Algorithmes de post-traitement

Sommaire

8.1 Règles heuristiques

On trouve de nombreux exemples dans la littérature de règles heuristiques destinées à corriger la présence d'éventuelles erreurs de classification sur la séquence des classes estimées $\hat{y}_i \in \{1, \ldots, C\}$ associées à la suite d'exemples $\boldsymbol{x}_i$, où i suit la séquence temporelle des trames. Ces heuristiques ciblent généralement la correction d'erreurs marginales (*outliers*). On peut définir ces dernières de manière informelle comme la présence accidentelle d'un label A dans une séquence longue de labels $B \neq A$.

La règle la plus simple [121][139] consiste donc à simplement remplacer toutes les occurrences de la séquence ABA par la séquence AAA :

- $ABA \rightarrow AAA$

Certains auteurs [260][261] préfèrent au préalable réunir les trames consécutives de même label en segments homogènes pour prendre en compte la durée de ces segments dans la détection d'erreurs marginales. Néanmoins ce procédé revient de fait à induire des règles heuristiques sur des séquences plus longues afin de prendre en compte un voisinage plus large de labels.

Ainsi, dans [52], Chou et Gu définissent un ensemble de règles plus complexes portant sur des séquences de longueurs variables s'échelonnant de 3 à 7 trames, par exemple :

- $AABAA \rightarrow AAAAA$
- $AABBAAA \rightarrow AAAAAAA$

celles-ci sont complétées par des règles empiriques plus fines guidées par la nature des classes mise en jeu. Dans le cadre d'un problème portant sur les quatre classes de parole, musique, chant et bruit, respectivement représentées par les labels « S », « M », « A » et « N » [1], les auteurs définissent par exemple certaines règles destinées à corriger les transitions erronées entre le bruit et les classes de parole et de chant (on note $[A|B]$ un label pouvant prendre indifféremment les valeurs A ou B) :

1. bien que le symbole « A » pour le chant soit contre-intuitif, nous reproduisons ici les notations des auteurs.

1. $N[M|A]SSS \rightarrow NSSSS$
2. $SSS[M|A]N \rightarrow SSSSN$
3. $N[M|S]AAA \rightarrow NAAAA$
4. $AAA[M|S]N \rightarrow AAAAN$
5. $NN[M|A][M|A]SSS \rightarrow NNSSSSS$
6. $SSS[M|A][M|A]NN \rightarrow SSSSSNN$

Les règles 2, 4 et 6 sont les symétriques des règles 1, 3 et 5. On voit que l'auteur choisit ici d'avantager implicitement les classes non bruitées lors de transitions marginales. On peut construire ainsi de nombreuses règles plus complexes encore pour prendre en compte les particularités de chacune des classes par rapport aux autres. Il devient cependant de plus en plus difficile de contrôler l'absence de contradiction entre les règles heuristiques édictées. Il est en outre aisé de construire des séquences indécidables au regard des règles habituelles, en particulier l'alternance entre deux classes, ou certains schémas de transition plus complexes :

- $AAAABABABABBBB$
- $AAAACBACCCC$
- $AAAABBABBAAAA$

Zhang et al. [262] contournent ce problème en reclassifiant les trames de transition marginale (qu'ils définissent comme des sous-séquences $X_1 \neq X_2 \neq \ldots \neq X_n$ dans une séquence $AAX_1 \ldots X_nBB$), en ajoutant la contrainte d'homogénéité de classe (soit $X_1 = X_2 = \ldots = X_n$). Toutefois si la reclassification suit le même processus de classification, cette correction n'a pour seul effet que d'élire la classe majoritaire parmi les labels X_i. Or, si l'on considère que les labels sont erronés, le vote majoritaire prend le risque d'étendre l'erreur sur toutes les trames X_i.

Les règles heuristiques apparaissent de fait comme un pis aller dans un processus où le choix prématuré des labels entraîne une perte importante d'information pour le post-traitement. La seule information sur les classes non choisies pour une trame donnée provient de l'énumération de labels des trames voisines qui, du fait de la discrétisation des valeurs, se heurte aux limitations classiques du vote majoritaire. L'utilisation des probabilités a posteriori comme base du post-traitement permet ainsi de systématiser la prise de décision, en écartant les cas ambigus, et en prenant en compte la vraisemblance des classes non majoritaires. La figure 8.1 donne une illustration de ce phénomène sur un exemple de transition marginale : tandis que les seuls labels (en haut) ne permettent pas de fixer la frontière entre les classes, l'allure des probabilités a posteriori (en bas) nous montre l'évolution homogène croissante de la classe B, qui croise l'évolution décroissante de la classe A, plus bruitée, en particulier par de claires erreurs d'estimation sur la probabilité de la classe C aux trames 5 et 7.

Nous proposons dans la suite de ce chapitre plusieurs algorithmes de post-traitement sur les probabilités a posteriori estimées à partir des approches SVM multi-classes. On notera $\boldsymbol{p}_i = [p_c(i)]_{1 \leq c \leq C}$ le vecteur de probabilités a posteriori associé à la trame d'indice i, et calculé à partir du vecteur de descripteurs $\boldsymbol{x}_i$. La prise de décision se fera selon le principe de maximisation de la vraisemblance :

$$\hat{y}_i = \arg\max_{1 \leq c \leq C} \tilde{p}_c(i),$$

à partir des probabilités corrigées $\tilde{p}_c(i)$ obtenues par post-traitement des probabilités a posteriori $p_c(i)$.

8.2 Filtrages

La méthode de post-traitement la plus simple consiste à lisser les probabilités en appliquant un filtrage moyenneur sur une fenêtre dite glissante couvrant L trames successives. On choisit en général un nombre impair de trames afin de prendre en compte un nombre égal de trames *passées* et *futures*. Le filtrage prend la forme suivante :

$$\tilde{p}_c(i) = \frac{1}{L} \sum_{j=0}^{L-1} p_c(i + j - \frac{L-1}{2}) \qquad \forall c \in [1, \ldots, C].$$

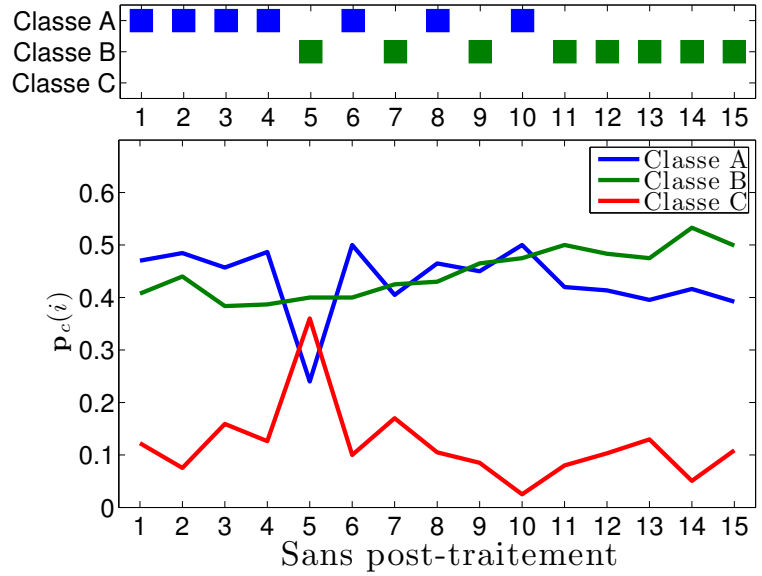


FIGURE 8.1 – Exemple de probabilités a posteriori sur une transition marginale entre les classes A et B. Les labels estimés sans post-traitement sont indiqués dans le cadre supérieur.

La fenêtre couvre donc les trames d'indice $i - \frac{L-1}{2}$ à $i + \frac{L-1}{2}$. Le choix de la longueur L est bien entendu déterminant et sera mené empiriquement par comparaison des performances sur un ensemble de validation. Les figures 8.2(a) et 8.2(b) comparent l'effet du filtre moyen sur l'exemple précédent pour des longueurs de fenêtre respectives de 3 et 7 trames.

On remarque que les probabilités résultantes ne respectent pas la contrainte stochastique $\sum_c \tilde{p}_c(i) = 1$, mais ce constat est de portée mineure puisqu'une normalisation éventuelle des probabilités n'a aucune influence sur la classe de probabilité maximale.

Cependant, le filtre moyenneur, bien qu'ayant l'avantage d'être linéaire et donc propice à une implémentation efficace, garde le défaut d'être relativement sensible aux brusques variations accidentelles. Ainsi on peut observer sur la figure 8.2(a) (concernant un filtre sur 3 trames) que les pics accidentels de la classe C sont amoindris mais restent visibles, bien que n'ayant aucun effet sur la décision finale ; en revanche, on voit que les variations accidentelles de la classe A restent suffisamment présentes pour maintenir une transition marginale.

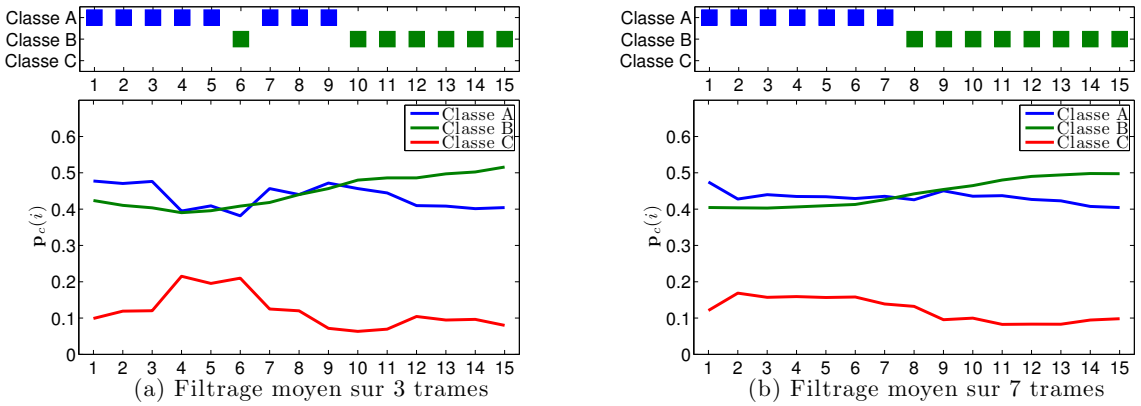
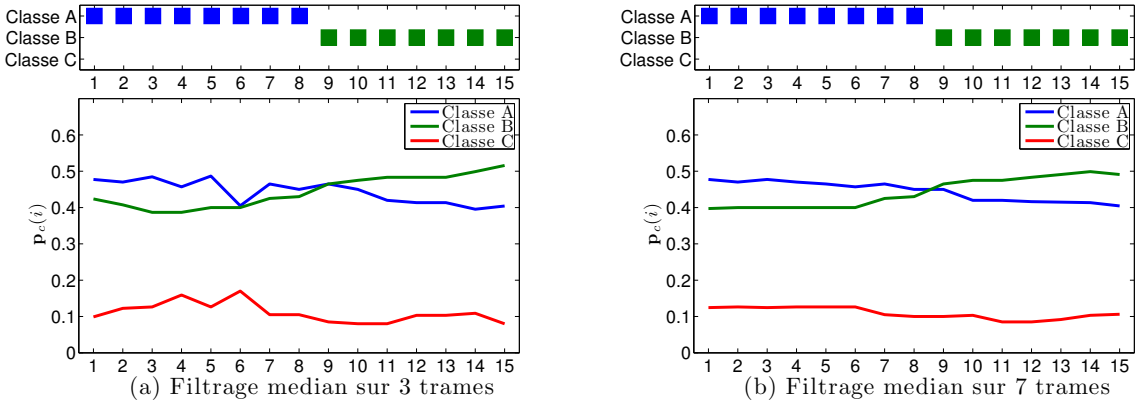


FIGURE 8.2 – Probabilités $\tilde{p}_c(i)$ après filtrage moyen sur des fenêtres glissantes de (a) 3 et (b) 7 trames. Les labels des classes majoritaires sont indiqués dans le cadre supérieur.

Le filtrage médian, généralement attribué à Gustav Fechner [122], est une alternative courante [2] pour corriger les défauts du filtre moyenneur. La médiane d'une distribution de probabilité est définie comme la valeur pour laquelle la fonction de densité de probabilité est égale à $\frac{1}{2}$. Sur un nombre impair d'exemples, on définit la valeur *médiane* comme l'exemple parmi ceux-ci qui sépare les autres en nombres égaux d'exemples inférieurs et supérieurs à ce dernier. Dans le cas d'un nombre pair d'exemples, on utilise généralement la valeur moyenne des deux exemples séparant les autres.

Il résulte que le résultat du filtrage médian n'est pas influencé par les valeurs aberrantes, si celles-ci sont suffisamment minoritaires. On peut d'ailleurs montrer que la valeur médiane est le point minimisant les déviations absolues des exemples. Les figures 8.3(a) et 8.3(b), illustrant l'effet du filtrage médian pour des fenêtres de 3 et 7 trames, montrent que ce dernier estime l'allure non bruitée des courbes de manière plus lisse et pour des fenêtres de taille moindre. On observe par exemple que le pic accidentel en trame 5 sur la classe A, que l'on retrouve après filtrage moyen sur 7 trames et qui implique ainsi une trame d'avance sur l'estimation de la transition, n'a pas cet effet après filtrage médian. En pratique on exploitera un filtre médian sur 9 trames longues (soit une fenêtre d'environ 5 secondes) ; cette envergure a été déterminée empiriquement.



(a) Filtrage median sur 3 trames (b) Filtrage median sur 7 trames

FIGURE 8.3 – Probabilités $\tilde{p}_c(i)$ après filtrage médian sur des fenêtres glissantes de (a) 3 et (b) 7 trames. Les labels des classes majoritaires sont indiqués dans le cadre supérieur.

8.3 Modèles de Markov Cachés (HMM)

Le lissage des probabilités par filtrage médian conduit généralement à une amélioration notable des performances. Néanmoins celui-ci est totalement aveugle au regard des classes considérées et ne peut donc prendre en considération la nature de classes impliquées dans une transition donnée. Pourtant, par exemple dans le contexte d'une émission de radio, la transition passagère de la musique pure vers la voix chantée sur fond musical est beaucoup plus vraisemblable que vers un segment de parole pure.

Les *Modèles de Markov* formalisent la modélisation des transitions entre un ensemble fini d'états. Nous proposons ici, après une brève description théorique de ce modèle stochastique et de ses applications courantes, des solutions de post-traitement des probabilités reposant sur ce dernier.

8.3.1 Description théorique

On modélise un processus par une suite d'états y_i évoluant dans un ensemble fini $S_1, \ldots, S_C$ (dans notre cas, l'état y_i représente la classe acoustique associée à la trame i, on parlera par la suite de *nombre d'instants* $j - i$ pour quantifier la durée qui sépare les états des trames i et j). Un processus Markovien consiste en une modélisation stochastique de la séquence d'états telle que la

2. que l'on trouve en particulier dans le domaine du traitement d'images.

probabilité d'être à un état S_c à l'instant i (soit $y_i = S_c$) ne dépend que de l'état occupé à l'instant précédant ($y_{i-1} = S_d$). De plus cette probabilité est indépendante de l'instant i. On peut donc définir les constantes a_{cd} suivantes :

$$a_{cd} = p(y_i = S_c | y_{i-1} = S_d) \quad 1 \leq c, d \leq C,$$

soumises aux contraintes stochastiques classiques, soit $a_{cd} \geq 0$ et $\sum_{c=1}^{C} a_{cd} = 1$. Si l'on introduit également les probabilités π_c de débuter la séquence sur l'état S_c (avec $\sum_{c=1}^{C} \pi_c = 1$), le modèle (a_{cd}, π_c) décrit entièrement le processus Markovien. On peut ainsi calculer la probabilité d'observer une séquence de n états $y_1, \ldots, y_n$:

$$p(y_1 = S_{c_1}, \ldots, y_n = S_{c_n}) = \pi_{c_1} \cdot \sum_{i=2}^{n} a_{c_i c_{i-1}}.$$

Le *Modèle de Markov Caché* (HMM *Hidden Markov Model*) [190] [3] substitue à la connaissance des états (qui sont désormais non observables, donc *cachés*), celle d'observations O_i dont la génération à chaque instant i est gouvernée par la *probabilité d'observation* b_c qui ne dépend que de l'état $y_i = S_c$. Soit un ensemble fini de M symboles d'observations $v_1, \ldots, v_M$, on a :

$$b_c(k) = p(O = v_k \,|\, y = S_c) \qquad 1 \leq c \leq C,\ 1 \leq k \leq M,$$

un processus ne sera plus caractérisé par sa séquence d'état $y_1, \ldots, y_n$ mais par la séquence d'observations $\boldsymbol{O} = O_1, \ldots, O_n$ avec $O_i = v_{k_i} \forall i$, ce qui suppose l'existence d'une séquence d'états inconnus ayant généré ces observations.

Pour résumer, un HMM est caractérisé par les paramètres suivants :

- Ses C **états** S_c, pour $1 \leq c \leq C$.
- L'alphabet des M **symboles** d'observations v_k, pour $1 \leq k \leq M$.
- Les C^2 **probabilités de transition** a_{cd}, synthétisées par la matrice $\boldsymbol{A} = [a_{cd}]_{cd}$.
- Les $M \times C$ **probabilités d'observation** $b_c(k)$, formant la matrice $\boldsymbol{B} = [b_c(k)]_{ck}$.
- Enfin, les **probabilités d'état initial** π_c, formant le vecteur $\boldsymbol{\pi} = [\pi_c]_c$.

L'ensemble des paramètres $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$ constitue le modèle HMM.

8.3.2 Estimation de la séquence d'états

Rabiner [190], reprenant Ferguson, pose les trois problèmes traduisant le champ d'applications concrètes des HMM sur la base d'une séquence d'observations $\boldsymbol{O}$:

1. Connaissant le modèle λ, comment évaluer la probabilité d'observation $p(\boldsymbol{O} \,|\, \lambda)$?
2. Connaissant le modèle λ, comment déterminer la séquence d'états $\boldsymbol{Y} = y_1, \ldots, y_n$ la plus susceptible d'avoir généré les observations $\boldsymbol{O}$?
3. Comment paramétrer le modèle λ maximisant la probabilité $p(\boldsymbol{O} \,|\, \lambda)$?

Notre but étant de déterminer les classes acoustiques associées aux différentes trames, donc la séquence d'états $\boldsymbol{Y}$, c'est bien sûr le problème 2 qui nous intéresse ici.

Ce dernier se résout en employant l'*algorithme de Viterbi* [78][238] qui consiste à rechercher de manière inductive la séquence maximisant la probabilité d'observer les i premières observations. Ainsi si l'on pose :

$$\begin{aligned} \delta_1(c) &= \pi_c \, b_c(O_1) & 1 \leq c \leq C, \\ \delta_i(c) &= \max_{1 \leq d \leq C} [\delta_{i-1}(d) \, a_{cd}] \, b_c(O_i), & \end{aligned} \tag{8.1}$$

alors $\delta_i(c)$ mesure bien la probabilité maximale parmi toutes les séquences d'états d'observer la séquence $O_1, \ldots, O_i$. On calcule ainsi les probabilités $\delta_i(c)$ jusqu'à la trame n. Les valeurs $\psi_i(c)$

3. Si le désormais célèbre tutoriel de Rabiner est la référence la plus courante sur le sujet, les HMM sont originellement proposés par Baum et d'autres auteurs dans une série d'articles datant des années 60 et référencés dans ce même tutoriel.

suivantes sont renseignées conjointement afin de conserver la trace du parcours maximisant la vraisemblance :

$$\begin{aligned} \psi_1(c) &= 0 \qquad\qquad\qquad\qquad \text{par convention} \\ \psi_i(c) &= \underset{1\leq d\leq C}{\arg\max} \left[\delta_{i-1}(d)\, a_{cd}\right]. \end{aligned} \tag{8.2}$$

On déduit ensuite les états $\hat{y}_1, \ldots, \hat{y}_n$ du chemin optimal par induction arrière :

$$\begin{aligned} \hat{y}_n &= \underset{1\leq c\leq C}{\arg\max}\, \delta_n(c) \\ \hat{y}_{i-1} &= \psi_i\left(\hat{y}_i\right). \end{aligned}$$

Un avantage de l'algorithme de Viterbi est qu'il ne dépend que des observations présente et passées. On peut ainsi appliquer ce dernier en temps-réel pour calculer à chaque instant i les valeurs $\delta_i(c)$ en fonction des valeurs passées. Cependant, la séquence d'états optimale étant évaluée par induction arrière, une prise de décision en temps réel impliquerait que les états soient déterminés indépendamment. On peut compenser ce phénomène en introduisant un léger retard de décision, de manière à ne déterminer l'état à un instant i qu'une fois que l'on dispose de suffisamment de valeurs $\delta_i(c)$ d'avance.

Il nous reste à déterminer quelles sont les observations $\boldsymbol{O}$ et comment fixer les paramètres du modèle $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi})$.

8.3.2.1 Approche classique par mélange de gaussiennes

En règle générale, les HMM sont utilisés en exploitant les vecteurs de descripteurs $\boldsymbol{x}_i$ comme observations du processus. C'est l'application que l'on trouve dans la plupart des articles de la littérature exploitant les HMM pour le problème de classification [125][195][9]. L'exploitation de données réelles multi-dimensionnelles (et non tirées dans un alphabet fini de symbole) suppose cependant une adaptation de l'algorithme puisqu'il n'est pas possible de définir la matrice $\boldsymbol{B}$ des probabilités d'observations.

Celles-ci sont classiquement modélisées par l'estimation des densités de probabilités, notamment par un Modèle de Mélange de Gaussiennes (GMM, *Gaussian Mixture Model*) :

$$b_c(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_{cm}\, \mathcal{N}\left(\boldsymbol{x}, \boldsymbol{\mu}_{cm}, \boldsymbol{\Sigma}_{cm}\right) \quad 1 \leq c \leq C,$$

où $\mathcal{N}\left(\boldsymbol{x}, \boldsymbol{\mu}_{cm}, \boldsymbol{\Sigma}_{cm}\right)$ est la loi normale multidimensionnelle de centre $\boldsymbol{\mu}_{cm}$ et de matrice de covariance $\boldsymbol{\Sigma}_{cm}$, et les α_{cm} sont les coefficients positifs de pondération respectant les contraintes stochastiques $\sum_{m=1}^{M} \alpha_{cm} = 1$. Les paramètres du modèle de mélange sont estimés au moyen de l'algorithme EM [62] (*Expectation Maximization*).

On voit que l'extension des données d'observation au domaine continu ne change en rien l'algorithme de Viterbi (équation 8.1). Le choix du nombre de gaussiennes M est bien sûr important mais ne sera pas traité ici.

8.3.2.2 Proposition de post-traitement par HMM

On trouve de nombreux exemples dans la littérature [92][220], généralement dans le domaine de la reconnaissance de la parole, d'approches hybrides substituant les SVM aux GMM pour l'évaluation des probabilités $b_c(\boldsymbol{x})$. En déterminant les probabilités a posteriori $p(y = S_c \,|\, \boldsymbol{x})$ avec les SVM, on déduit les probabilités d'observation avec la règle de Bayes :

$$p(\boldsymbol{x} \,|\, y = S_c) = \frac{p(y = S_c \,|\, \boldsymbol{x}) \cdot p(\boldsymbol{x})}{p(y = S_c)},$$

où l'on peut supposer $p(\boldsymbol{x})$ uniforme ; les probabilités a priori $p(y = S_c)$ peuvent également être choisies uniformes ou déterminées à partir de la base d'apprentissage.

Nous proposons une autre approche où l'application du post-traitement par HMM est indépendante de la nature du classifieur. On exploite, comme observations, non plus les descripteurs mais les probabilités a posteriori déduites de la classification par SVM ou, dans un cas général, de tout autre processus de classification.

On modélise ainsi, de manière similaire à l'approche classique, la densité de probabilité des observations $b_c(\boldsymbol{x})$ par un mélange de M gaussiennes. Toutefois les probabilités $b_c(\boldsymbol{x})$, respectant la contrainte stochastique de somme unitaire, sont fortement corrélées et situées sur l'hyperplan $x_C = 1 - \sum_{i=1}^{C-1} x_c$, ce qui induit de fortes singularités dans la modélisation gaussienne. On réduit donc le vecteur d'observations d'une dimension en excluant la dernière composante, soit :

$$\boldsymbol{O}_i = [p(y = S_1|\boldsymbol{x}_i), \ldots, p(y = S_{C-1}|\boldsymbol{x}_i)] \quad \forall i.$$

Les probabilités a posteriori étant à priori fortement corrélées aux états du modèle, la modélisation peut se contenter d'un nombre assez limité de gaussiennes, voire d'une seule.

L'apprentissage du modèle gaussien nécessite l'estimation des probabilités a posteriori sur un ensemble de validation disjoint de l'ensemble d'apprentissage afin de prendre en compte le biais du système de classification dans le post-traitement.

8.3.3 Estimation du modèle λ

Contrairement aux applications en reconnaissance de la parole, où un modèle est appris pour chaque mot sur des états inconnus, ici nous avons l'avantage de connaître lors de l'apprentissage les états relatifs aux séquences, par le biais des annotations du corpus. Ainsi, si le corpus est constitué de K fichiers audio, où à chaque fichier d'indice k, de n_k trames, est associée la séquence des classes $y_1^k, \ldots, y_{n_k}^k$, on estime empiriquement à partir du corpus d'apprentissage les probabilités de transition a_{cd} et les probabilités d'état initial π_c (pour $1 \leq c, d \leq C$) :

$$\begin{aligned} a_{cd} &= \frac{1}{\sum_{k=1}^{K} n_k} \sum_{k=1}^{K} \mathrm{Card}\left\{y_i^k = c,\, y_{i-1}^k = d,\, 2 \leq i \leq n_k\right\} \\ \pi_c &= \frac{1}{K} \sum_{k=1}^{K} \mathrm{Card}\left\{y_1^k = c\right\}. \end{aligned}$$

On remarque que ces grandeurs ne dépendent que des annotations et que le processus de classification n'intervient aucunement. On peut donc utiliser la totalité du corpus d'apprentissage, pour accroître la confiance, sans introduire de biais dans l'estimation du modèle HMM.

8.4 Hidden Semi-Markov Models

Les HMM ont montré leur efficacité pour de nombreuses applications. Néanmoins le modèle implique que la probabilité de rester sur un état c durant d instants successifs, suit une distribution géométrique : $p(d) = a_{cc}^{d-1}(1 - a_{cc})$. Cette contrainte ne traduit pas nécessairement le modèle d'une application donnée. Dans notre cas, par exemple, sur des archives radiophoniques, les segments de parole ne suivent en aucun cas une loi géométrique et peuvent d'ailleurs atteindre des longueurs très conséquentes. La figure 8.4 illustre ce constat en représentant la distribution des nombres de trames consécutives pour les classes de musique (à gauche) et de parole (à droite) sur le corpus ESTER (que nous présenterons dans la section 10.1.1).

On constate que si les durées des segments de musique suivent approximativement une distribution géométrique, on trouve au contraire de nombreux segments longs de parole, qui rendent la modélisation géométrique inadéquate.

Les *modèles semi-markoviens cachés* [165] (HSMM, *Hidden Semi-Markov Model*), également appelés *modèles de segments* [174], sont généralement attribués à Ferguson qui introduit dès 81 la modélisation des durées d'états [75]. Ils étendent le modèle HMM pour relâcher la contrainte précédente, en associant à chaque état y_i, non plus une observation mais une séquence d'observations. Ainsi on ajoute au système une variable d'état supplémentaire ℓ_i associant à chaque état y_i le nombre d'observations $O_{i,1}, \ldots, O_{i,\ell_i}$ générées par le système à cet état ; on parle ainsi de

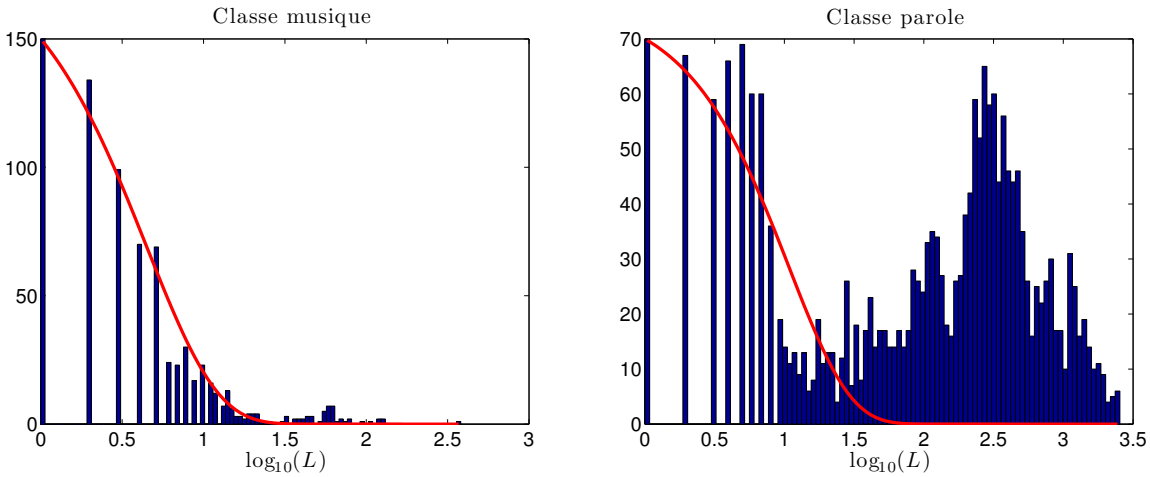


FIGURE 8.4 – Distribution des nombres de trames par segments (L) pour les classes de musique (à gauche) et de parole (à droite) sur le corpus ESTER. Les abscisses suivent une échelle logarithmique.

segments homogènes modélisés par la séquence des états. De plus, le système est caractérisé par deux nouvelles probabilités : $p(\ell = L \,|\, y = S_c)$ et $p(O_1, \ldots, O_\ell \,|\, y = S_c, \ell = L)$ gouvernant respectivement la distribution des longueurs de segments pour un état donné S_c et la probabilité d'observer la séquence d'observations $O_1, \ldots, O_L$ à l'état S_c si celle-ci est de longueur $\ell = L$.

L'ajout du paramètre de durée des segments complexifie considérablement le modèle, en premier lieu parce que la variabilité de la longueur des segments introduit un nouveau paramètre dans l'espace de recherche, qui implique un facteur multiplicatif D sur la complexité [254][256] (où D est la longueur maximale d'un segment) pour l'algorithme Forward-Backward [4], lequel n'est pas présenté ici mais apporte une réponse aux deux autres questions de la section 8.3.2. De plus le champ des probabilités d'observation se trouve largement élargi puisqu'il couvre la modélisation de séquences d'observations de longueurs variables. Ostendorf et al. [174] proposent un large panorama de modèles dynamiques possibles traduisant les caractéristiques de séquences d'observation. Toutefois, nous nous limiterons au cas d'observations indépendantes, soit :

$$p(O_1, \ldots, O_\ell \,|\, y_i = S_c, \ell = L) = \prod_{l=1}^{L} p(O_l \,|\, y_i = S_c),$$

où l'on exploitera les probabilités d'observation $p(O_l \,|\, y_i = S_c)$ définies pour le modèle HMM. Il est important de noter que l'hypothèse d'indépendance des observations est loin d'être arbitraire si l'on considère le fait que les observations proviennent du processus de classification par SVM, où chaque vecteur descripteur de trame est traité indépendamment des autres.

De plus, plutôt que de considérer que chaque état produit plusieurs observations, on introduit une variable d'état f_i qui décrit le nombre d'instants successifs passés sur l'état actuel à un instant i, ce qui nous permet de revenir au formalisme des HMM avec seulement une légère modification dans l'algorithme de Viterbi. On parlera par la suite de *segments* pour désigner une suite d'états de même classe, distincte des classes des segments adjacents.

On adapte ce dernier en court-circuitant les probabilités de transition a_{cc} d'un état à lui-même, desquelles résulte, comme nous l'avons expliqué, la distribution géométrique de la probabilité de rester dans un état c. Ainsi la probabilité de transition de l'état c à lui-même dépend, dans le cas des HSMM, de la probabilité $p(\ell = L \,|\, y = S_c)$ des durées de segments générés par l'état S_c, introduite précédemment. On note $e_c(L)$ la probabilité qu'un segment soit de longueur supérieure ou égale à L, appelée *probabilité de stagnation* :

$$e_c(L) = p(\ell \geq L \,|\, y = S_c) = \sum_{\ell=L}^{\infty} p(\ell = L \,|\, y = S_c),$$

4. également désigné sous le nom d'algorithme de Baum-Welch.

en supposant que l'on se trouve sur l'état c depuis $f_i = F$ instants, la probabilité d'y rester un instant de plus est donc égale à :

$$\begin{aligned} p(y_{i+1} = S_c \,|\, y_i = S_c,\, f_i = F) &= p(\ell \geq F+1 \,|\, \ell \geq F,\, y = S_c) && (8.3)\\ &= \frac{p(\ell \geq F+1,\, \ell \geq F,\, y = S_c)}{p(\ell \geq F,\, y = S_c)} && (8.4)\\ &= \frac{e_c(F+1)}{e_c(F)}. && (8.5) \end{aligned}$$

ce qui revient donc à substituer à la constante a_{cc} la valeur $\frac{e_c(F+1)}{e_c(F)}$, dépendant du nombre d'instants F déjà passés sur l'état actuel, à l'instant i. Si l'on reprend l'expression de $\delta_i(c)$ (équation 8.1), on propose ainsi la règle d'induction suivante :

$$\delta_i(c) = \max_{1 \leq d \leq C} [\delta_{i-1}(d)\, \tilde{a}_{cd}]\, b_c(O_i),$$

avec :

$$\tilde{a}_{cd} = \begin{cases} a_{cd} & \text{si } c \neq d \\ \frac{e_c(f_i(c)+1)}{e_c(f_i(c))} & \text{si } c = d \end{cases}$$

où l'on introduit la variable $f_i(c)$, gardant la trace du nombre d'instants depuis le dernier changement d'état, également renseignée par induction durant l'algorithme :

$$\begin{aligned} f_1(c) &= 1 \\ f_i(c) &= \begin{cases} 1 & \text{si } \psi_i(c) \neq c \\ f_{i-1}(c) + 1 & \text{sinon.} \end{cases} \end{aligned}$$

Les probabilités de stagnation $e_c(\ell)$, que nous avons introduites dans le modèle HSMM, sont également estimées empiriquement à partir des annotations du corpus d'apprentissage. On regroupe dans un premier temps la séquence des classes par trames $y_1^k, \ldots, y_{n_k}^k$ du fichier k en une séquence de S_k segments de longueur $l_1^k, \ldots, l_{S_k}^k$ et de classe homogène $c_1^k, \ldots, c_{S_k}^k$ (avec $c_i^k \neq c_{i+1}^k$). On estime à partir des segments, les probabilités de stagnation :

$$e_c(\ell) = \frac{1}{l_{T,c}} \sum_{k=1}^{K} \text{Card}\left\{c_s^k = c, l_s^k \geq \ell\right\}_{1 \leq s \leq S_k},$$

où $l_{T,c}$ est un facteur de normalisation égal au nombre total de sous-séquences de segments partant de débuts de segments de classe c, sur l'ensemble du corpus :

$$l_{T,c} = \sum_{k=1}^{K} \sum_{s | c_s^k = c} \sum_{l=1}^{l_s^k} l.$$

Ce dernier garantit le respect de la contrainte $e_c(1) = 1\ \forall c$.

Même sur un corpus conséquent, le nombre de segments longs n'est jamais très élevé, si bien que généralement les grandeurs $e_c(\ell)$ ont une allure en escalier pour de grandes valeurs de ℓ. On résout ce problème par un lissage classique ou par une simple interpolation affine entre les points de changement. On peut également chercher à modéliser statistiquement $e_c(\ell)$, mais un modèle trop pauvre peut se révéler équivalent au formalisme des HMM (c'est-à-dire à une modélisation par une loi exponentielle).

On a donc proposé un algorithme simple, sans coût additionnel prohibitif par rapport aux HMM, pour prendre en compte explicitement les durées des segments de classe homogène dans l'algorithme de Viterbi. Cependant, les résultats sont très décevants en pratique puisque l'application du post-traitement par HSMM n'apporte aucun gain en performances notable par rapport au modèle HMM. Ce résultat semble montrer que l'inadéquation théorique des HMM aux segments longs ne se traduit pas par un réel handicap en pratique. Nous n'inclurons donc pas les HSMM dans le chapitre d'évaluation, puisque les résultats sont globalement les mêmes qu'avec les HMM.