentation inspiration/expiration a été réalisée, en Python là où l'original était en Matlab. Le modèle présenté est un modèle de Markov dont la séquence des états est directement observable dans la base d'entraînement. Les probabilités d'émissions sont estimées à chaque instant en convertissant la décision de classification d'un SVM par Platt scaling. La principale amélioration par rapport à l'algorithme de JJ.A dans [12] est l'utilisation d'une validation croisée pour l'estimation du modèle probabilistique en sortie du SVM. On obtient alors une accuracy moyenne de 89.02% avec une validation croisée sur la base annotée contre 85.62% dans [12].

D'autre part, un ensemble de caractéristiques statiques ont été extraites des portions expirées des cris du corpus puis analysées statistiquement en fonction de l'âge et du contexte du cri en utilisant le formalisme des modèles linéaires à effets mixtes. Il a alors principalement été mis en avant l'influence significative du mois sur la durée des expirations et des inspirations et sur la rugosité de la voix de l'enfant, notamment la baisse significative de cette dernière entre le troisième et le quatrième mois. Ce résultat est en accord avec la littérature physiologique, et constitue la première confirmation acoustique de ce phénomène sur un corpus de cette taille.

Enfin, un algorithme de clustering automatique de profils prosodiques optimisé pour les grosses bases de données (plusieurs dizaines de milliers d'unités) a été implémenté. Son application au corpus constitué des contours de pitch extraits des cris expirés a alors permis de définir 7 modes de cris représentatifs du corpus. Ces contours de pitch prototypiques ont ensuite été validés perceptivement en testant l'association statistique entre ces derniers et des catégories perceptives construites à partir des résultats d'une expérience de tri libre réalisée sur un échantillon de mamans. Puis la répartition des contours de pitch a été analysé statistiquement en utilisant un ensemble de GLMEs. Il a été mis en avant des variations significatives de la distribution de ces profils en fonction du sexe, du contexte et de l'âge des enfants.

Perspectives

Les perspectives d'évolution des travaux présentés dans ce mémoire sont nombreuses.

En restant au plus près des techniques développées ici, de nombreuses variantes algorithmiques sont possibles pour étudier d'autres aspects du corpus. Par exemple, au-delà de la simple F0, il serait possible d'étudier les caractéristiques timbrales de la voix des bébés, et en particulier la position des premiers formants. Baeck [13] montre en particulier une différence du F1 moyen entre des cris d'inconfort et ceux de douleur, et ce, indépendamment de la F0 des cris [13]).

Quelles que soient les régularités statistiques découvertes au cours de ce travail, et de ses extensions futures, il est à noter que l'analyse statistique fournie ici ne permet en aucun cas de conclure de façon "mécanistique" sur l'utilisation de telle ou telle caractéristique acoustique par les bébés pour communiquer telle ou telle intention [20]. Nos présents résultats sont seulement d'ordre corrélationnel : dans ce corpus, les bébés exprimant, par exemple, le besoin de sommeil l'ont fait avec des cris correspondant en plus grande proportion aux clusters 2 et 3. Pour établir une causalité entre ce type de prosodie et la communication de ce besoin, il faudrait réaliser une "expérience de perturbation" où des cris sont modifiés acoustiquement pour présenter un type de prosodie ou un autre, et où l'on testera l'impact de cette manipulation sur la perception du contexte du cri par les parents. En d'autres termes, le présent travail a découvert des patterns, qu'il convient maintenant de valider expérimentalement. Ce travail, plus purement psychologique, sera mené à la suite du stage par l'équipe CREAM, en collaboration avec le laboratoire japonais.

Si tout le code Python nécessaire aux analyses est d'ores et déjà disponible sous forme de notebook *notebook* ré-utilisable par la communauté, les objectifs de dissémination de ce travail énoncés dans l'introduction n'ont pas été complètement atteints. En particulier, l'architecture de ce code devra être adaptée pour permettre d'accéder aux données audio du corpus, possiblement via un répertoire publié avec des outils comme Nature Scientific Data (http://www.nature.com/sdata/) ou Figshare (https://figshare.com/). Enfin, l'algorithme de clustering prosodique et sa validation expérimentale pourront également être présenté sous la forme d'un article de recherche, par exemple dans la revue Journal of the Acoustical Society of America. Il est à noter que ce type d'algorithme (découverte de patterns prosodiques dans de grands corpus) pourrait notamment être intéressant pour d'autres types de données que les cris de bébés, comme les vocalisations animales. Picot [58] présente en particulier un travail similaire sur les chants de baleine à bosse, mais avec un paradigme plus limité de k-mean et distance euclidienne.

La lecture de la littérature met également en avant le nombre limité de contextes de cri utilisé dans ce corpus. De nombreux autres contextes auraient, en effet, été pertinents pour nos analyses des caractéristiques statiques et dynamiques des cris. On peut par exemple déplorer l'absence d'indicateurs comme la douleur, la fatigue, ou les émotions dans notre éventail de contextes disponibles. Fuller dans [29] met par exemple en avant une différence significative pour le pitch moyen des expirations entre le contexte hungry et le contexte pain. Il serait donc nécessaire de construire une base de données plus conséquente et possédant un échantillon de contexte plus large afin de lui appliquer les différents outils développés au cours du stage. Scheiner dans [63] utilise lui 7 contextes différents : joy, contentment, interest, surprise, unease, anger, and pain.

Enfin, on pourrait également s'intéresser à l'étude des séquences de descripteurs et de leur distribution. Wermke dans [70] met, par exemple, en avant les quatre types de cris basiques (presentés figure 0.3) représentant les premiers éléments constitutifs de cris complexes, ces cris étant alors constitués de la concaténation d'un ensemble de ces cris simples. De la même manière, il serait intéressant d'étudier la durée de chacun des éléments constitutifs de ces cris complexes. C'est-à-dire, musicalement, d'étudier le rythme de la voix de l'enfant, là où l'on s'est limité pour l'instant à étudier l'intonation de chaque note.

BIBLIOGRAPHIE

[1] Github de la bibliotheque pyclustering. Web resource, available: https://pypi.python.org/pypi/pyclustering.

[2] Site internet de la bibliothèque rpy2. Web resource, available: https://rpy2.readthedocs.io/.

[3] Site internet de l'ircam. Web resource, available: https://www.ircam.fr/.

[4] Site internet de l'outil praat. Web resource, available: http://www.praat.org/.

[5] Site internet de l'outil sound analysis pro. Web resource, available: http://soundanalysispro.com.

[6] Site internet de l'outil tcl-labx. Web resource, available: http://petra.univ-tlse2.fr/spip.php?article252.

[7] Site internet du laboratoire okanoya. Web resource, available: http://marler.c.u-tokyo.ac.jp/home/en/.

[8] Site internet du projet cream. Web resource, available: http://cream.ircam.fr/.

[9] Vaglio A. Notebook du code du stage, 2017.

[10] Hesam Farsaie Alaie, Lina Abou-Abbas, and Chakib Tadj. Cry-based infant pathology classification using gmms. *Speech communication*, 77:28–52, 2016.

[11] Jean-Julien Aucouturier and Emmanuel Bigand. Seven problems that keep mir from attracting the interest of cognition and neuroscience. *Journal of Intelligent Information Systems*, 41(3):483–497, 2013.

[12] Jean-Julien Aucouturier, Yulri Nonaka, Kentaro Katahira, and Kazuo Okanoya. Segmentation of expiratory and inspiratory sounds in baby cry audio recordings using hidden markov models. *The Journal of the Acoustical Society of America*, 130(5):2969–2977, 2011.

[13] HE Baeck and MN Souza. Study of acoustic features of newborn cries that correlate with the context. In *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, volume 3, pages 2174–2177. IEEE, 2001.

[14] Heidi Elisabeth Baeck and Marcio Nogueira de Souza. Longitudinal study of the fundamental frequency of hunger cries along the first 6 months of healthy babies. *Journal of Voice*, 21(5):551–559, 2007.

[15] COLIN B BECG and Robert Gray. Calculation of polychotomous logistic regression parameters using individualized regressions. *Biometrika*, 71(1):11–18, 1984.

[16] Pavel Berkhin et al. A survey of clustering data mining techniques. *Grouping multidimensional data*, 25:71, 2006.

[17] Christopher M Bishop. Pattern recognition. *Machine Learning*, 128:1–58, 2006.

[18] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the Institute of Phonetic Sciences*, 17:97–110, 1993.

[19] Arturo Camacho. *SWIPE: A sawtooth waveform inspired pitch estimator for speech and music*. University of Florida Gainesville, 2007.

[20] A. Casadevall and F. C. Fang. Descriptive science. *Infection and immunity*, 76(9):3835–3836, 2008.

[21] Anshu Chittora and Hemant A Patil. Newborn infant's cry analysis. *International Journal of Speech Technology*, 19(4):919–928, 2016.

[22] Anshu Chittora and Hemant A Patil. Spectral analysis of infant cries and adult speech. *International Journal of Speech Technology*, 19(4):841–856, 2016.

[23] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[24] Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[25] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[26] Daniel Ellis. Dynamic time warp (dtw) in matlab. Web resource, available: http://www.ee.columbia.edu/ dpwe/resources/matlab/dtw/.

[27] Ronald A Fisher. Xv.—the correlation between relatives on the supposition of mendelian inheritance. *Transactions of the royal society of Edinburgh*, 52(02):399–433, 1919.

[28] Ronald A Fisher. On the interpretation of χ 2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922.

[29] Barbara F Fuller and Yoshiyuki Horii. Differences in fundamental frequency, jitter, and shimmer among four types of infant vocalizations. *Journal of communication disorders*, 19(6):441–447, 1986.

[30] Pascal Gaillard. Laissez-nous trier. *TCL-LabX et les tâches de catégorisation libre de sons, Le sentir et le dire. Concepts et méthodes en psychologie et linguistique cognitive*, 2009.

[31] Ellen R Girden. *ANOVA: Repeated measures*. Number 84. Sage, 1992.

[32] Priscilla E Greenwood and Michael S Nikulin. *A guide to chi-squared testing*, volume 280. John Wiley & Sons, 1996.

[33] Charles R Henderson. Best linear unbiased estimation and prediction under a selection model. *Biometrics*, pages 423–447, 1975.

[34] Hui-Chin Hsu, Alan Fogel, and Rebecca B Cooper. Infant vocal development during the first 6 months: Speech quality and melodic complexity. *Infant and Child Development*, 9(1):1–16, 2000.

[35] John D Hunter. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[36] Dean L Isaacson and Richard W Madsen. *Markov chains, theory and applications*, volume 4. Wiley New York, 1976.

[37] Eric Jones, Travis Oliphant, and Pearu Peterson. {SciPy}: open source scientific tools for {Python}. 2014.

[38] Geraldo Pereira Jotz, Onivaldo Cervantes, Márcio Abrahão, Flávio Aurélio Parente Settanni, and Elisabete Carrara de Angelis. Noise-to-harmonics ratio as an acoustic measure of voice disorders in boys. *Journal of voice*, 16(1):28–31, 2002.

[39] Eamonn J Keogh and Michael J Pazzani. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 285–289. ACM, 2000.

[40] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.

[41] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Stanford, CA, 1995.

[42] Linda L LaGasse, A Rebecca Neal, and Barry M Lester. Assessment of infant cry: acoustic cry analysis and parental perception. *Developmental Disabilities Research Reviews*, 11(1):83–93, 2005.

[43] Sungbok Lee, Alexandros Potamianos, and Shrikanth Narayanan. Acoustics of children's speech: Developmental changes of temporal and spectral parameters. *The Journal of the Acoustical Society of America*, 105(3):1455–1468, 1999.

[44] Beth Logan et al. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, 2000.

[45] Birgit Mampe, Angela D Friederici, Anne Christophe, and Kathleen Wermke. Newborns' cry melody is shaped by their native language. *Current biology*, 19(23):1994–1997, 2009.

[46] Charles E McCulloch and John M Neuhaus. *Generalized linear mixed models*. Wiley Online Library, 2001.

[47] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, 2015.

[48] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. van der Voort S, Millman J, 2010.

[49] Meinard Müller. *Information retrieval for music and motion*, volume 2. Springer, 2007.

[50] Thierry Nazzi and Josiane Bertoncini. Before and after the vocabulary spurt: two modes of word acquisition? *Developmental Science*, 6(2):136–142, 2003.

[51] Raymond T. Ng and Jiawei Han. Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016, 2002.

[52] Y Nonaka, K Katahira, R Shiba, and K Okanoya. Development of infant cry acoustics: a basis of musical and linguistic skills. In *Proceedings of the 10th International Conference on Music Perception and Cognition, Sapporo, Japan*, 2008.

[53] Yulri Nonaka, Jean-Julien Aucouturier, Kentaro Katahira, and Kazuo Okanoya. Developmental differentiation in human infant cry through dynamic interaction with caregivers. In *International Ethological Conference, Association for the Study of Animal Behaviour*, 2013.

[54] Stavros Ntalampiras. Audio pattern recognition of baby crying sound events. *Journal of the Audio Engineering Society*, 63(5):358–369, 2015.

[55] Robert F Orlikoff, RJ Baken, and Dennis H Kraus. Acoustic and physiologic characteristics of inspiratory phonation. *The Journal of the Acoustical Society of America*, 102(3):1838–1845, 1997.

[56] C Papaeliou, G Minadakis, and D Cavouras. Acoustic patterns of infant vocalizations expressing emotions and communicative functions. *Journal of Speech, Language, and Hearing Research*, 45(2):311–317, 2002.

[57] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[58] Gautier Picot, Olivier Adam, Maïtine Bergounioux, Hervé Glotin, and François-Xavier Mayer. Automatic prosodic clustering of humpback whales song. In *New Trends for Environmental Monitoring Using Passive Systems, 2008*, pages 1–6. IEEE, 2008.

[59] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999.

[60] Athanassios Protopapas and Peter D Eimas. Perceptual differences in infant cries revealed by modifications of acoustic features. *The Journal of the Acoustical Society of America*, 102(6):3723–3734, 1997.

[61] Lawrence R Rabiner and Biing-Hwang Juang. Fundamentals of speech recognition. 1993.

[62] Lior Rokach and Oded Maimon. Clustering methods. In *Data mining and knowledge discovery handbook*, pages 321–352. Springer, 2005.

[63] Elisabeth Scheiner, Kurt Hammerschmidt, Uwe Jürgens, and Petra Zwirner. Acoustic analyses of developmental changes and emotional expression in the preverbal vocalizations of infants. *Journal of Voice*, 16(4):509–529, 2002.

[64] R Core Team. R language definition. *Vienna, Austria: R foundation for statistical computing*, 2000.

[65] João Paulo Teixeira and Paula Odete Fernandes. Jitter, shimmer and hnr classification within gender, tones and vowels in healthy voices. *Procedia Technology*, 16:1228–1237, 2014.

[66] Hiroko Terasawa, Malcolm Slaney, and Jonathan Berger. The thirteen colors of timbre. In *Applications of Signal Processing to Audio and Acoustics, 2005. IEEE Workshop on*, pages 323–326. IEEE, 2005.

[67] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[68] Geert Verbeke. Linear mixed models for longitudinal data. In *Linear mixed models in practice*, pages 63–153. Springer, 1997.

[69] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

[70] Kathleen Wermke and Werner Mende. Musical elements in human infants' cries: in the beginning is the melody. *Musicae Scientiae*, 13(2_suppl):151–175, 2009.

[71] Bodo Winter. A very basic tutorial for performing linear mixed effects analyses. *arXiv preprint arXiv:1308.5499*, 2013.

[72] Philip Sanford Zeskind and Timothy R Marshall. The relation between variations in pitch and maternal perceptions of infant crying. *Child Development*, pages 193–196, 1988.

Annexes A

ALGORITHME DU CHEMIN D'ALIGNEMENT OPTIMAL

Algorithm: OPTIMALWARPINGPATH

Input: Accumulated cost matrix D.
Output: Optimal warping path p^*.

Procedure: The optimal path $p^* = (p_1, \ldots, p_L)$ is computed in reverse order of the indices starting with $p_L = (N, M)$. Suppose $p_\ell = (n, m)$ has been computed. In case $(n, m) = (1, 1)$, one must have $\ell = 1$ and we are finished. Otherwise,

$$p_{\ell-1} := \begin{cases} (1, m-1), & \text{if } n = 1 \\ (n-1, 1), & \text{if } m = 1 \\ \text{argmin}\{D(n-1, m-1), & \\ \qquad D(n-1, m), D(n, m-1)\}, & \text{otherwise,} \end{cases} \tag{4.6}$$

where we take the lexicographically smallest pair in case "argmin" is not unique.

Figure A – Algorithme du chemin d'alignement optimal

Annexes B

ALGORITHME DE VITERBI

Given $x = x_1 x_2 \ldots\ldots x_N$
Find $\pi = \pi_1, \ldots\ldots, \pi_N,$
to maximize $P[x, \pi]$

$\pi^* = \text{argmax}_\pi P[x, \pi]$

Initialization:
$V_0(0) = 1$ (0 is the imaginary first position)
$V_k(0) = 0$, for all $k > 0$

Iteration:
$V_j(i) = e_j(x_i) \times \max_k a_{kj} V_k(i - 1)$

$Ptr_j(i) = \text{argmax}_k a_{kj} V_k(i - 1)$

Termination:
$P(x, \pi^*) = \max_k V_k(N)$

Traceback:
$\pi_N^* = \text{argmax}_k V_k(N)$
$\pi_{i-1}^* = Ptr_{\pi i} (i)$

Figure B – Algorithme de Viterbi

x est la séquence d'état, π^* l'enchaînement d'état le plus probable étant donné la séquence d'état, $e_j(x_i)$ probabilité d'émission de l'état j à l'instant i et a_{kj} probabilité de transition entre l'état k et l'état j

Annexes C

ÉTUDE STATISTIQUE DE LA VARIATION DES CARACTÉRISTIQUES STATIQUES DES CRIS (1/2)

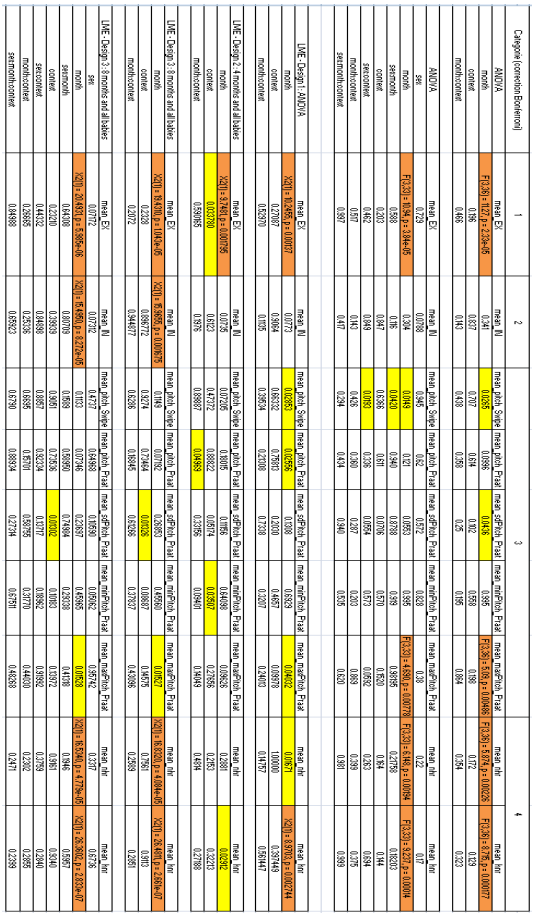


Figure C – Valeur de p pour différentes caractéristiques statiques selon plusieurs effets fixes dans le cas de tests ANOVA et LME selon plusieurs design d'expérience : partie 1. Ici les effets statistiquement significatifs sans correction de Bonferroni sont colorés en jaune et ceux significatifs avec correction de Bonferroni sont colorés en orange. Pour alléger le tableau, les paramètres et valeurs des différents tests sont indiqués seulement dans le cas d'effets statistiquement significatifs avec correction de Bonferroni

Annexes D

ÉTUDE STATISTIQUE DE LA VARIATION DES CARACTÉRISTIQUES STATIQUES DES CRIS (2/2)

Figure D – Valeur de p pour différentes caractéristiques statiques selon plusieurs effets fixes dans le cas de tests ANOVA et LME selon plusieurs designs d'expérience : partie 2. Ici les effets statistiquement significatifs sans correction de Bonferroni sont colorés en jaune et ceux significatifs avec correction de Bonferroni sont colorés en orange. Pour alléger le tableau, les paramètres et valeurs des différents tests sont indiqués seulement dans le cas d'effet statistiquement signifiants avec correction de Bonferroni

Annexes E

VISUALISATION DE LA VARIATION DE CARACTÉRISTIQUES STATIQUES EN FONCTION D'EFFETS STATISTIQUEMENT SIGNIFICATIFS DANS LE CAS DU DESIGN ANOVA

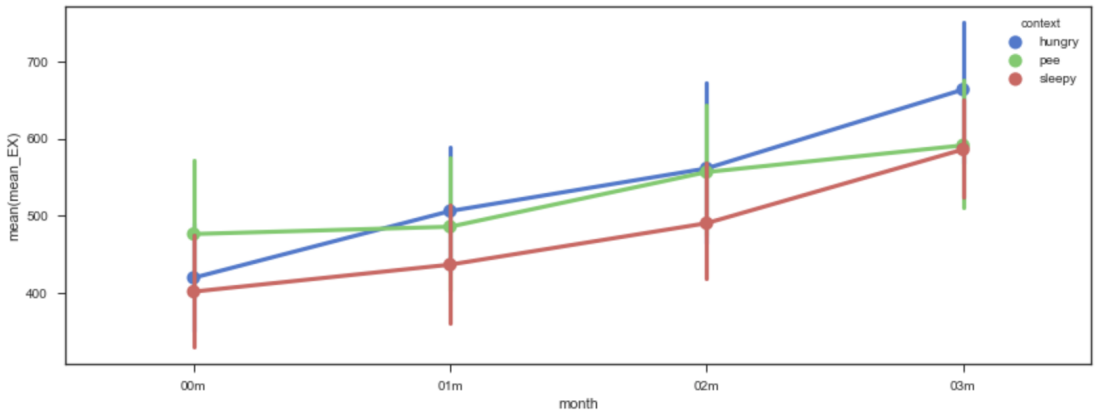


Figure E.1 – Durée moyenne des expirations voisées (en ms), moyennée sur 14 bébés, pour différents contextes (hungry, pee, sleepy) et âge (de 0 à 3 mois). Une différence statistiquement significative de la durée des expirations en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

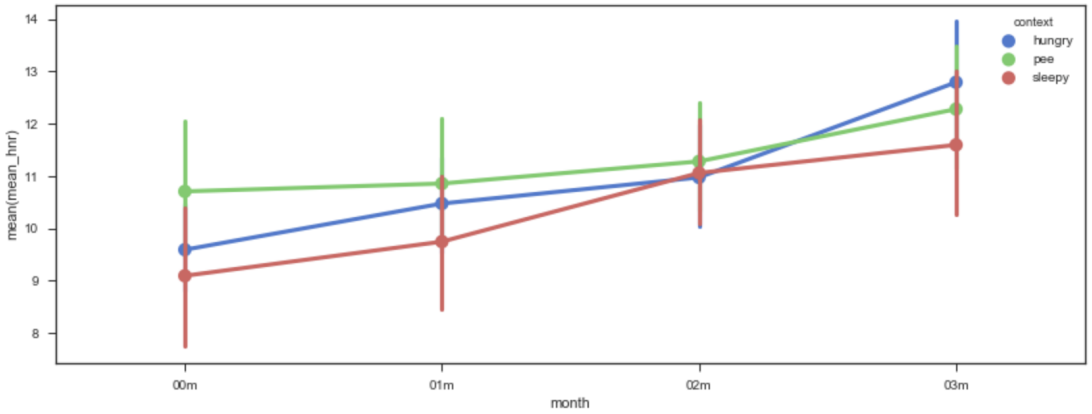


Figure E.2 – Valeur moyenne du rapport harmonique sur bruit (en dB), moyennée sur 14 bébés, pour différents contextes (hungry, pee, sleepy) et âge (de 0 à 3 mois). Une différence statistiquement significative de ce rapport en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

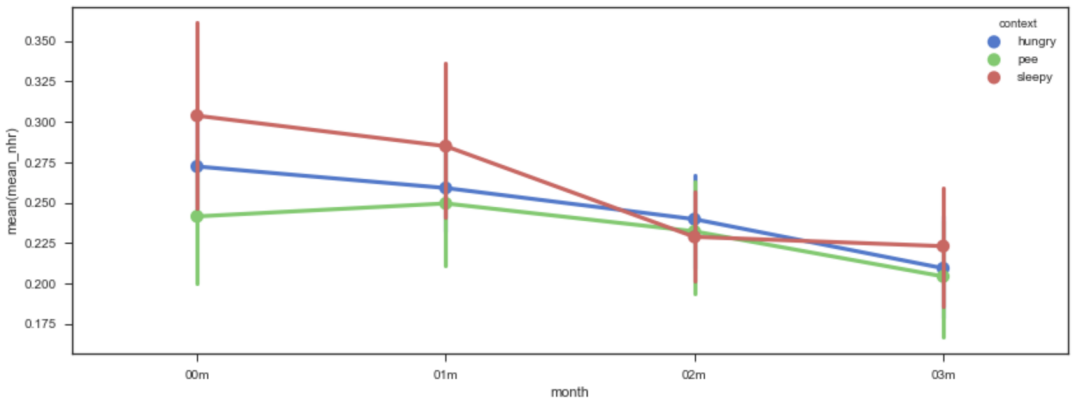


Figure E.3 – Valeur moyenne du rapport bruit sur harmonique (en dB), moyennée sur 14 bébés, pour différents contextes (hungry, pee, sleepy) et âge (de 0 à 3 mois). Une différence statistiquement significative de ce rapport en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

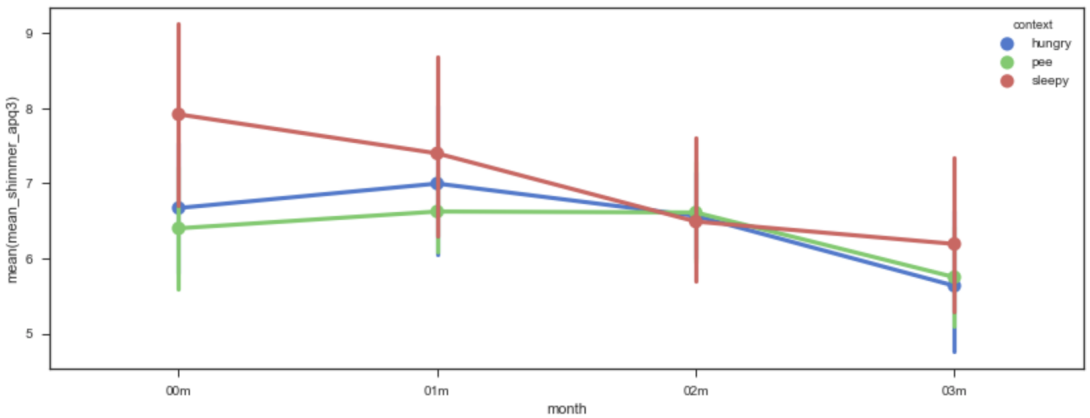


Figure E.4 – Valeur moyenne du shimmer-apq3 (sans dimensions), moyennée sur 14 bébés, pour différents contextes (hungry, pee, sleepy) et âge (de 0 à 3 mois). Une différence statistiquement significative de la valeur du shimmer-apq3 en fonction du mois est observée. Les intervalles de confiance à 95 % sont donnés par les barres d'erreurs

Annexes F

ALGORITHME PAM

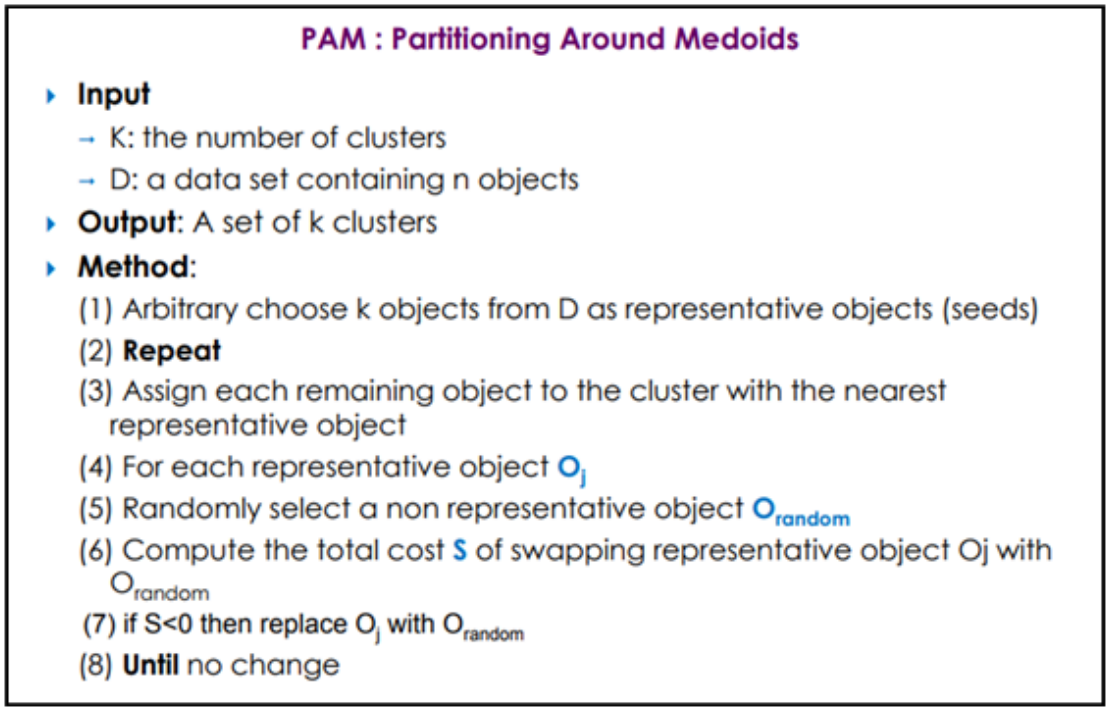


Figure F – Algorithme PAM

Annexes G

ALGORITHME CLARA

Algorithm CLARA

1. For $i = 1$ to 5, repeat the following steps:
2. Draw a sample of $40 + 2k$ objects randomly from the entire data set [1], and call Algorithm PAM to find k medoids of the sample.
3. For each object O_j in the entire data set, determine which of the k medoids is the most similar to O_j.
4. Calculate the average dissimilarity of the clustering obtained in the previous step. If this value is less than the current minimum, use this value as the current minimum, and retain the k medoids found in Step (2) as the best set of medoids obtained so far.
5. Return to Step (1) to start the next iteration. □

Figure G – Algorithme CLARA

Annexes H

ALGORITHME CLARANS

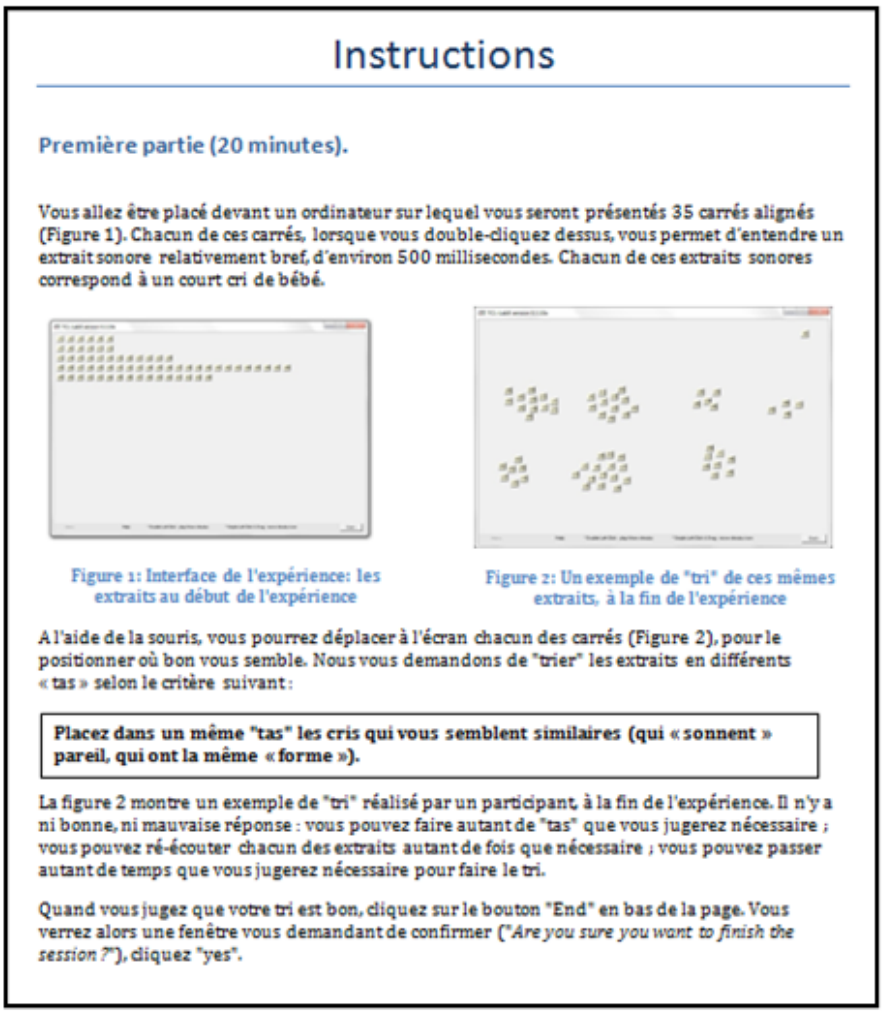
Algorithm CLARANS

1. Input parameters *numlocal* and *maxneighbor*. Initialize i to 1, and *mincost* to a large number.
2. Set *current* to an arbitrary node in $G_{n,k}$.
3. Set j to 1.
4. Consider a random neighbor S of *current*, and based on Equation (5), calculate the cost differential of the two nodes.
5. If S has a lower cost, set *current* to S, and go to Step (3).
6. Otherwise, increment j by 1. If $j \leq maxneighbor$, go to Step (4).
7. Otherwise, when $j > maxneighbor$, compare the cost of *current* with *mincost*. If the former is less than *mincost*, set *mincost* to the cost of *current*, and set *bestnode* to *current*.
8. Increment i by 1. If $i > numlocal$, output *bestnode* and halt. Otherwise, go to Step (2). □

Figure H – Algorithme CLARANS

Annexes I

INSTRUCTIONS TRI LIBRE (1/2)



Instructions

Première partie (20 minutes).

Vous allez être placé devant un ordinateur sur lequel vous seront présentés 35 carrés alignés (Figure 1). Chacun de ces carrés, lorsque vous double-cliquez dessus, vous permet d'entendre un extrait sonore relativement bref, d'environ 500 millisecondes. Chacun de ces extraits sonores correspond à un court cri de bébé.

Figure 1: Interface de l'expérience: les extraits au début de l'expérience

Figure 2: Un exemple de "tri" de ces mêmes extraits, à la fin de l'expérience

A l'aide de la souris, vous pourrez déplacer à l'écran chacun des carrés (Figure 2), pour le positionner où bon vous semble. Nous vous demandons de "trier" les extraits en différents « tas » selon le critère suivant :

Placez dans un même "tas" les cris qui vous semblent similaires (qui « sonnent » pareil, qui ont la même « forme »).

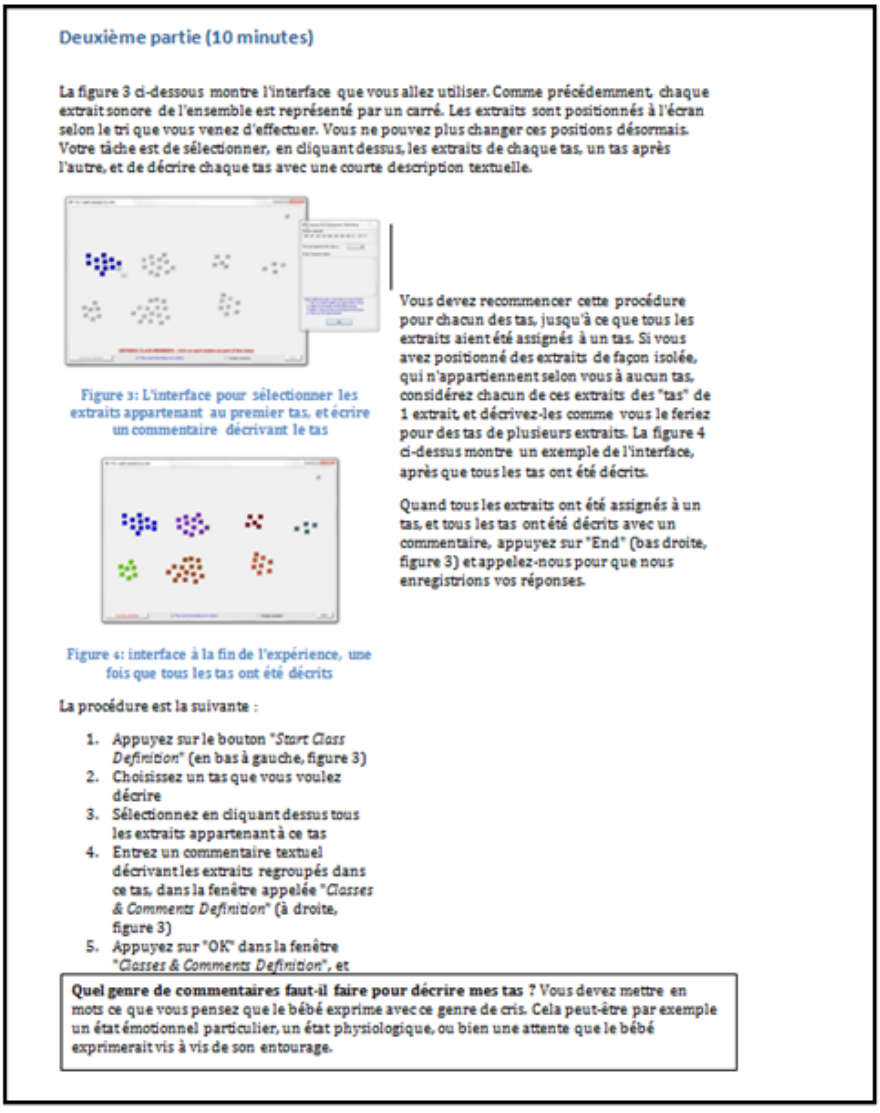
La figure 2 montre un exemple de "tri" réalisé par un participant, à la fin de l'expérience. Il n'y a ni bonne, ni mauvaise réponse : vous pouvez faire autant de "tas" que vous jugerez nécessaire ; vous pouvez ré-écouter chacun des extraits autant de fois que nécessaire ; vous pouvez passer autant de temps que vous jugerez nécessaire pour faire le tri.

Quand vous jugez que votre tri est bon, cliquez sur le bouton "End" en bas de la page. Vous verrez alors une fenêtre vous demandant de confirmer (*"Are you sure you want to finish the session ?"*), cliquez "yes".

Figure I – Instructions tri libre : partie 1

Annexes J

INSTRUCTIONS TRI LIBRE (2/2)



Deuxième partie (10 minutes)

La figure 3 ci-dessous montre l'interface que vous allez utiliser. Comme précédemment, chaque extrait sonore de l'ensemble est représenté par un carré. Les extraits sont positionnés à l'écran selon le tri que vous venez d'effectuer. Vous ne pouvez plus changer ces positions désormais. Votre tâche est de sélectionner, en cliquant dessus, les extraits de chaque tas, un tas après l'autre, et de décrire chaque tas avec une courte description textuelle.

Figure 3: L'interface pour sélectionner les extraits appartenant au premier tas, et écrire un commentaire décrivant le tas

Vous devez recommencer cette procédure pour chacun des tas, jusqu'à ce que tous les extraits aient été assignés à un tas. Si vous avez positionné des extraits de façon isolée, qui n'appartiennent selon vous à aucun tas, considérez chacun de ces extraits des "tas" de 1 extrait, et décrivez-les comme vous le feriez pour des tas de plusieurs extraits. La figure 4 ci-dessus montre un exemple de l'interface, après que tous les tas ont été décrits.

Quand tous les extraits ont été assignés à un tas, et tous les tas ont été décrits avec un commentaire, appuyez sur "End" (bas droite, figure 3) et appelez-nous pour que nous enregistrions vos réponses.

Figure 4: interface à la fin de l'expérience, une fois que tous les tas ont été décrits

La procédure est la suivante :

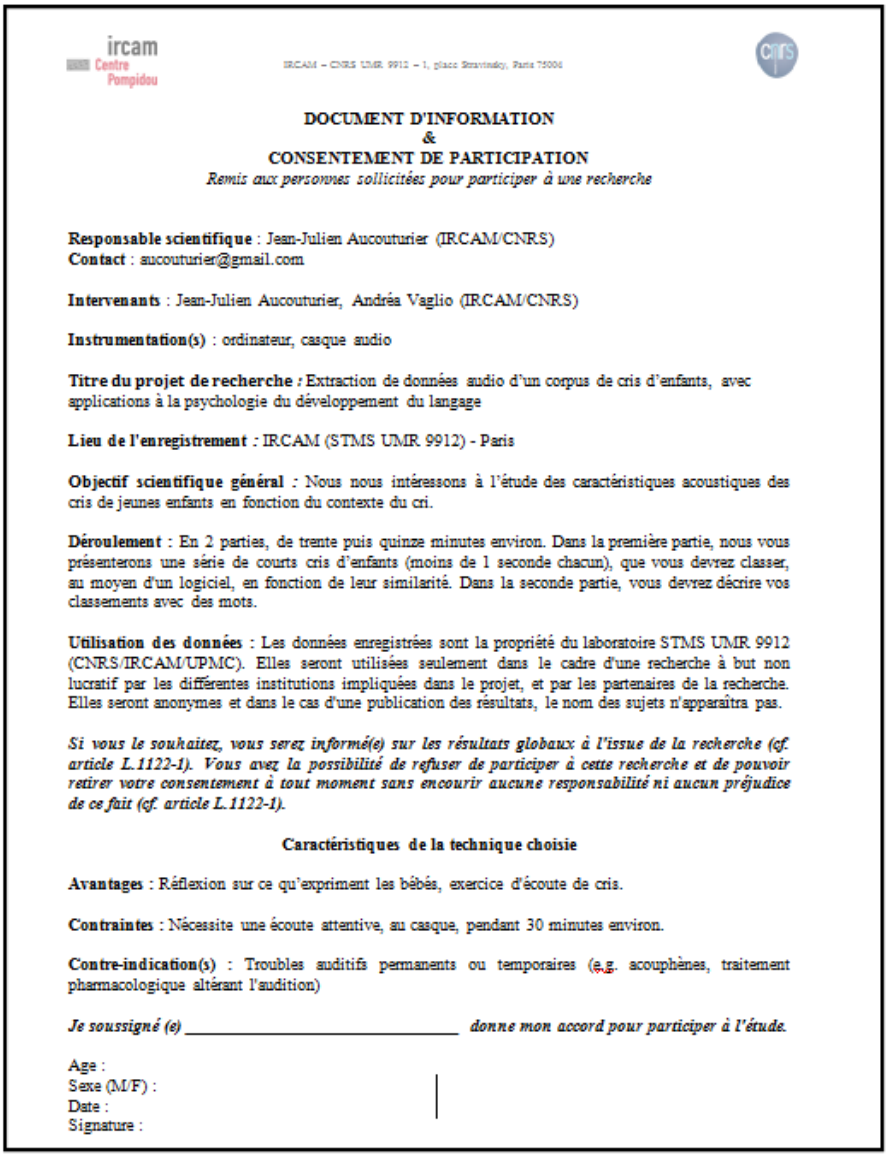
1. Appuyez sur le bouton "*Start Class Definition*" (en bas à gauche, figure 3)
2. Choisissez un tas que vous voulez décrire
3. Sélectionnez en cliquant dessus tous les extraits appartenant à ce tas
4. Entrez un commentaire textuel décrivant les extraits regroupés dans ce tas, dans la fenêtre appelée "*Classes & Comments Definition*" (à droite, figure 3)
5. Appuyez sur "OK" dans la fenêtre "*Classes & Comments Definition*", et

Quel genre de commentaires faut-il faire pour décrire mes tas ? Vous devez mettre en mots ce que vous pensez que le bébé exprime avec ce genre de cris. Cela peut-être par exemple un état émotionnel particulier, un état physiologique, ou bien une attente que le bébé exprimerait vis à vis de son entourage.

Figure J – Instructions tri libre : partie 2

Annexes K

FORMULAIRE DE CONSENTEMENT POUR EXPÉRIENCE DU TRI LIBRE



ircam Centre Pompidou

IRCAM – CNRS UMR 9912 – 1, place Stravinsky, Paris 75004

DOCUMENT D'INFORMATION
&
CONSENTEMENT DE PARTICIPATION
Remis aux personnes sollicitées pour participer à une recherche

Responsable scientifique : Jean-Julien Aucouturier (IRCAM/CNRS)
Contact : aucouturier@gmail.com

Intervenants : Jean-Julien Aucouturier, Andréa Vaglio (IRCAM/CNRS)

Instrumentation(s) : ordinateur, casque audio

Titre du projet de recherche *:* Extraction de données audio d'un corpus de cris d'enfants, avec applications à la psychologie du développement du langage

Lieu de l'enregistrement *:* IRCAM (STMS UMR 9912) - Paris

Objectif scientifique général *:* Nous nous intéressons à l'étude des caractéristiques acoustiques des cris de jeunes enfants en fonction du contexte du cri.

Déroulement : En 2 parties, de trente puis quinze minutes environ. Dans la première partie, nous vous présenterons une série de courts cris d'enfants (moins de 1 seconde chacun), que vous devrez classer, au moyen d'un logiciel, en fonction de leur similarité. Dans la seconde partie, vous devrez décrire vos classements avec des mots.

Utilisation des données : Les données enregistrées sont la propriété du laboratoire STMS UMR 9912 (CNRS/IRCAM/UPMC). Elles seront utilisées seulement dans le cadre d'une recherche à but non lucratif par les différentes institutions impliquées dans le projet, et par les partenaires de la recherche. Elles seront anonymes et dans le cas d'une publication des résultats, le nom des sujets n'apparaîtra pas.

Si vous le souhaitez, vous serez informé(e) sur les résultats globaux à l'issue de la recherche (cf. article L.1122-1). Vous avez la possibilité de refuser de participer à cette recherche et de pouvoir retirer votre consentement à tout moment sans encourir aucune responsabilité ni aucun préjudice de ce fait (cf. article L.1122-1).

Caractéristiques de la technique choisie

Avantages : Réflexion sur ce qu'expriment les bébés, exercice d'écoute de cris.

Contraintes : Nécessite une écoute attentive, au casque, pendant 30 minutes environ.

Contre-indication(s) : Troubles auditifs permanents ou temporaires (e.g. acouphènes, traitement pharmacologique altérant l'audition)

Je soussigné (e) _____ donne mon accord pour participer à l'étude.

Age :
Sexe (M/F) :
Date :
Signature :

Figure K – Formulaire de consentement de participation