Estimation de similarité entre séquences de descripteurs à l'aide de machines à vecteurs supports

Romain Tavenard*, Laurent Amsaleg** et Guillaume Gravier**
*IRISA/ENS de Cachan, **IRISA/CNRS
Campus de Beaulieu
F - 35 042 Rennes Cedex
{rtavenar, lamsaleg, ggravier}@irisa.fr

Résumé

Les bases de données contenant des séquences multimédia se trouvent maintenant partout. Archives de l'INA, balladodiffusion (podcast)*, production et partage de vidéo font désormais partie de notre quotidien. Faire des recherches par le contenu dans ces bases où l'information est en flux est difficile, notamment à grande échelle. Une des questions fondamentales concerne la mesure de similarité entre la séquence requête et celles de la base. Nous proposons dans cet article d'utiliser des SVM pour modéliser chaque séquence et de comparer les modèles pour établir la similarité. Nous comparons cette approche à celle habituelle où l'on utilise l'alignement dynamique de séquences* (DTW)*. Les résultats obtenus sur des données audio réelles montrent l'utilité d'une approche par modèles.*

Mots clefs

Séquences de descripteurs multidimensionnels, machines à vecteurs support, aspects temporels.

1 Motivations

Notre quotidien se nourrit désormais de bases de données contenant des séquences multimédia : accès à des fonds documentaires, production et partage de vidéo ou encore balladodiffusion *(podcast)*. Cependant, il existe actuellement assez peu de techniques permettant de faire des recherches efficaces par le contenu (ou encore des recherches par similarité) dans ces documents. La nécessité de disposer de ces techniques se fait particulièrement sentir dans le domaine de l'archivage multimédia où l'on veut annoter les flux en prévision de recherches documentaires ultérieures. Cela implique de segmenter ou encore de structurer un flux vidéo (télévisé) ou audio (radio) en programmes distincts pour ensuite les annoter. Cette tâche nécessite des recherches dans les flux pour y repérer, par exemple, les bandes annonce, les ruptures, le retour régulier de séquences particulières, ou pour y repérer les interventions de telle ou telle personnalité, etc.

Actuellement, aucune technique de recherche n'est encore suffisamment efficace pour permettre d'exploiter des archives sonores ou audio-visuelles de taille réelle (dizaines voire centaines de milliers d'heures). Une des principales difficultés vient du caractère temporel de ces flux. Décrire de l'audio et de la vidéo revient en effet à fabriquer des séquences de descripteurs multidimensionnels dont il est important de préserver l'ordre et l'enchaînement. Aussi, on ne peut simplement étendre les approches d'indexation multidimensionnelle traditionnelles qui gèrent des descriptions d'où est généralement absente la notion de séquence (voir ty-

piquement [2, 6, 8] pour des images fixes).

Cet article traite des manières de comparer deux séquences de descripteurs issus de documents multimédias en flux. Il s'agit ici de définir comment juger de la similarité entre une séquence requête et une base de séquences de référence. Nous explorons deux pistes différentes. La première, habituellement suivie dans les travaux apparentés, mène à évaluer la similarité entre deux séquences en comparant directement les descripteurs de ces deux séquences, généralement au travers d'un calcul de distance par alignement dynamique, ou *DTW – Dynamic Time Warping* [9]. La seconde se place à un niveau plus élevé et vise à modéliser chaque séquence pour ensuite comparer les modèles entre eux et établir la similarité. La modélisation que nous proposons dans cet article se base sur l'utilisation de machines à vecteurs supports, ou *SVM – Support Vector Machines* [10].

Le but principal de cette étude est d'analyser la capacité qu'a chaque approche d'absorber les distorsions sur l'axe du temps. Il est connu que les DTW sont incapables de comparer des séquences non recalées. Il est alors difficile de comparer deux séquences ayant seulement une partie en commun. La segmentation préalable en séquences est ici cruciale car les DTW ne tolèrent aucune distorsion temporelle sur les instants de début et de fin des séquences. À l'inverse, il est connu que les DTW tolèrent aisément les distorsions temporelles durant les séquences (recalées).

La question fondamentale que pose cet article est donc d'observer le comportement des SVM face à ces mêmes problèmes. Aussi, les contributions de cet article portent d'une part sur l'utilisation des SVM pour modéliser des séquences de descripteurs et, d'autre part, sur une mise en œuvre sur des données réelles mettant en lumière les avantages et inconvénients de la modélisation par rapport à l'utilisation directe des descriptions face à des distorsions sur l'axe du temps.

Cet article est structuré comme suit. Les sections 2 et 3 présentent chacune les principes des DTW et des SVM, respectivement. La section 4 décrit comment modéliser des séquences avec des SVM. La section 5 analyse les points communs et les différences entre les deux approches. La section 6 décrit le dispositif expérimental et les évaluations. La section 7 conclut et expose quelques pistes de recherche.

2 DTW – Dynamic Time Warping

L'algorithme d'alignement temporel dynamique (DTW), introduit par [9], est couramment utilisé dans les bases de données pour comparer des séquences temporelles [5, 11]. La DTW permet de comparer des séquences de longueurs différentes mais recalées, c'est-à-dire dont les instants de début et de fin coïncident, résolvant ainsi les problèmes de variation d'échelle et de distorsion temporelle. Le principe est que l'on s'autorise à étirer localement chacune des séquences pour compenser ces distorsions.

Comparer deux séquences par DTW revient à chercher le chemin de coût minimal dans la matrice des distances entre les éléments des séquences à comparer. Pour des séquences recalées, on cherche un appariement optimal en ne considérant que les chemins reliant le début des deux séquences à leur fin. La DTW se basant sur les distances entre les éléments de la matrice, elle est très sensible aux valeurs aberrantes et ne respecte pas l'inégalité triangulaire..

2.1 Principe de l'alignement dynamique

Soient $Q = q_1, q_2, \ldots, q_n$ et $C = c_1, c_2, \ldots, c_m$ deux séries d'observations dont on souhaite connaître la similarité. On commence par construire une matrice S de correspondance entre les deux séquences, de taille $n \times m$, telle que :

$$\forall (i,j) \in [1,n] \times [1,m], s_{i,j} = d(q_i, c_j)$$

On peut alors trouver dans cette matrice un chemin, noté W de longueur K, tel que $w_1 = (1,1)$

et $w_K = (n, m)$, correspondant à un minimum de distance :

$$DTW(Q, C) = \min_W \left(\sum_{i=1}^{K} S(w_i) \right)$$

Le chemin W considéré doit vérifier les propriétés suivantes :

- **Continuité** : Si $w_k = (a, b)$ et $w_{k-1} = (a', b')$, alors $a - a' \leq 1$ et $b - b' \leq 1$
- **Monotonie** : Si $w_k = (a, b)$ et $w_{k-1} = (a', b')$, alors $a - a' \geq 0$ et $b - b' \geq 0$

Ce chemin peut être trouvé en utilisant la programmation dynamique pour évaluer le terme $\gamma_{i,j}$ défini comme suit :

$$\gamma_{i,j} = d(q_i, c_j) + min\{\gamma_{i-1,j-1}, \gamma_{i-1,j}, \gamma_{i,j-1}\}$$

On peut déterminer la distance DTW entre deux séquences par l'algorithme 1.

Algorithme 1 Calcul de la distance DTW.

```
ENTRÉES: q[1..n], c[1..m]
  int D[0..n][0..m]
  int i, j, cost
  D[0][1..m] ← ∞
  D[1..n][0] ← ∞
  D[0][0] ← 0
  pour i = 1 à n faire
    pour j = 1 à m faire
      D[i][j] ← min(D[i-1][j], D[i][j-1], D[i-1][j-1]) + dist(q[i], c[j])
    fin pour
  fin pour
  Retourner D[n][m]
```

2.2 Problème du recalage des séquences

On remarque que cette implémentation du *dynamic time warping* présente un inconvénient majeur : puisque l'on fixe $w_1 = (1, 1)$ et $w_K = (n, m)$, les séquences considérées doivent être recalées avant d'être comparées. Pour cela, il faut trouver des points d'ancrage correspondant à des sous-parties des séquences considérées fortement similaires [1], ce qui est très coûteux en temps de calcul. On peut contourner ce problème autrement en modifiant légèrement l'algorithme ci-dessus de telle manière que l'on relâche les contraintes aux bords : $w_1 = (i_1, 1), i_1 \in [1, n]$ et $w_K = (i_2, m), i_2 \in [1, n]$. Pour cela, il suffit de fixer :

$$\text{D}[i_1][0] = 0, \forall i_1 \in [1, n]$$

et de considérer une distance cumulée $\gamma_{i,j}$ normalisée par la longueur du meilleur appariement se terminant au point (i, j). La distance retournée est alors définie par

$$DTW(Q, C) = \min_{i_2 \in [1,n]} \frac{\text{D}[i_2][\text{m}]}{l(i_2)}$$

où $l(i_2)$ est la longueur du chemin menant à (i_2, m).

Notons que dans ce cas la DTW n'est pas symétrique.

3 SVM – Support Vector Machines

Les SVM [10] sont très utilisées dans le domaine de la classification. L'idée maîtresse se base sur l'utilisation de fonction noyau *(kernel functions)* permettant de séparer de manière optimale des points en deux classes. La séparation se fait à partir d'un jeu de données d'apprentissage permettant de trouver un (hyper)plan séparant les points.

Elles peuvent également être utilisées dans des problèmes de régression, lesquels consistent à approximer une fonction inconnue $g : \mathbb{R}^D \rightarrow \mathbb{R}$ à partir des observations $\{\mathbf{x_i}, y_i\}_{i=1}^N$ telles que $y_i = g(\mathbf{x_i}) + \eta$, η étant du bruit.

Tant pour la classification que pour la régression, on estime g par une fonction de la forme :

$$\hat{g}(\mathbf{x}) = \sum_{i=1}^{L} c_i \phi_i(\mathbf{x}) + b$$

où $\{\phi_i\}_{i=1}^{L}$ est un ensemble de fonctions de base et où les coefficients c_i et b sont à optimiser. Pour cela, on cherche à minimiser la fonctionnelle :

$$R(\hat{g}) = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{g}(\mathbf{x_i})|_\epsilon + \lambda\|c\|^2$$

où l'on définit la fonction d'erreur :

$$|x|_\epsilon = \begin{cases} \epsilon & \text{si } x < \epsilon \\ x & \text{sinon} \end{cases}$$

[10] a montré que la fonction minimisant cette fonctionnelle peut s'écrire sous la forme :

$$\hat{g}(\mathbf{x}, \alpha, \alpha^*) = \sum_{i=1}^{L} (\alpha_i^* - \alpha_i) K(\mathbf{x}, \mathbf{x_i}) + b$$

sous les conditions :

$$\begin{aligned} \forall i = 1, \ldots, N \qquad & \alpha_i^* \alpha_i = 0 \\ & \alpha_i \geq 0 \\ & \alpha_i^* \geq 0 \\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^D \quad & K(\mathbf{x}, \mathbf{y}) = \textstyle\sum_{i=1}^{L} \phi_i(\mathbf{x})\phi_i(\mathbf{y}) \end{aligned}$$

Seuls quelques termes de la somme sont non nuls, les $\mathbf{x_i}$ correspondants étant alors appelés vecteurs supports. Plus les vecteurs supports sont nombreux, meilleure est l'approximation. Ceci se contrôle au travers d'un paramètre ϵ. La figure 1 donne un exemple de modélisation d'une fonction sinus cardinal. Cette fonction est représentée sur la figure par les points de données fins. Les vecteurs supports déterminés par le modèle sont choisis parmi les points de la courbe et sont représentés sur la figure par des points épais. On remarque bien l'absence de vecteurs supports là où la fonction est facile à interpoler. La ligne continue est la fonction g estimée à partir de ces vecteurs. La figure montre bien que g est une approximation de la fonction originale. De plus, cette figure met en avant le lien entre la valeur de ϵ, la fidélité de la fonction g résultante et le nombre de vecteurs supports retenus.

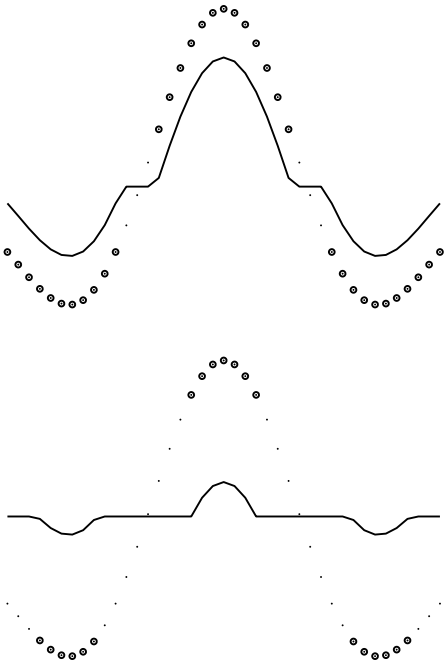

FIG. 1 – Approximation d'un sinus cardinal. En haut, $\epsilon = 0,2$. En bas, $\epsilon = 0,5$.

On remarque par ailleurs qu'il n'est pas nécessaire de connaître les ϕ_i pour construire la fonction voulue, mais seulement un noyau K associé, qui doit être un produit scalaire sur un certain espace à définir. Il suffit donc de connaître une caractérisation des fonctions K acceptables (cf. [10]), ce que nous fournit le théorème de Mercer [7], à partir duquel on peut construire de nombreuses fonctions dont les plus utilisées sont résumées en tableau 1. Nous évaluons l'intérêt de ces quatre noyaux en section 6.

4 Modéliser des séquences par SVM

Cette section décrit comment il est possible de modéliser l'évolution temporelle de descripteurs multidimensionnels avec des SVM dans le but de juger de la similarité de séquences. L'aspect ré-

Dénomination	Expression de $K(x, y)$
Produit scalaire usuel	$(x \cdot y)$
Fonction polynomiale	$[(x \cdot y) + 1]^d$
Fonction à base radiale	$\exp[-\gamma \|x - y\|^2]$
Réseau de neurones à 2 niveaux	$\tanh[(x \cdot y)v + c]$

TAB. 1 – Noyaux usuels pour les SVM.

gression est utilisé ici puisque l'on veut obtenir une approximation de la fonction décrivant la manière dont les descripteurs évoluent durant la séquence.

Il n'est pas nécessaire d'estimer cette fonction dans l'espace multidimensionnel. En effet, en supposant les d dimensions des descripteurs décorrélées, la modélisation de l'évolution des valeurs des descripteurs selon une dimension particulière ne nécessite pas de prendre en compte les autres dimensions. On utilisera alors d SVM pour modéliser l'évolution des d dimensions de la séquence décrite.

Pour chaque $i \in [1, d]$, on construit donc un modèle qui met en relation, pour chaque temps $t \in [1, M]$ (où M est la longueur de la séquence considérée), $x_{i,t}$ avec ses voisins $\{x_{i,\tilde{t}}\}_{\tilde{t} \in v_t}$ où $v_t = \{t - \frac{H}{2}, ..., t - \frac{H}{2}\}$ est le voisinage temporel considéré. On notera H la taille de ce voisinage que l'on choisira centré. Les fonctions $\{g_i\}_{i \in [1,d]}$ recherchées seront donc telles que $x_{i,t} = g_i(\{x_{i,\tilde{t}}\}_{\tilde{t} \in v_t}) + \eta$.

L'approche SVM sur chaque dimension ne conservera qu'un certain nombre de points jugés caractéristiques qui sont les vecteurs supports. Leur nombre dépend directement de la précision demandée au modèle, fixée à l'apprentissage par l'intermédiaire d'ϵ.

Dans l'approche par SVM, chaque séquence de descripteurs est ainsi représentée par une fonction de régression composée de d SVM ayant les mêmes noyaux. Comparer deux séquences revient ici à juger de la similarité de leurs modèles. Malheureusement, on ne dispose pas, en général, d'une mesure de similarité paramétrique entre modèles. Cette mesure n'existe que si l'on choisit un noyau linéaire. Dans les autres cas, on doit donc avoir recours à une mesure de vraisemblance croisée [3].

Autrement dit, si $G = \{g_i\}_{i \in [1,d]}$ est la fonction de régression estimée sur la séquence de descripteurs $\{\mathbf{X}_t = \{x_{1,t}, \cdots, x_{d,t}\}\}_{t=0}^{N}$ associée à la séquence S_1 et que $\{\mathbf{Y}_t\}_{t=0}^{M}$ est la séquence de descripteurs associée à S_2, nous construisons une nouvelle séquence $\{\tilde{\mathbf{Y}}_t\}_{t=0}^{M}$ telle que, pour tout $t \in [1, M]$:

$$\tilde{\mathbf{Y}}_t = G(\mathbf{Y}_{t-\frac{H}{2}}, \ldots, \mathbf{Y}_{t-1}, \mathbf{Y}_{t+1}, \ldots, \mathbf{Y}_{t+\frac{H}{2}})$$

Ainsi, $\{\tilde{\mathbf{Y}}_t\}_{t=0}^{M}$ est la séquence des valeurs prédites par le modèle issu de S_1. On considère que deux séquences sont similaires si le modèle issu de l'une prédit l'autre de manière fiable. Pour mesurer cette fiabilité, on construit la mesure de similarité suivante entre les séquences S_1 et S_2 :

$$d(S_1, S_2) = \sqrt{\frac{1}{N} \sum_{i=1}^{D} \sum_{t=1}^{N} (\tilde{y}_{i,t} - y_{i,t})^2}$$

5 Analyse des deux approches

Nous présentons dans cette partie une analyse des principes des deux approches afin de mettre en avant leurs points communs et leurs différences.

5.1 Mesure de similarité

La DTW que nous utilisons et les approches SVM ont en commun de se baser sur des distances non symétriques. Cela signifie qu'à partir des séquences S_1 et S_2 on peut construire les mesures de similarités $D_1(S_1, S_2) = d(S_1, S_2)$ et $D_2(S_1, S_2) = d(S_2, S_1)$, où $d(.,.)$ est la mesure de similarité considérée.

Pour la DTW, $D_1(S_1, S_2)$ correspond à aligner S_1 avec un sous-ensemble de S_2. Pour les SVM, $D_1(S_1, S_2)$ sera l'erreur obtenue en cherchant à prédire S_1 à partir du modèle calculé sur S_2. On construit D_2 de manière symétrique. On remarque alors que D_1 apparaît adaptée pour une mesure de

similarité dans le cas où $S_1 \subset S_2$ alors que D_2 sera plus efficace si $S_2 \subset S_1$.

Les mesures D_1 et D_2 ont donc des caractéristiques différentes et apportent chacune une part d'information quant à la similarité des séquences. Pour cette raison, nous proposons d'introduire deux nouvelles mesures de similarité qui permettent de fusionner ces informations :

$$\begin{aligned} D_{avg}(S_1, S_2) &= \frac{D_1 + D_2}{2} \\ D_{min}(S_1, S_2) &= \min(D_1, D_2) \end{aligned}$$

5.2 Complexité

La complexité des approches est cruciale pour les applications visées, principalement à cause des volumes de données qu'elles doivent manipuler. Bien que la présente étude soit préliminaire, et qu'en aucune manière nous puissions pour le moment passer à l'échelle, nous avons tenu à analyser la complexité algorithmique des deux approches. Ici, nous supposons que les deux séquences à comparer sont composées chacune de N vecteurs de dimension d.

La DTW a une complexité en $\mathrm{O}(d \times N^2)$. Il existe toutefois des approximations économes de la distance permettant de ne la calculer complètement que sur les éléments prometteurs [5]. Dans le cas des SVM, il nous faut prédire $N \times d$ valeurs, et pour chacune de ces valeurs, calculer la somme de n termes, où n est le nombre de vecteurs supports. La complexité algorithmique est donc en $\mathrm{O}(d \times n \times N)$, avec $n \ll N$.

5.3 Exemple d'une séquence déformée

Pour mieux comprendre les points forts et faibles de chacune des deux approches, nous proposons de les illustrer par un exemple précis. Supposons que l'on veuille interroger une base de séquences à partir d'une requête qui soit un élément de la base dont une partie a été remplacée par une constante (par exemple un écran noir pour une base de vidéo ou un *bip* dans le cas du son). Ici, S_1 correspond à la séquence requête et S_2 à une séquence de la base.

Intéressons nous pour le moment à la DTW. Que l'on utilise D_1 ou D_2, le principe reste le même : le fait d'introduire une distorsion particulière au sein de la requête aura pour effet d'augmenter la mesure de similarité, quelle que soit la séquence de la base considérée, du fait de la difficulté d'aligner cette partie spécifique du signal. Au final, toutes les distances seront augmentées d'une quantité à peu près égale, et donc l'impact sur le classement final par similarité décroissante des extraits de la base sera limité.

Dans le cas des SVM, il convient de faire la distinction entre D_1 et D_2. Si l'on utilise la première de ces mesures de similarités, le modèle utilisé est donc celui issu de S_2. Ce modèle, qui prédit la valeur prise par un descripteur en fonction des valeurs de ses voisins, sera d'autant plus performant à prédire la partie constante de la requête que les vecteurs supports de la séquence modélisée auront un voisinage constant. Ainsi, des séquences qui n'avaient *a priori* rien à voir avec la séquence de base pourront être jugées similaires, ce qui peut perturber fortement les performances de cette approche. Si l'on considère maintenant la mesure de similarité D_2, pour laquelle le modèle est calculé sur la requête, l'effet de cette distorsion sera moindre car les SVM, comme on a pu le voir sur la figure 1, ne retiennent qu'un faible nombre de vecteurs supports dans les parties les moins variantes du signal. Ainsi, peu de vecteurs supports correspondraient à la partie altérée de la séquence, et donc cette partie aurait peu de poids au moment de la prédiction des séquences de la base.

Nous voyons donc que les divergences d'approche entre la SVM et la DTW mènent à de véritables différences dans les résultats que l'on peut attendre de ces méthodes. Il est à noter que le cas présenté ici est très spécifique au type de déformation appliquée à la requête et qu'il n'est donc pas possible d'en déduire une quelconque généralité quant à la capacité de l'une ou l'autre des mé-

thodes à faire face à tout type d'altération de la requête.

6 Expériences

Dans cette partie, nous comparons les performances des deux méthodes présentées pour une tâche de recherche par le contenu de séquences au sein d'une base de données. Cette base a été constituée à partir d'un enregistrement d'une durée d'une heure de la station de radio *France Info* effectué le 14 avril 2003. Celui-ci contient des répétitions de *jingles* et d'émissions. Ce flux a ensuite été découpé en extraits de 5 secondes chacun, sans recouvrement entre extraits successifs. Pour chaque extrait, la séquence de descripteurs correspond aux 12 premiers coefficients cepstraux sur une échelle de fréquence Mel (MFCC) [4], calculés toutes les 10 ms. sur une fenêtre de 20 ms.

Les expériences menées consistent à ordonner les séquences de la base par rapport à des requêtes, les requêtes apparaissant 4 fois dans la base. Les résultats sont présentés sous la forme de deux indicateurs, notés R_m et R_-. Le premier correspond à la médiane des rangs assignés par la méthode considérée aux 4 vrais positifs et le deuxième est égal au plus mauvais de ces rangs. Le classement optimal, qui correspondrait aux 4 vrais positifs classés aux 4 premiers rangs, donnerait $R_m = 2,5$ et $R_- = 4$. Pour tout autre classement, on aura $R_m \geq 2,5$ et $R_- > 4$.

6.1 Représentation par SVM

Le but de cette première série d'expériences est de comprendre et régler les paramètres noyau et ϵ du modèle SVM. Pour cela, nous utilisons la moitié de la base décrite plus haut, ce qui est suffisant pour obtenir les résultats qualitatifs voulus. La requête utilisée pour interroger cette base est issue du *jingle* précédant le *flash info* de *France Info*. Ce *jingle* se retrouve à trois reprises dans la base. Pour deux de ces trois occurrences, le découpage choisi pour notre base coïncide avec le temps de début du *jingle*. Par contre, pour la dernière, ce *jingle* s'étend sur deux extraits consécutifs, couvrant les deux dernières secondes de l'un et les trois premières du suivant. On obtient donc une vérité terrain faite de 4 extraits que l'on considère comme similaires à la requête, tout en sachant que les degrés de similarité sont différents. La similarité de cette requête à chacun des éléments de la base sera ici calculée en utilisant les modèles des séquences de la base pour prédire la requête.

Noyau	R_m	R_-
Produit scalaire usuel	4	44
Fonction polynomiale	4	44
Fonction à base radiale	2,5	4
Réseau de neurones à 2 niveaux	4	44

TAB. 2 – Classements retournés pour différents noyaux. Ici, $\epsilon = 0,25$.

6.1.1 Choix du noyau

Nous regardons d'abord la capacité des différents noyaux à modéliser nos séquences efficacement. Une modélisation efficace, dans ce cas, se doit de n'être ni trop générique (car, alors, tous les modèles seraient assimilables), ni trop spécifique (car les modèles seraient trop peu robustes aux déformations). Pour toutes les expériences, nous avons fixé $H = 50$. En effet, cette valeur réglant la taille des plus petites séquences reconnaissables, une durée d'une demi-seconde nous est apparue comme raisonnable.

Les résultats donnés dans le tableau 2 montrent que l'utilisation d'un noyau de type radial se révèle être la meilleure option, puisqu'elle permet d'obtenir un classement idéal : les 4 vrais positifs classés aux 4 premiers rangs. Comme il était prévisible, les deux extraits les moins évidents à reconnaître sont ceux qui n'ont qu'une similarité partielle avec la requête. Ceci explique que l'on a parfois une valeur de R_m faible voire optimale (3 des 4 extraits ont été relativement bien reconnus) alors que R_-

Valeur de ϵ	R_m	R_-
0,25	2,5	4
0,5	3,5	6
0,75	5,5	19

TAB. 3 – Classements retournés pour différentes valeurs de ϵ avec un noyau radial.

est assez élevé (l'extrait correspondant au vrai positif qui n'a que 2 secondes en commun avec la requête s'est retrouvé mal classé). Les bons résultats du noyau radial nous semblent liés à la présence de la norme $||x - y||$ plus adaptée à la prédiction que le produit scalaire $x \cdot y$.

6.1.2 Choix de ϵ

Le tableau 3 montre qu'une faible valeur de ϵ permet une modélisation plus fiable. Il est à noter que nous n'avons pas cherché à utiliser des valeurs encore plus faibles pour ce paramètre car, pour $\epsilon = 0,25$, on n'arrive à trouver un modèle convergent que dans moins de 8% des cas. Dans tous les autres cas, le critère d'arrêt $err \leq \epsilon$ n'est pas vérifié et c'est une limitation du nombre d'itérations qui stoppe le processus de modélisation.

6.2 Comparaison des deux approches

Nous évaluons maintenant les performances des SVM en les comparant à celles obtenues avec la DTW sur la base de données complète. Nous présentons des résultats avec 4 types de requêtes différentes : une requête incluse dans la base, une sous-séquence, une sur-séquence et une version déformée de la séquence d'origine où la troisième seconde de l'extrait a été remplacée par un *bip*.

Nous évaluons également ici l'influence du choix de la mesure de similarité. Nous noterons dans la suite D_1 les expériences pour lesquelles la similarité entre une requête q et un élément c de la base est donnée par $D_1(q, c) = d(q, c)$. De la même façon, D_2 correspond à $D_2(q, c) = d(c, q)$.

Requête	D_1		D_2	
	R_m	R_-	R_m	R_-
Séquence de la base	2,5	111	2,5	69
Sous-séquence	2,5	37	5,5	83
Sur-séquence	3	167	2,5	69
Séquence déformée	2,5	86	8	59

Requête	D_{avg}		D_{min}	
	R_m	R_-	R_m	R_-
Séquence de la base	2,5	76	2,5	111
Sous-séquence	2,5	58	2,5	37
Sur-séquence	2,5	97	2,5	129
Séquence déformée	2,5	65	2,5	87

TAB. 4 – Expériences sur la base complète. Valeurs de R_m et R_- pour plusieurs types de distances dérivées de la DTW et pour les 4 requêtes décrites.

6.2.1 Recherche par DTW

Les résultats obtenus pour une recherche par DTW sont donnés dans le tableau 4. Dans la partie supérieure du tableau, les valeurs de R_m sont assez faibles, ce qui montre que la plupart des vrais positifs sont bien classés. Par contre, pour chacune des expériences, la valeur de R_- est assez éloignée de l'optimum. Ceci fait écho à nos propos section 1 : la DTW n'est pas pleinement efficace lorsqu'il s'agit de comparer des séquences mal recalées, malgré le relâchement des contraintes aux bords.

On note également la complémentarité des mesures de similarité D_1 et D_2, la première étant plus efficace pour la recherche de sous-séquences tandis que la seconde est un meilleur choix pour la recherche de sur-séquences. Ceci justifie l'utilisation d'une mesure de similarité faisant intervenir à la fois D_1 et D_2, comme le présente la partie basse du tableau, où D_{avg} apparaît plus appropriée que D_{min}.

6.2.2 Recherche par SVM

Les résultats avec une approche SVM sont donnés dans le tableau 5. On remarque que, comme expliqué dans la section 5.3, cette approche est peu performante dans le cas d'une requête déformée. Pour les autres résultats, on voit là encore la complémentarité de D_1 et D_2. On note également dans le cas des SVM que D_{min} permet d'obtenir de meilleurs résultats que D_{avg}.

La comparaison des deux approches étudiées met en évidence l'intérêt des deux méthodes pour retrouver efficacement les extraits bien recalés, dans le cas de requêtes peu déformées. Par contre, les SVM sont clairement moins sensibles aux problèmes de recalages, ce qui se traduit par une valeur plus faible de R_- : l'extrait le plus mal classé étant, dans les faits, l'extrait qui n'a que deux secondes de similaires avec la requête, la valeur de R_- renvoie directement à la capacité de chacune des méthodes de reconnaître un extrait mal recalé. On note toutefois que les SVM sont plus sensibles aux déformations telles que l'insertion d'un *bip* d'une durée non négligeable au sein de la requête.

Requête	D_1		D_2	
	R_m	R_-	R_m	R_-
Séquence de la base	2,5	4	4	117
Sous-séquence	2,5	6	3,5	132
Sur-séquence	6	168	3,5	143
Séquence déformée	499	610	6,5	102
Requête	D_{avg}		D_{min}	
	R_m	R_-	R_m	R_-
Séquence de la base	2,5	90	2,5	4
Sous-séquence	2,5	104	2,5	7
Sur-séquence	3	119	3,5	11
Séquence déformée	42	130	21,5	244

TAB. 5 – Expériences sur la base complète. Valeurs de R_m et R_- pour plusieurs types de distances dérivées de la SVM et pour les 4 requêtes décrites.

7 Conclusion

L'étude menée dans cet article montre que l'utilisation des SVM comme modèle de prédiction offre une alternative intéressante à la DTW pour la comparaison de séquences et la recherche approximative de motifs. En particulier, les expériences menées montrent que les SVM sont plus robustes que la DTW aux troncatures et donc plus adaptés à des requêtes partielles. En revanche, les SVM sont plus sensibles que la DTW à certaines transformations du signal.

Cette première étude ouvre des perspectives sur la faisabilité d'un système d'indexation de séquences basé sur la comparaison de modèles SVM. Notamment, une piste de travail intéressante consiste à définir une distance entre modèles, basée sur les vecteurs supports, afin de permettre une comparaison rapide des séquences. Par ailleurs, si les résultats mettent en évidence les limites des SVM face à une occultation partielle par un *bip*, il convient d'étudier dans un cadre applicatif réel les déformations que permettent de compenser une approche par régression SVM.

Références

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3) :403–410, October 1990.

[2] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *47th IEEE Symp. on Foundations of Computer Science*, 2006.

[3] E. Bruno and S. Marchand-Maillet. Prédiction temporelle de descripteurs visuels pour la mesure de similarité entre vidéos. In *Proc. of GRETSI*, 2003.

[4] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. on ASSP*, 28(4), 1980.

[5] E. Keogh. Exact indexing of dynamic time warping. In *Proc. VLDB*, 2002.

[6] H. Lejsek, F. H. Ásmundsson, B. Þ. Jónsson, and L. Amsaleg. Efficient and effective image copyright enforcement. In *Proc. of BDA*, Saint Malo, France, 2005.

[7] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209 :415–446, 1909.

[8] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proc. of CVPR*, 2006.

[9] H. Sakoe and S. Chiba. Dynamic programming optimization for spoken word recognition. *IEEE Trans. on ASSP*, 26(27), 1978.

[10] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag, 1995.

[11] B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *Proc. ICDE*, 1998.