Conditional Random Fields pour le traitement de séquences

Sujet IREC - Master 2 IAD

Nicolas Cheifetz

Encadrant : Thierry Artières

Janvier 2009

Table des matières

1 Introduction

Les conditional random fields (ou CRF) sont des modèles Markoviens conditionnels qui ont été proposés pour remédier à certains défauts des modèles Markoviens plus classiques. Ils peuvent être appliqués à des données séquentielles, des données de type arbre et de façon plus générale à tout type de données structurées. Ce sont des modèles conditionnels implémentant une loi de probabilité conditionnelle des labels conditionnellement aux données qui sont donc appris avec un critère discriminant. Le but de l'étude est de s'intéresser à une variante dite "segmentale" ou semi-Markovienne des CRFs et à un algorithme d'apprentissage efficace pour ces modèles.

Ici, on étudie des données séquentielles (une séquence est un ensemble ordonné d'éléments) dans un cadre supervisé (apprendre la formation des données d'observations et évaluer ses performances par des essais sur les données de test).

1.1 Cadre du problème

L'objectif est l'étiquetage (ou segmentation) d'une séquence finie d'observations $\mathbb{X}$. Autrement dit il faut identifier la meilleure séquence de labels (également appelés étiquettes, ou classes) $\mathbb{Y}^*$ connaissant la séquence $\mathbb{X}$.

Notation :
- séquence d'observations : $\mathbb{X} = (x_1, \ldots, x_T)$
- séquence de labels : $\mathbb{Y} = (y_1, \ldots, y_T)$

On a bien $\#(\mathbb{X}) = \#(\mathbb{Y}) = T$, fini.

Le but est de trouver la séquence de labels optimale :

$$\begin{aligned}\mathbb{Y}^* &= \arg\max_{\mathbb{Y}}\{P(\mathbb{Y}|\mathbb{X})\} \\ &= \arg\max_{\mathbb{Y}}\left\{\frac{P(\mathbb{Y},\mathbb{X})}{P(\mathbb{X})}\right\} \\ &= \arg\max_{\mathbb{Y}}\{P(\mathbb{Y},\mathbb{X})\}\end{aligned}$$

Autement dit, il faut déterminer quelle est la séquence d'étiquettes $\mathbb{Y}$ qui maximise la probabilité que la séquence de données $\mathbb{X}$ soit étiquetée par $\mathbb{Y}$. Une classe est attribuée pour toute observation.

Remarque : on ne traitera pas le cas d'une séquence d'observations représentant un signal.
Dans une représentation graphique, on supposera ici qu'une étiquette correspondra à un seul état. Dans le cas d'un signal, une étiquette peut être représentée par un ou plusieurs états. Il s'agit, par exemple, de données échantillonnées sur de la parole, de données manuscrites ou encore des accélérations de manettes Wii 4.5.

2 Les différents modèles étudiés

Tout d'abord, rappelons que tous les modèles d'apprentissage de type Markovien ont un point commun : l'**hypothèse de Markov**. Cette propriété signifie que chaque état du modèle graphique ne dépend que des états précédents. Autrement dit, la probabilité d'avoir une certaine étiquette à l'étape t ne dépend que de l'étiquette à l'étape $t-1$.

Cette hypothèse est fondamentale pour la conception des modèles suivants.

2.1 HMM

Les HMM [1] (Hidden Markov Model) sont des modèles génératifs (discrets ou continus). Ce sont des modèles très répandus pour des tâches de classification. Les HMMs ont été introduits dans les années 1960-70s par le professeur Baum et ses collaborateurs (Baum, 1970[1]).

La segmentation est faite à partir de la loi jointe : $\mathbb{Y}^* = \arg\max_{\mathbb{Y}}\{P(\mathbb{Y}, \mathbb{X})\}$.

$$P(\mathbb{X}, \mathbb{Y}) = P(x_1, y_1) \times \prod_{t=2}^{T} P(y_t|y_{t-1}) \times P(x_t|y_t)$$

On observe alors deux défauts :
- calculer la loi jointe est une manière détournée de calculer $P(\mathbb{Y}|\mathbb{X})$
- une hypothèse forte est l'indépendance des données $\mathbb{X}$

2.2 MEMM

Les MEMM [2] (Maximum Entropy Markov Models) sont des modèles conditionnels. Ce type de modèle opère uniquement ses calculs sur la loi conditionnelle $P(\mathbb{Y}|\mathbb{X})$ et la loi marginale $P(\mathbb{Y})$.

La segmentation est faite selon la formule : $\mathbb{Y}^* = \arg\max_{\mathbb{Y}}\{P(\mathbb{Y}|\mathbb{X})\}$.

$$P(\mathbb{Y}|\mathbb{X}) = P(y_1|x_1) \times \prod_{t=2}^{T} P(y_t|y_{t-1}, \mathbb{X}), \quad \text{où } \sum_{y'} P_y(y'|\mathbb{X}) = 1 \quad (1)$$

$P_y(y'|\mathbb{X})$ désigne la probabilité de transition pour passer de l'état y à l'état y'.

Les MEMM améliorent les propriétés des modèles génératifs : il n'y a pas d'a priori sur les données. Cependant, ces modèles admettent un certain défaut nommé "label biais" (Lafferty, 2007[6]). Le défaut est dû à la propriété de normalisation des lois de transition (1).

2.3 CRF

Les CRF [3] (Conditional Random Fields) résolvent en partie les défauts des modèles précédents. Par rapport aux MMEM, la propriété de normalisation est appliquée sur toutes les transitions du modèle.

La segmentation est faite selon la formule : $\mathbb{Y}^* = \arg\max_{\mathbb{Y}}\{P(\mathbb{Y}|\mathbb{X})\}$.

Des modèles conditionnels comme les CRFs induisent le paramétrage d'un certain nombre de variables au cours de l'apprentissage. C'est le cas du vecteur de poids W. Donc, la phase d'inférence pour résoudre la segmentation peut aussi s'écrire : $\mathbb{Y}^* = \arg\max_{\mathbb{Y}}\{P(\mathbb{Y}|\mathbb{X}, W)\}$.

$$P(\mathbb{Y}|\mathbb{X}, W) = \frac{\text{score}(\mathbb{X}, \mathbb{Y}, W)}{Z_W(\mathbb{X})} = \frac{e^{W \cdot F(\mathbb{X}, \mathbb{Y})}}{\sum_{\mathbb{Y}'} e^{W \cdot F(\mathbb{X}, \mathbb{Y}')}}$$

où, • W est le *vecteur de poids* (de taille L)

1. Dans la littérature en français, on les appelle MMC (Modèles de Markov Cachés).
2. Dans la littérature en français, on les appelle MMEM (Modèles de Markov à Entropie Maximale).
3. Dans la littérature en français, on les appelle CMC (Champs de Markov Conditionnels).

- $Z_W(\mathbb{X})$ est le *facteur de normalisation* ; $Z_W(\mathbb{X}) = \sum_{\mathbb{Y}} e^{W \cdot F(\mathbb{X},\mathbb{Y})}$
- $F(\mathbb{X},\mathbb{Y})$ est le *vecteur de caractéristiques* de taille L (doit être défini dans les hypothèses)

$$F(\mathbb{X},\mathbb{Y}) = \sum_{t=1}^{T} f(t,\mathbb{X},\mathbb{Y}) = \left(\sum_{t=1}^{T} f_1(t,\mathbb{X},\mathbb{Y}), \ldots, \sum_{t=1}^{T} f_L(t,\mathbb{X},\mathbb{Y}) \right)$$

Selon la topologie du graphe à états, on applique différents algorithmes durant l'inférence :
chaîne : `algorithme de Viterbi`
arbre : `algorithme Belief Propagation` (Weiss, 2001[12])
graphe qcq : `algorithme Loopy Belief Propagation` (Murphy, 1999[9])

L'étape d'apprentissage détermine W^*. Nous sommes dans un cadre supervisé, nous fixons donc W avec la base d'apprentissage. On choisit le W qui maximise la Log-vraisemblance sur la segmentation :
Soit $BA = \{(\mathbb{X}_n, \mathbb{Y}_n)_{n\in\{1,\ldots,N\}}\}$.
On a : $W^* = \arg\max_W \{L(W)\}$
où, $L(W) = \sum_{n=1}^{N} \log\left(P(\mathbb{Y}_n|\mathbb{X}_n, W)\right) = \sum_{n=1}^{N} \log\left(\frac{e^{W\cdot F(\mathbb{X}_n,\mathbb{Y}_n)}}{Z_W(\mathbb{X}_n)}\right) = \sum_{n=1}^{N} \left(W \cdot F(\mathbb{X}_n,\mathbb{Y}_n) - \log(Z_W(\mathbb{X}_n)) \right)$
La fonction objectif $L(W)$ est convexe (i.e. $\exists! W \ / \ \nabla L(W) = 0$ et ce W est W^*), on utilise alors un algorithme de programmation dynamique pour résoudre la maximisation, par exemple une méthode de descente de gradient comme la méthode de quasi-Newton à mémoire limitée (Liu, 1989[7]).

Remarque : Un modèle CRF peut être semblable à un modèle de Markov caché pour une certaine fonction caractéristique (Lafferty, 2001[6]).

2.4 Semi-CRF

Le modèle Semi-CRF [4] (Semi-Markov Conditional Random Fields) est un modèle segmental. On le distingue d'un CRF par la notion de segment. Un segment est une sous-séquence de la séquence d'observation $\mathbb{X}$.
Formellement, on définit une séquence de segments de la manière suivante :

$$\mathbb{S} = (S_1, \ldots, S_J) \quad \text{où,} \quad S_j = (t_j, u_j, y_j), \forall j \in \{1,\ldots,J\} \quad \text{et} \quad J \leq T$$

où : t_j est la position de début
u_j est la position de fin
y_j est la classe des observations entre t_j et u_j inclus

Autrement dit, on a : $\forall j, (\forall x_i / t_j \leq i \leq u_j, \ \mathbb{Y}(x_i) = y_j)$.

Deux propriétés : $\forall j, \quad 1 \leq t_j \leq u_j \leq |\mathbb{S}| = J$
et, $t_{j+1} = u_j + 1$

Remarque : à l'intérieur d'un segment, les transitions entre états peuvent être non markoviennes. Cependant les transitions entre segments sont bien markoviennes.

De par la définition d'un segment, cette séquence nous suffit pour réaliser l'étiquetage. En effet, un modèle segmental ne retourne plus une séquence d'étiquettes mais une séquence segmentale.
La segmentation est faite selon la formule : $\mathbb{S}^* = \arg\max_{\mathbb{S}} \{P(\mathbb{S}|\mathbb{X})\}$.

4. Dans la littérature en français, on les appelle CMCS (Champs de Markov Conditionnels Segmentaux).

De manière analogue aux CRFs, on calcule en réalité (après apprentissage) :

$$\mathbb{S}^* = \arg\max_{\mathbb{S}}\{P(\mathbb{S}|\mathbb{X}, W)\}.$$

$$P(\mathbb{S}|\mathbb{X}, W) = \frac{\text{score}(\mathbb{X}, \mathbb{S}, W)}{Z_W(\mathbb{X})} = \frac{e^{W \cdot G(\mathbb{X},\mathbb{S})}}{\sum_{\mathbb{S}'} e^{W \cdot G(\mathbb{X},\mathbb{S}')}}$$

où :
- W est le vecteur de poids (de taille K)
- $Z_W(\mathbb{X})$ est le *facteur de normalisation* ; $Z_W(\mathbb{X}) = \sum_{\mathbb{S}} e^{W \cdot G(\mathbb{X},\mathbb{S})}$
- $G(\mathbb{X}, \mathbb{S})$ est le *vecteur de caractéristiques segmentales* de taille K (doit être défini dans les hypothèses)

$$G(\mathbb{X}, \mathbb{S}) = \sum_{j=1}^{J} g(j, \mathbb{X}, \mathbb{S}) = \left(\sum_{j=1}^{J} g_1(j, \mathbb{X}, \mathbb{S}), \ldots, \sum_{j=1}^{J} g_K(j, \mathbb{X}, \mathbb{S}) \right)$$

Rappelons qu'un semi-CRF est un modèle Markovien (chaque état ne dépend que de l'état précédent), on pose alors : $\forall j \in \{2, \cdots, J\},\ g(j, \mathbb{X}, \mathbb{S}) = g(y_j, y_{j-1}, \mathbb{X}, t_j, u_j)$.

On reformule alors le problème d'inférence :

$$\begin{aligned} \mathbb{S}^* = \arg\max_{\mathbb{S}}\{P(\mathbb{S}|\mathbb{X}, W)\} &= \arg\max_{\mathbb{S}} \left\{ \frac{e^{W \cdot G(\mathbb{X},\mathbb{S})}}{Z_W(\mathbb{X})} \right\} \\ &= \arg\max_{\mathbb{S}}\{\exp(W \cdot G(\mathbb{X}, \mathbb{S}))\} \\ &= \arg\max_{\mathbb{S}}\{W \cdot \sum_{j=1}^{J} g(y_j, y_{j-1}, \mathbb{X}, t_j, u_j)\} \end{aligned}$$

L'étape d'inférence d'un modèle Semi-CRF diffère de celle d'un CRF dans le sens où elle inclut deux "pseudo-inférences". Cette étape a toujours pour but d'attribuer la meilleure séquence d'étiquettes $\mathbb{Y}$ à la séquence d'observation $\mathbb{X}$ mais ici, l'inférence détermine la meilleure séquence de segments $\mathbb{S}^*$ et assure donc, la meilleure segmentation $\mathbb{Y}^*$. Ces deux inférences sont réalisées simultanément, on considère alors que ces inférences peuvent se réduire à une seule.

Beaucoup de caractéristiques (features) peuvent être extraites d'une séquence numérique ; les segments en font partie. Dans de très nombreux cas, certaines caractéristiques riches d'informations sont de type segmentales par nature. La recherche de la meilleure séquence de segments a été classée dans la même catégorie de problèmes d'apprentissage que les méthodes maximisant la marge (Mc Donald, 2005[8]) et les méthodes de vraisemblance conditionnelle basées sur les CRFs (Sarawagi, 2004[11]).

En particulier, on constate expérimentalement que ce modèle est plus performant face à un modèle CRF non segmental dans le cas de problèmes NER. Par contre, le déroulement d'un modèle Semi-CRF est plus lent que celui d'un CRF.
Dans le pire des cas, l'inférence pour un modèle segmental implique des calculs de programmation dynamique cubiques en la taille de la plus longue séquence (=$Lseq$). En revanche, l'étape d'inférence pour un modèle d'apprentissage séquentiel non segmental entraîne des calculs linéaires en $Lseq$. C'est par exemple le comportement des méthodes de coupe par plan (Tsochantaridis et al., 2005) et des méthodes de gradient (Bartlett et al., 2005) lorsqu'ils sont utilisés comme algorithmes d'inférence.

3 Algorithme d'inférence efficace pour des modèles segmentaux

Notation : on note *Lseq*, la taille de la plus longue séquence.

Un modèle d'apprentissage segmental inclut toujours un algorithme de programmation dynamique lors de l'étape d'inférence. Dans l'article (Sarawagi, 2006[10]), il est montré que ces algorithmes impliquent des calculs cubiques en *Lseq*. L'algorithme présenté ici permet d'abaisser la complexité de l'inférence pour un modèle segmental au niveau d'un modèle séquentiel classique de segmentation.
Une solution erronée serait de limiter la taille maximale d'un segment. Dans ce cas, un modèle segmental met en 3 à 10 fois plus de temps pour l'apprentissage, qu'un modèle non segmental.

L'idée de l'algorithme se base sur le fait qu'après la segmentation la plupart des caractéristiques se trouvent sur plusieurs segments. Il est possible d'exprimer ces caractéristiques de manière plus compacte. Nous allons décrire un algorithme d'inférence proportionnel au nombre de caractéristiques sans tenir compte du nombre de segments qu'elles "chevauchent".
Pour des tâches standard d'extraction, l'apprentissage se déroule alors dans un temps comparable à celui d'un modèle non segmental.

3.1 Score d'une séquence segmentale

Soit la séquence segmentale $\mathbb{S} = (S_1, \ldots, S_J)$, où $S_j = (t_j, u_j, y_j)$, $\forall j \in \{1, \ldots, J\}$ et $J \leq T$.

Pour déterminer le score de cette séquence, il faut se donner deux types de "potentiels" :

$\theta_{t_j}(y_{-j}, y_j)$: **potentiel de transition** sur un segment qui commence à t_j avec le label y_j, et le label du segment précédent est y_{j-1}

$\theta_{t_j:u_j}(y_j)$: **potentiel segmental** de t_j à u_j ; $\quad \theta_{t_j:u_j}(y_j) = exp\left(\sum_{k=1}^{K} w_k \times g(y_j, y_{j-1}, \mathbb{X}, t_j, u_j)\right)$, où w_k est le poids de la k^e fonction caractéristique segmentale g_k

Le score total pour une séquence de segments $\mathbb{S}$ de taille J est :

$$\text{score}(\mathbb{S}) = \prod_{j=1}^{J} \theta_{t_j}(y_{-j}, y_j) \times \theta_{t_j:u_j}(y_j) \,.$$

L'inférence se fait sur le logarithme de ce score. Sa résolution se fait par un algorithme de forward/-backward (programmation dynamique) en temps polynomial.
Autrement dit :

$$\begin{aligned} \mathbb{S}^* &= \arg\max_{\mathbb{S}} \{\log(\text{score}(\mathbb{S}))\} \\ &= \arg\max_{\mathbb{S}} \left\{ \sum_{j=1}^{J} \log\left(\theta_{t_j}(y_{-j}, y_j)\right) + \log\left(\theta_{t_j:u_j}(y_j)\right) \right\} \end{aligned}$$

4 Applications possibles

4.1 NER

Le problème **Named entity recognition** (NER) (aussi appelé entity identification et entity extraction) correspond à une sous-tâche d'extraction d'informations qui vise à localiser et classer les éléments atomiques (mots) dans le texte en catégories prédéfinies telles que les noms de personnes, les organisations, les lieux, les expressions de temps, les quantités, les valeurs monétaires, les pourcentages, etc.

4.2 NP Chunking

La tâche **Text chunking** divise des phrases en expressions sans chevauchement.

NP chunking (traite une partie de la tâche du Text chunking) divise des phrases en expressions nominales sans chevauchement (NP=Noun Phrases).
La performance de l'algorithme d'apprentissage est mesurée avec deux scores : la précision et le rappel.

- Précision = mesure combien de NPs ont été trouvés correctement par l'algorithme
- Rappel = taux qui contient le pourcentage de NPs définis dans le corpus qui ont été trouvés par l'algorithme

4.3 POS tagging

Part-of-speech tagging (POS tagging ou POST), aussi appelé grammatical tagging ou word-category disambiguation, est un processus d'attribution d'un marqueur (étiquette) à un mot du Part-of-speech, comme substantif, verbe, pronom, préposition, adverbe, adjectif ou un autre marqueur de catégorie lexicale
L'entrée à un algorithme de POST est une chaîne de mots d'une phrase en langage naturel et un tagset (= liste finie d'étiquettes du Part-of-speech). Le résultat est la segmentation optimale : la meilleure étiquette pour chaque mot.

Remarques :

Beaucoup de travaux ont été faits sur le POST pour l'anglais. Le premier algorithme pour attribuer automatiquement des étiquettes à une partie du discours a été fondé sur des règles. Le ENGTWOL tagger (Voutilainen, 1995) est fondé sur une règle d'étiquetage construite en deux étapes. Les probabilités dans l'étiquetage ont été utilisées par (Stolz et al. 1965) et de nombreux algorithmes d'étiquetage stochastiques ont été construits dans les années 1980 (Church 1988). Il y avait également Transformation-Based Tagging, un exemple d'apprentissage par Transformation-Based.

Aujourd'hui, on utilise plutôt des modèles stochastiques issus de la reconnaissance de données manuscrites. Les modèles d'étiquetage stochastiques comme les HMMs sont fondés soit sur le choix de la séquence d'étiquettes qui maximise le produit entre la vraisemblance du mot et la probabilité de la séquence d'étiquettes, soit à l'aide d'arbres de décision ou de modèles d'entropie maximale pour combiner caractéristiques probabilistes.

4.4 Références

Utilisation d'un modèle segmental pour certaines applications :

- Extraction d'information : Sarawagi & Cohen, 2004
- NER : McDonalds et al., 2005
- Syntactic Chunking : DauméIII & Marcu, 2005
- localisation de gènes/protéines

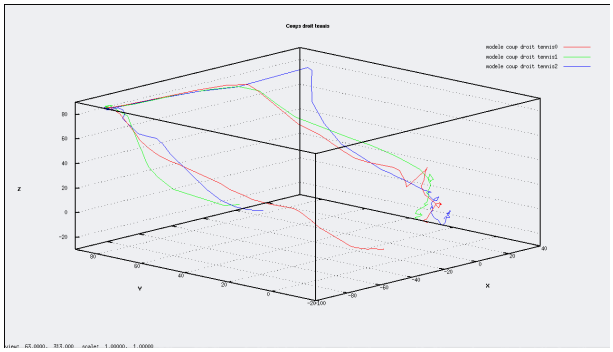
4.5 Données de type signal

Les exemples d'application cités précédemment ne traitent pas des données de type signal. Les données de type signal comprennent une dimension particulière, celle du temps. Des échantillons tels que les données manuscrites ou les accélérations d'une manette Wii contiennent des caractéristiques segmentales reliées au temps. L'utilisation d'un modèle segmental pour ce type de données semble alors cohérent.

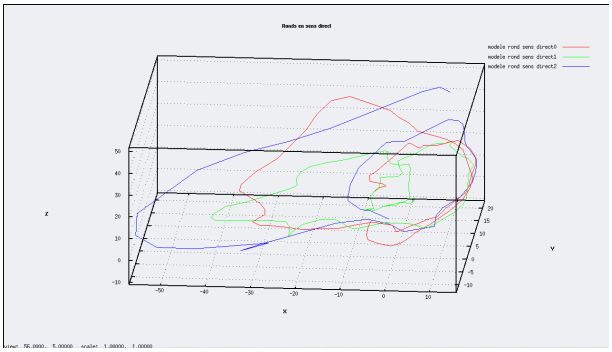
Nous prenons ici l'exemple des accélérations d'une Wiimote car il a fait l'objet d'un projet en M1[3]. Ce travail encadré avait pour but de concevoir un outil de base pour la manipulation, le traitement de données et la reconnaissance de gestes.

Exemples de gestes captés par une WiiRemote (accélérations tridimensionnelles) :

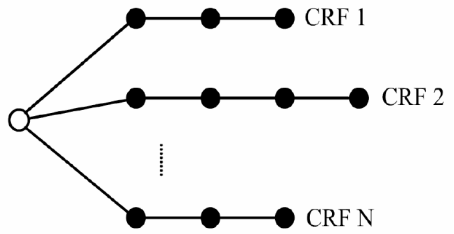
Trois coups droits :



Trois ronds dans le sens direct :



On s'intéresse ici à une instance particulière des CRFs : les CRFBranch.
Dans une représentation graphique, son architecture est un mélange de structures en chaînes. Chaque "branche" représente une classe et le nombre d'états par branche est variable.



Un CRFBranch pour la classification de séquences.

Résultats expérimentaux à partir de trois modèles : `DTW`, `LibSVM` et `CRFBranch`[5].

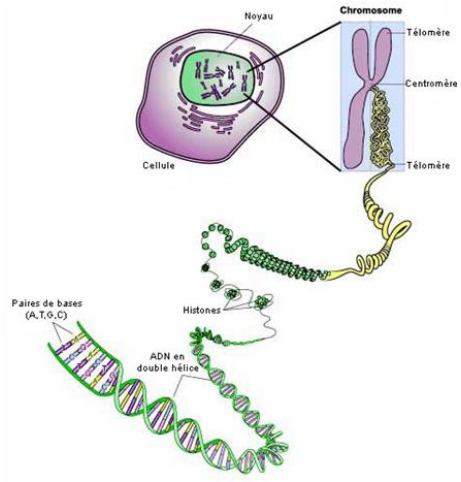
Nom du modèle	Taille de la BA	Taille de la BT	Nb d'expériences	Taux de reco pondéré	Ecart-type
`DTW1`[6]	10	20	3	57%	15.13%
`DTW3`[7]	10	20	3	66%	11.36%
`LibSVM`[8]	10	20	4	39.25%	25.61%
`CRFBranch1`[9]	90	180	6	93.36%	2.76%
`CRFBranch2`[10]	90	180	6	93.56%	3.01%

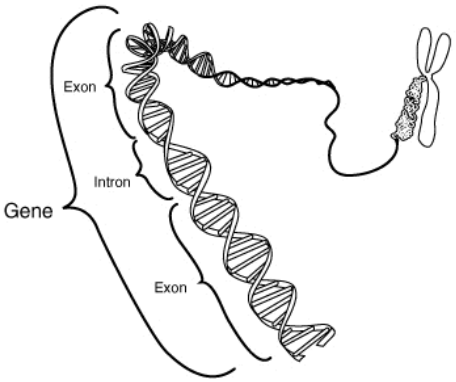
5. Ce modèle a été développé par Trinh Minh Tri Do en C, C++ et Fortran [4].
6. Modèle Dynamic Time Warping à 1 séquence.
7. Modèle Dynamic Time Warping à 3 séquences.
8. LibSVM version 2.86 (Chang, 2001[2]).
9. `CRFBranch` à 1 branche, 30 états par branche et au maximum 200 itérations LBFGS[7] pour l'apprentissage.
10. `CRFBranch` à 1 branche, 30 états par branche et au maximum 100 itérations LBFGS[7] pour l'apprentissage.

4.6 Recherche de gènes dans un génome

Une des principales problématiques posée dans le domaine de la BioInformatique est la recherche de gènes connus dans un génome. On étudie généralement des génomes d'ADN (plus stables que ceux de l'ARN) constitués de bases nucléiques, tels que l'adénine (`A`), la thymine (`T`), la cytosine (`C`) et la guanine(`G`).

Le problème peut se présenter comme la recherche de sous-séquences (gènes) dans une séquence numérique (génome) : lors de l'apprentissage, le modèle apprend sur les gènes mis bout à bout dans des séquences , et, pour l'inférence il essaye de reconnaître des sous-séquences (gènes) dans la séquence (génome) de test.





Typiquement, un gène commence par le motif `ATG`. Il semble alors cohérent d'associer la notion de *gène* dans un génome à la notion de *segment* pour un modèle segmental. Ce modèle comprendrait alors une fonction caractéristique segmentale capable de déterminer des segments commençant par `ATG`. Et chaque classe correspondrait au nom d'un gène.

Selon la présence de certains gènes dans un génome, il est possible de déduire les fonctions biologiques de l'être vivant dont est issu le génome.

5 Conclusion

Nous avons étudié différents modèles séquentiels Markoviens dans un cadre supervisé, notamment les modèles conditionnels. Nous avons vu que le modèle CRF segmental exploite des outils pertinents pour des données séquentielles, comme la notion de segments et des caractéristiques segmentales. Un des principaux avantages des modèles CRFs segmentaux est qu'ils permettent de calculer les propriétés des caractéristiques sur les segments et non plus sur des observations seules. Les étapes d'apprentissage et d'inférence ont été décrites.
Enfin, nous avons présenté de nombreuses applications pour ce type de modèle.

Références

[1] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. In *The Annals of Mathematical Statistics*, volume 41(1), pages 164–171, 1970.

[2] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM : a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[3] Nicolas Cheifetz and Yasmina Seddik. *Traitement de séquences et manette Wii (PIAD 25)*, 2008. Project available at http://che.nico.ifrance.com/PIAD/.

[4] Trinh Minh Tri Do. *Multi branch conditional random fields implementation*, 2007. http://webia.lip6.fr/~do/pmwiki/pmwiki.php/Main/Codes.

[5] Trinh Minh Tri DO and Thierry Artières. Conditional random fields for online handwriting recognition. 2006. http://hal.inria.fr/docs/00/10/42/07/PDF/cr1053220965852.pdf.

[6] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001. http://www.cis.upenn.edu/~pereira/papers/crf.pdf.

[7] D. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. In *Mathematical Programming*, volume 45, pages 503–528, 1989. http://www.springerlink.com/content/k5653wt4q8061176/.

[8] Ryan Mcdonald, Koby Crammer, and Fernando Pereira. Flexible text segmentation with structured multilabel classification. In *In Proc. HLT-EMNLP*, pages 987–994, 2005.

[9] Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. Loopy belief propagation for approximate inference : An empirical study. In *In Proceedings of Uncertainty in AI*, pages 467–475, 1999.

[10] Sunita Sarawagi. Efficient inference on sequence segmentation models. In *Proceedings of the 23^{rd} International Conference on Machine Learning (ICML), Pittsburgh, PA, USA*, 2006. http://www.it.iitb.ac.in/~sunita/papers/icml06.pdf.

[11] Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. In *In Advances in Neural Information Processing Systems 17*, pages 1185–1192, 2004. http://www.cs.cmu.edu/~wcohen/postscript/semiCRF.pdf.

[12] Yair Weiss and William T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13 :2173–2200, 2001.