Versions récursive et adaptative d'une mesure d'indépendance.

Pierre-Olivier AMBLARD[1,2] *

[1]GIPSA-lab, DIS(UMR CNRS 5216), ENSE3-BP 46 38402 Saint Martin d'Hères Cedex

[2]Dept. Math&Stat, The University of Melbourne , Parkville, VIC3010, Australie.

bidou.amblard@gipsa-lab.inpg.fr

Résumé – L'objet de ce papier est de développer des algorithmes récursifs et adaptatifs pour mesurer l'indépendance entre deux suites de variables aléatoires. Nous considérons la mesure HSIC fondée sur le plongement des probabilités dans des espaces de Hilbert à noyau reproduisant (RKHS). Nous donnons deux versions récursive et adaptative d'un algorithme d'estimation de cette mesure d'indépendance. Une première version récursive permet le calcul de HSIC rapidement en utilisant peu de mémoire, autorisant ainsi sont utilisation pratique sur des échantillons de grandes tailles. Une deuxième version récursive incorpore une représentation parcimonieuse (alternative aux approximations de rang faible) qui permet de traiter des données en-ligne et qui se prête à la mise sous forme adaptative.

Abstract – Recursive and adaptive algorithms to measure independence between two series of random variables are developed. We consider the HSIC measure based on embeddings of probability measures in reproducing kernel Hilbert spaces. We give two algorithms for the recursive calculation of HSIC. The first one uses low memory and is adapted to large data sets. The second one support a sparsification front-end and can easily be turned into an adaptive algorithm.

1 Introduction

De nombreux problèmes de traitement du signal font appel explicitement à des mesures d'indépendance. Un exemple fameux est la séparation de source, qui fût initialement résolue en utilisant des mesures d'indépendance. Un autre exemple où l'indépendance intervient est la modélisation graphique de signaux multivariés. En neuroscience, la notion d'indépendance est utilisée pour rendre compte de la connectivité fonctionnelle entre deux aires cérébrales. Dans ce contexte, des études actuelles examinent l'évolution au cours du temps de la connectivité. Rendre dépendant du temps des mesures d'indépendance est donc une question pertinente. Dans [10] par exemple, une mesure de causalité fondée sur une mesure d'indépendance est évaluée au cours du temps en utilisant une fenêtre glissante.

Dans cet article, nous examinons une mesure utilisant les plongements de mesures de probabilité dans des espaces de Hilbert à noyau reproduisant (RKHS pour reproducing kernel Hilbert space). Implicitement, le plongement est effectuée à l'aide d'une fonction non linéaire φ qui envoie l'espace des observations dans un espace de Hilbert $\mathcal{H}$. Si φ est bien choisie, l'espace est un RKHS, et les produits scalaires s'y calculent remarquablement aisément, puisque φ est associée à une fonction définie positive symétrique k, le noyau, qui vérifie $k(x,y) = \left\langle \varphi(x) \middle| \varphi(y) \right\rangle_{\mathcal{H}}$. Les données sont transformées non linéairement, mais plongées dans un espace linéaire, dans lequel les traitements linéaires (utilisant le produit scalaire) se calculent facilement. Autrement dit, il est possible de réaliser des traitements non linéaires à l'aide d'outils linéaires ! Dans cet ordre d'idée, on peut se demander s'il est possible de mesurer l'indépendance entre deux variables en utilisant des mesures de corrélation entre des transformées non linéaires des variables. Cette heuristique fût utilisée par exemple par Jutten&Hérault dans leur algorithme de séparation aveugle de sources. La formalisation récente de cette idée repose sur des idées anciennes de Rényi, et a conduit à utiliser les normes d'opérateurs de covariance entre espaces de Hilbert à noyau reproduisant. Une mesure emblématique de cette approche est appelée HSIC pour Hilbert-Schmidt Independence Criterion.

Dans la suite de cette introduction, nous rappelons l'essentiel relatif à cette mesure. Nous en donnerons ensuite deux versions récursives. Ces versions permettent un calcul rapide (mais toujours en $O(n^2)$) de la mesure sans considérer explicitement les matrices de Gram. La deuxième version autorise alors une implantation approchée utilisant une procédure parcimonieuse (initialement proposée par C. Richard et ses collaborateurs dans un contexte tout différent [6]). Enfin, on propose une version adaptative permettant de détecter des changements dans la structure de dépendance entre deux grandeurs. Le dernier paragraphe sera dédié à des illustrations des différents algorithmes proposés.

Quelques rappels sur HSIC. HSIC a été introduite par Gretton et ses collaborateurs durant la dernière décennie [4, 8]. Soient deux variables aléatoires X, Y définies sur un espace probabilisé $(\Omega, \mathcal{F}, P)$ et prenant leurs valeurs dans respectivement $\mathbb{X}$ and $\mathbb{Y}$ (dotés de tribus adéquates). Soient deux noyaux (fonctions symétriques définies positives) $k_x : \mathbb{X} \times \mathbb{X} \longrightarrow \mathbb{R}$ et $k_y : \mathbb{Y} \times \mathbb{Y} \longrightarrow \mathbb{R}$ (typiquement des noyaux gaussiens dans tout le résumé). Les espaces de Hilbert à noyau reproduisant correspondants sont notés $\mathcal{H}_x$ et $\mathcal{H}_y$ et sont supposés ici séparables [9, 7]. L'opérateur de covariance entre X et Y est l'unique opérateur linéaire borné de $\mathcal{H}_x$ dans $\mathcal{H}_y$ tel que $\mathrm{Cov}\left[f(X), g(Y)\right] = \left\langle g \middle| \Sigma_{YX} f \right\rangle_{\mathcal{H}_y}$. Cet opérateur est bien défini moyennant $E[k_x(X,X)] < +\infty$ et $E[k_y(Y,Y)] < +\infty$. L'opérateur est alors de Hilbert-Schmidt, *i.e.* sa norme de Hilbert-Schmidt (HS) définie par $\|\Sigma_{YX}\|^2 := \sum_i \left\|\Sigma_{YX}\varphi_i\right\|^2_{\mathcal{H}_y}$ est finie, pour toute base orthonormée $\{\varphi_i\}_{i\in\mathbb{N}}$ de $\mathcal{H}_x$ [3].

*P.O.A. est soutenu par une bourse Marie Curie International Outgoing Fellowship de l'Union Européenne.

Un théorème prouvé par Gretton étend un résultat de Rényi stipulant l'équivalence entre indépendance et $\sup_{f,g} \mathrm{Cov}\left[f(X), g(Y)\right] = 0$ pour des fonctions continues bornées f et g [5]. L'extension proposée par Gretton consiste à choisir f et g dans un espace à noyau reproduisant qui doit être suffisamment riche pour que la propriété de Rényi reste valide. Cette richesse est conférée par la propriété d'universalité des noyaux. Un noyau est universel si le RKHS associé est dense dans l'ensemble des fonctions continues bornées. Le théorème de Gretton stipule alors que $\sup_{f\in\mathcal{U}_x, g\in\mathcal{U}_y} \left\langle g\middle|\Sigma_{YX} f\right\rangle_{\mathcal{H}_y} = 0$ est équivalent à l'indépendance entre X et Y. $\mathcal{U}$ est le sous-ensemble des vecteurs de $\mathcal{H}$ de norme inférieure ou égale à 1. L'hypothèse d'universalité permet d'approcher n'importe quelle fonction continue bornée par une suite de fonctions du RKHS, et donc de satisfaire le théorème de Rényi. Cette hypothèse est vérifiée par de nombreux noyaux, dont le noyau gaussien très largement utilisé. Par contre un noyau de type $(1 + \langle x|y\rangle)^q$ génère un espace vectoriel de dimension finie et n'est pas universel.

La magie du résultat de Gretton réside dans le fait que $\sup_{f\in\mathcal{U}_x, g\in\mathcal{U}_y} \left\langle g\middle|\Sigma_{YX} f\right\rangle_{\mathcal{H}_y}$ n'est rien d'autre que la norme usuelle de l'opérateur de covariance ! La norme usuelle est inférieure ou égale à la norme de Hilbert-Schmidt, de sorte que le résultat de Gretton reste valide avec la norme HS. Or cette norme est très simple à évaluer à partir de données [4]. Les calculs de l'estimée peuvent être effectués en $O(n^2)$ pour un échantillon de taille n, une complexité élevée qui peut être réduite en utilisant des approximations de rang faible pour les matrices de Gram. Une version largement utilisée dans le contexte, voir [4, 8], est la transformée de Cholesky incomplète [2].

Dans la suite, des grandeurs ont x ou y en indice. Lorsque les définitions ou les expressions sont communes, ou nous ne les écrivons que pour un des deux indices, ou nous supprimons l'indice ; il est implicitement entendu que la définition ou l'expression en jeu est valide pour les deux indices.

2 Algorithmes

Première forme récursive. L'implantation proposée permet de calculer en des temps raisonnables la mesure pour des échantillons de taille très grande (10^4 à 10^5 échantillons), chose impossible en utilisant des matrices. De nombreuses multiplications peuvent être éliminées et les nécessités de stockage réduites en ne manipulant que des vecteurs. La version empirique que nous utilisons ici n'est seulement qu'asymptotiquement non biaisé. Les développements peuvent se réaliser sur la version non biaisée de [8].

Considérons n données X_i, Y_i. L'estimateur $\widehat{H}_n$ de HSIC $\|\Sigma_{YX}\|^2$ est obtenu à partir des matrices de Gram $\boldsymbol{K}_x$ and $\boldsymbol{K}_y$, dont les (i,j)èmes éléments sont respectivement $k_x(X_i, X_j)$ et $k_y(Y_i, Y_j)$. Soit $\boldsymbol{C}_n := \boldsymbol{I}_n - \mathbf{1}\mathbf{1}^\top/n$ la matrice de centrage ($\mathbf{1}$ est un vecteur de 1 de dimension appropriée). Alors l'estimateur de HSIC est simplement $\widehat{H}_n = n^{-2}\mathrm{Tr}\left(\widetilde{\boldsymbol{K}}_x^n \widetilde{\boldsymbol{K}}_y^n\right)$ où $\tilde{\boldsymbol{K}} = \boldsymbol{C}_n \boldsymbol{K} \boldsymbol{C}_n$. $\boldsymbol{K}^n$ est la matrice de Gram de taille $n \times n$ obtenue après observation des n premières données. Comme $\mathrm{Tr}(\boldsymbol{ABC}) = \mathrm{Tr}(\boldsymbol{BCA})$ et $\boldsymbol{C}_n^2 = \boldsymbol{C}_n$, une forme plus simple pour $\tilde{\boldsymbol{K}}$ est $\tilde{\boldsymbol{K}} = \boldsymbol{K}\boldsymbol{C}_n$, définition que nous adoptons. Notons toutefois que $\tilde{\boldsymbol{K}} = \boldsymbol{K}\boldsymbol{C}_n$ n'est plus symétrique.

Pour obtenir une version récursive, les matrices de Gram sont partitionnées. Soient $\widetilde{\boldsymbol{k}}^n$ et $\widetilde{\boldsymbol{\ell}}^n$ deux vecteurs de dimension $n-1$, $\widetilde{\boldsymbol{K}}^{n-1}$ une matrice carrée de dimension $n-1$ et $\widetilde{\kappa}^n$ une constante. Alors

$$\widetilde{\boldsymbol{K}}_x^n \widetilde{\boldsymbol{K}}_y^n = \left(\begin{array}{c|c} \widetilde{\boldsymbol{K}}_x^{n-1} & \widetilde{\boldsymbol{k}}_x^n \\ \hline (\widetilde{\boldsymbol{\ell}}_x^n)^\top & \widetilde{\kappa}_x^n \end{array}\right)\left(\begin{array}{c|c} \widetilde{\boldsymbol{K}}_y^{n-1} & \widetilde{\boldsymbol{k}}_y^n \\ \hline (\widetilde{\boldsymbol{\ell}}_y^n)^\top & \widetilde{\kappa}_y^n \end{array}\right) \text{ et}$$

$$n^2 \widehat{H}_n = \left(\mathrm{Tr}\left(\widetilde{\boldsymbol{K}}_x^{n-1}\widetilde{\boldsymbol{K}}_y^{n-1}\right) + (\widetilde{\boldsymbol{\ell}}_x^n)^\top \widetilde{\boldsymbol{k}}_y^n + (\widetilde{\boldsymbol{\ell}}_y^n)^\top \widetilde{\boldsymbol{k}}_x^n + \widetilde{\kappa}_x^n \widetilde{\kappa}_y^n\right) \quad (1)$$

Nous cherchons alors une version récursive pour chacun des termes. Soient $\boldsymbol{k}_x^n$ le vecteur de dimension $(n-1)$ d'éléments $k_x(X_i, X_n), i = 1, \ldots, n-1$, $\kappa_x^n = k_x(X_n, X_n)$ (et les mêmes définitions en y) . Nous avons $\widetilde{\boldsymbol{K}}^n = \boldsymbol{K}^n \boldsymbol{C}_n$, matrices dont les partitions s'écrivent

$$\left(\begin{array}{c|c} \boldsymbol{K}^{n-1} & \boldsymbol{k}^n \\ \hline (\boldsymbol{k}^n)^\top & \kappa^n \end{array}\right)\left(\begin{array}{c|c} \boldsymbol{I}_{n-1} - \frac{1}{n}\mathbf{1}\mathbf{1}^T & -\frac{1}{n}\mathbf{1} \\ \hline -\frac{1}{n}\mathbf{1}^\top & 1 - \frac{1}{n} \end{array}\right)$$

Comme $\boldsymbol{I}_{n-1} - \frac{1}{n}\mathbf{1}\mathbf{1}^T = \boldsymbol{C}_{n-1} + \frac{1}{n(n-1)}\mathbf{1}\mathbf{1}^T$, on a par identification

$$\widetilde{\boldsymbol{K}}^{n-1} = \boldsymbol{K}^{n-1}\boldsymbol{C}_{n-1} - \frac{1}{n}\big(\boldsymbol{k}^n - \frac{1}{n-1}\boldsymbol{K}^{n-1}\mathbf{1}\big)\mathbf{1}^T \quad (2)$$

$$\widetilde{\boldsymbol{k}}^n = \boldsymbol{k}^n - \frac{1}{n}\big(\boldsymbol{k}^n + \frac{1}{n}\boldsymbol{K}^{n-1}\mathbf{1}\big) \quad (3)$$

$$\widetilde{\kappa}^n = \kappa^n - \frac{1}{n}\big(\kappa^n + \frac{1}{n}\mathbf{1}^\top \boldsymbol{k}^n\big) \quad (4)$$

$$\widetilde{\boldsymbol{\ell}}^n = \boldsymbol{k}^n - \frac{1}{n}\big(\kappa^n + \mathbf{1}^\top \boldsymbol{k}^n\big)\mathbf{1} \quad (5)$$

Posons $\boldsymbol{m}^n = \boldsymbol{K}^n\mathbf{1}$. En utilisant la matrice de Gram en partitions, $\boldsymbol{m}^n$, (3), (4) s'écrivent

$$\boldsymbol{m}^n = \left\{\begin{array}{l} \boldsymbol{m}^{n-1} + \boldsymbol{k}^n \\ \mathbf{1}^\top \boldsymbol{k}^n + \kappa^n \end{array}\right. ; \left(\begin{array}{c} \widetilde{\boldsymbol{k}}^n \\ \widetilde{\kappa}^n \end{array}\right) = \left(\begin{array}{c} \boldsymbol{k}^n \\ \kappa^n \end{array}\right) - \frac{1}{n}\boldsymbol{m}^n \quad (6)$$

qui avec (5), permet d'obtenir une récursion pour $(\widetilde{\boldsymbol{\ell}}_x^n)^\top \widetilde{\boldsymbol{k}}_y^n + (\widetilde{\boldsymbol{\ell}}_y^n)^\top \widetilde{\boldsymbol{k}}_x^n + \widetilde{\kappa}_x^n \widetilde{\kappa}_y^n$ dans (1). Le dernier terme est $\mathrm{Tr}\left(\widetilde{\boldsymbol{K}}_x^{n-1}\widetilde{\boldsymbol{K}}_y^{n-1}\right)$. Remarquons que $\boldsymbol{k}^n - \frac{1}{n-1}\boldsymbol{K}^{n-1}\mathbf{1}$ dans (2) est égal à $\boldsymbol{k}^n - \frac{1}{n-1}\boldsymbol{m}^{n-1}$ et peut être interprété comme une estimation de $\widetilde{\boldsymbol{k}}^n$ à partir des données passées (comparer à 6). On note ce terme $\widetilde{\boldsymbol{k}}^{n|n-1}$. Nous avons alors $\widetilde{\boldsymbol{K}}^{n-1} = \boldsymbol{K}^{n-1}\boldsymbol{C}_{n-1} - \frac{1}{n}\widetilde{\boldsymbol{k}}^{n|n-1}\mathbf{1}^T$, et par suite la trace de $\widetilde{\boldsymbol{K}}_x^{n-1}\widetilde{\boldsymbol{K}}_y^{n-1}$ s'écrit $\mathrm{Tr}\left(\boldsymbol{K}_x^{n-1}\boldsymbol{C}_{n-1}\boldsymbol{K}_y^{n-1}\boldsymbol{C}_{n-1}\right) - \frac{1}{n}\mathbf{1}^T\boldsymbol{K}_y^{n-1}\boldsymbol{C}_{n-1}\widetilde{\boldsymbol{k}}_x^{n|n-1} - \frac{1}{n}\mathbf{1}^T\boldsymbol{K}_x^{n-1}\boldsymbol{C}_{n-1}\widetilde{\boldsymbol{k}}_y^{n|n-1} + \frac{1}{n^2}\mathbf{1}^T\widetilde{\boldsymbol{k}}_x^{n|n-1}\mathbf{1}^T\widetilde{\boldsymbol{k}}_y^{n|n-1}$. En utilisant $\boldsymbol{m}^n = \boldsymbol{K}^n\mathbf{1}$ et quelques calculs, on obtient :

HSIC récursif :

$$\boldsymbol{m}_x^n = \left\{\begin{array}{l} \boldsymbol{m}_x^{n-1} + \boldsymbol{k}_x^n \\ (\boldsymbol{k}_x^n)^\top\mathbf{1} + \kappa_x^n \end{array}\right. ; \left(\begin{array}{c} \widetilde{\boldsymbol{k}}_x^n \\ \widetilde{\kappa}_x^n \end{array}\right) = \left(\begin{array}{c} \boldsymbol{k}_x^n \\ \kappa_x^n \end{array}\right) - \frac{1}{n}\boldsymbol{m}_x^n$$

$$\widetilde{\boldsymbol{\ell}}_x^n = \boldsymbol{k}_x^n - \frac{1}{n}\big(\kappa_x^n + \mathbf{1}^\top\boldsymbol{k}_x^n\big)\mathbf{1} \quad ; \quad \widetilde{\boldsymbol{k}}_x^{n|n-1} = \boldsymbol{k}_x^n - \frac{1}{n-1}\boldsymbol{m}_x^{n-1}$$

$$c_{xy}^{n|n-1} = \frac{1}{n}(\boldsymbol{m}_x^{n-1})^\top\widetilde{\boldsymbol{k}}_y^{n|n-1} - \frac{1}{n(n-1)}\mathbf{1}^\top\boldsymbol{m}_x^{n-1}\mathbf{1}^\top\widetilde{\boldsymbol{k}}_y^{n|n-1}$$

$$\widehat{H}_n = \widehat{H}_{n-1} + \frac{1}{n^2}\Big((\widetilde{\boldsymbol{\ell}}_x^n)^\top\widetilde{\boldsymbol{k}}_y^n + (\widetilde{\boldsymbol{\ell}}_y^n)^\top\widetilde{\boldsymbol{k}}_x^n + \widetilde{\kappa}_x^n\widetilde{\kappa}_y^n - c_{xy}^{n|n-1} - c_{yx}^{n|n-1} + \frac{1}{n^2}\mathbf{1}^\top\widetilde{\boldsymbol{k}}_x^{n|n-1}\mathbf{1}^\top\widetilde{\boldsymbol{k}}_y^{n|n-1} - (2n-1)\widehat{H}_{n-1}\Big)$$

Les vecteurs m, $\boldsymbol{k}$ voient leur dimension augmenter au cours du temps, rendant évidemment inutilisable cet algorithme de manière permanente. Cette difficulté est à l'origine de la deuxième forme récursive, qui se prête aisément à un calcul approché fondé sur une notion de parcimonie, offrant une alternative aux approximations de rang faible.

Deuxième forme et parcimonie. Pour réduire la taille mémoire et accélérer le calcul, nous avons recours à une stratégie de parcimonie, comme pour les algorithmes de régression en-ligne. L'idée force

est celle de [6] d'un dictionnaire cohérent $\mathcal{D}_n$. Ce dictionnaire est constitué de données passées et se construit récursivement. Intuitivement, une nouvelle donnée n'est incluse dans le dictionnaire que si elle ne peut être correctement estimée à partir des données déjà incluses.

Soit $\mathcal{D}_n$ un sous-ensemble de $\{1,\dots,n\}$, $\mathcal{D}_1=\{1\}$. Soit $Z_i=(X_i,Y_i)$. Le dictionnaire est mis à jour par la règle $\mathcal{D}_n=\mathcal{D}_{n-1}\cup\{n\} \iff \max_{\alpha\in\mathcal{D}_{n-1}}\left|k^c_{xy}(Z_\alpha,Z_n)\right|<\mu_0$ où $0<\mu_0\leq 1$, et k^c_{xy} est un noyau normalisé sur l'espace produit $\mathbb{X}\times\mathbb{Y}$ ($k^c(x,y)=k(x,y)(k(x,x)k(y,y))^{-1/2}$). Une donnée nouvelle est incluse dans le dictionnaire si elle n'est pas suffisamment cohérente avec les précédentes. Si $\mu=1$ les données sont systématiquement agrégées : on obtient alors HSIC usuel. La règle de cohérence simplifie le critère ALD (approximate linear dependence) proposé dans [1] qui repose sur la prédiction linéaire. Comme montré par Richard dans [6], pour $\mu_0<1$, la taille du dictionnaire reste finie si les données vivent dans un sous-espace compact. Ceci interdit l'utilisation de la technique pour des données non bornées (variables gaussiennes par exemple) puisque la taille du dictionnaire ne peut que croître. En pratique toutefois, l'approche peut être appliquée avec une grande confiance, la croissance du dictionnaire étant très lente pour des données raisonnables.

Pour appliquer le principe à un algorithme récursif, chaque mise à jour est précédée d'un test d'inclusion dans le dictionnaire. Dans le cas où une nouvelle donnée n'est pas incluse, nous ajoutons par rapport aux travaux de [6] une nouvelle information. Le dictionnaire réalise une quantification vectorielle des données. Pour des données à support compact, lorsque la taille finale du dictionnaire est atteinte, chaque membre du dictionnaire est le centre d'une cellule, appelée *cellule de cohérence*, dont la taille et la forme sont définies par le critère de cohérence. Une nouvelle donnée ne sera jamais incluse dans le dictionnaire puisque sa taille finale est atteinte. Par contre la nouvelle donnée appartient évidemment à une des cellules. Ne pas prendre en compte cette information provoque une perte d'information. Pour la minimiser, nous suggérons d'utiliser la mesure empirique des données calculée sur la partition induite par le critère de cohérence. Pour $\alpha\in\mathcal{D}_n$, on note $V_\alpha=\{Z,/\left|k^c_{xy}(Z_\alpha,Z)\right|\geq\mu_0\}$ la cellule de cohérence de centre Z_α. On pose $\pi_n(\alpha)=\sum_{k=1}^n \mathbf{1}_{V_\alpha}(Z_k)$.

Dans la deuxième version, nous allons utiliser le critère de cohérence pour ne garder qu'un nombre restreint d'éléments représentatifs des variables analysées. Pour obtenir la forme de l'algorithme, on ne travaille que sur les données conservées par le critère de cohérence : à une date n donnée, on n'utilise que les centres de cellules Z_α et la mesure empirique $\pi_n(\alpha)$. On peut alors établir un algorithme récursif exact sur ce jeu de données. Les estimateurs empiriques de $m_x(.)=E[k_x(.,X)]$ et $M_{yx}=E[k_x(.,X)\otimes k_y(.,Y)]$ s'écrivent sous la forme générique e^n

$$e^n=\frac{1}{n}\sum_{\alpha\in\mathcal{D}_n}\pi_n(\alpha)f(X_\alpha,Y_\alpha)=\frac{n-1}{n}e^{n-1}+\frac{1}{n}f(X_a,Y_a) \quad (7)$$

où $f(X,Y)=k_x(.,X)\otimes k_y(.,Y)$ pour $e=M_{yx}$ et $f(X,Y)=k_x(.,X)$ pour m_x. Rappelons que le produit tensoriel de deux vecteurs f,g est défini par $(f\otimes g)(u,v)=\langle u|f\rangle\langle v|g\rangle$. De plus, $a=n$ si la nouvelle donnée est incluse dans le dictionnaire, et $a=\arg\max_{\alpha\in\mathcal{D}_{n-1}}\left|k^c_{xy}(Z_\alpha,Z_n)\right|$ est l'indice de la cellule de cohérence de la nouvelle donnée si elle n'est pas incluse. En jouant avec les deux formes des estimateurs données en (7), on obtient un algorithme récursif. Il faut évaluer la norme HS de $C^n_{yx}=M^n_{yx}-m^n_y\otimes m^n_x$, c'est-à-dire calculer $\sum_l\left\langle C^n_{yx}\varphi_l\middle|C^n_{yx}\varphi_l\right\rangle_{\mathcal{H}_y}$ où $\{\varphi_l\}_{l\in\mathbb{N}}$ est une base orthonormée quelconque de $\mathcal{H}_x$. On obtient alors $\|C^n_{yx}\|^2=\|M^n_{yx}\|^2+\|m^n_x\|^2\|m^n_y\|^2-2c^n_{yx}$ où $c^n_{yx}=\left\langle M^n_{yx}m^n_x\middle|m^n_y\right\rangle$. Le calcul des normes est assez aisé, une fois remarquées les expressions $\left\langle M^{n-1}_{yx}k_x(.,X_a)\middle|k_y(.,Y_a)\right\rangle=\frac{1}{n-1}\boldsymbol{\pi}^\top_{n-1}\boldsymbol{k}^n_{xa}\circ\boldsymbol{k}^n_{ya}$ et $\left\langle m^{n-1}_x\middle|k_x(.,X_a)\right\rangle=\frac{1}{n-1}\boldsymbol{\pi}^\top_{n-1}\boldsymbol{k}^n_{xa}$, où $\boldsymbol{k}^n_{xa}$ est le vecteur contenant les $k_x(X_\alpha,X_a),\alpha\in\mathcal{D}_{n-1}$ et $\boldsymbol{\pi}_n$ le vecteur de dimension $|\mathcal{D}_n|$ contenant les $\pi_n(\alpha)$. Le produit scalaire c^n_{yx} est un peu plus délicat. On introduit $v^n_x(\beta)=\sum_{\gamma\in\mathcal{D}_n}\pi_n(\gamma)k_x(X_\gamma,X_\beta)$ pour $\beta\in\mathcal{D}_n$ et $\boldsymbol{v}^n_x$ le vecteur correspondant. On a alors $c^n_{yx}=n^{-3}\boldsymbol{\pi}^\top_n\boldsymbol{v}^n_x\circ\boldsymbol{v}^n_y$. Une version récursive de $v^n_x(\beta)$ est disponible, mais dépend de l'inclusion ou non de la nouvelle donnée dans le dictionnaire. Si $\mathcal{D}_n=\mathcal{D}_{n-1}\cup\{n\}$, la nouvelle donnée est $X_a=X_n$ puisqu'une nouvelle cellule est créée, le vecteur $\boldsymbol{\pi}_{n-1}$ voit sa dimension augmenter de 1 et on a la récursion $\boldsymbol{\pi}_n=(\boldsymbol{\pi}^\top_{n-1},\ 1)^\top$. Il vient $v^n_x(\beta)=\sum_{\gamma\in\mathcal{D}_{n-1}}\pi_{n-1}(\gamma)k_x(X_\gamma,X_\beta)+k_x(X_a,X_\beta)$, et $\boldsymbol{v}^n_x$ suit la récursion $\boldsymbol{v}^n_x=((\boldsymbol{v}^{n-1}_x+\boldsymbol{k}^n_{xa})^\top,\ \boldsymbol{\pi}^\top_{n-1}\boldsymbol{k}^n_{xa}+\kappa^n_{xa})^\top$, où $\kappa^n_{xa}=k_x(X_a,X_a)$. Si la nouvelle donnée n'est pas incluse dans le dictionnaire, elle est remplacée par le centre de sa cellule de cohérence, à savoir Z_a où $a=\arg\max_{\alpha\in\mathcal{D}_{n-1}}\left|k^c_{xy}(Z_\alpha,Z_n)\right|$. Dans ce cas $\mathcal{D}_n=\mathcal{D}_{n-1}$, le vecteur $\boldsymbol{\pi}_n$ reste égal $\boldsymbol{\pi}_{n-1}$ excepté pour l'élément $\pi_n(a)$ qui est incrémenté de 1, et on a $\boldsymbol{v}^n_x=\boldsymbol{v}^{n-1}_x+\boldsymbol{k}^n_{xa}$ qui ne change pas de dimension. En collectionnant les divers éléments on obtient une forme récursive exacte pour l'estimateur n'utilisant que la quantification proposée. D'ailleurs, les données arrivant sont classées dans leur cellule et remplacée par le centre a de cette cellule. Pour terminer et améliorer l'algorithme, il suffit alors non pas d'utiliser Z_a comme nouvelle donnée, mais simplement Z_n systématiquement. Dans ce qui précède, on remplace alors $\boldsymbol{k}_{xa}$ par $\boldsymbol{k}_x$, vecteur qui contient les $k_x(X_n,X_\alpha),\forall\alpha\in\mathcal{D}_{n-1}$. En résumé, on a obtient :

HSIC parcimonieux :

$$\widehat{H}^{\mu_0}_n=\|M^n_{yx}\|^2+\|m^n_x\|^2\|m^n_y\|^2-2c^n_{yx}$$
$$\|M^n_{yx}\|^2=\frac{(n-1)^2}{n^2}\|M^{n-1}_{yx}\|^2+\frac{2}{n^2}\boldsymbol{\pi}^\top_{n-1}\boldsymbol{k}^n_x\circ\boldsymbol{k}^n_y+\frac{\kappa^n_x\kappa^n_y}{n^2}$$
$$\|m^n_x\|^2=\frac{(n-1)^2}{n^2}\|m^{n-1}_x\|^2+\frac{2}{n^2}\boldsymbol{\pi}^\top_{n-1}\boldsymbol{k}^n_x+\frac{\kappa^n_x}{n^2}$$
$$c^n_{yx}=\frac{1}{n^3}\boldsymbol{\pi}^\top_n\boldsymbol{v}^n_x\circ\boldsymbol{v}^n_y \text{ où :}$$

1. Si $\mathcal{D}_n=\mathcal{D}_{n-1}\cup\{n\}$

$$\boldsymbol{v}^n_x=\begin{pmatrix}\boldsymbol{v}^{n-1}_x+\boldsymbol{k}^n_x\\ \boldsymbol{\pi}^\top_{n-1}\boldsymbol{k}^n_x+\kappa^n_x\end{pmatrix} \quad \text{et} \quad \boldsymbol{\pi}_n=\begin{pmatrix}\boldsymbol{\pi}_{n-1}\\ 1\end{pmatrix}$$

2. Si $\mathcal{D}_n=\mathcal{D}_{n-1}$: $a=\arg\max_{\alpha\in\mathcal{D}_{n-1}}\left|k^c_{xy}(Z_\alpha,Z_n)\right|$,

$$\boldsymbol{v}^n_x=\boldsymbol{v}^{n-1}_x+\boldsymbol{k}^n_x \quad \text{et} \quad \boldsymbol{\pi}_n=\boldsymbol{\pi}_{n-1}+\delta_{a\alpha}$$

Dans ces expressions, on a $\kappa^n_x=k_x(X_n,X_n)$, $\boldsymbol{\pi}_n$ le vecteur de dimension $|\mathcal{D}_n|$ contenant les $\pi_n(\alpha)$ (initialisé par $\pi_1(1)=1$).

Une version adaptative. La version adaptative s'obtient de manière analoque que la forme précédente, mais repose sur une moyenne exponentielle.

HSIC adaptatif parcimonieux :

$$\widehat{H}^{\mu_0}_n=\|M^n_{yx}\|^2+\|m^n_x\|^2\|m^n_y\|^2-2c^n_{yx}$$
$$\|M^n_{yx}\|^2=(1-\gamma)^2\|M^{n-1}_{yx}\|^2+2\gamma(1-\gamma)\boldsymbol{\pi}^{\gamma\top}_{n-1}\boldsymbol{k}^n_x\circ\boldsymbol{k}^n_y+\gamma^2\kappa^n_x\kappa^n_y$$
$$\|m^n_x\|^2=(1-\mu)^2\|m^{n-1}_x\|^2+2\mu(1-\mu)\boldsymbol{\pi}^{\mu\top}_{n-1}\boldsymbol{k}^n_x+\mu^2\kappa^n_x$$
$$c^n_{yx}=\boldsymbol{\pi}^{\gamma\top}_n\boldsymbol{v}^\mu_{x,n}\circ\boldsymbol{v}^\zeta_{y,n} \text{ où :}$$

HSIC parcimonieux (suite) :

1. Si $\mathcal{D}_n = \mathcal{D}_{n-1} \cup \{n\}$

$$\boldsymbol{v}^{\mu}_{x,n} = \begin{pmatrix} (1-\mu)\boldsymbol{v}^{\mu}_{x,n-1} + \mu \boldsymbol{k}^n_x \\ (1-\mu)\boldsymbol{\pi}^{\mu\,\top}_{n-1}\boldsymbol{k}^n_x + \mu\kappa^n_x \end{pmatrix} \text{ et } \boldsymbol{\pi}^{\eta}_n = \begin{pmatrix} (1-\eta)\boldsymbol{\pi}^{\eta}_n \\ \eta \end{pmatrix}$$

2. Si $\mathcal{D}_n = \mathcal{D}_{n-1}$: $a = \arg\max_{\alpha \in \mathcal{D}_{n-1}} \left| k^c_{xy}(Z_\alpha, Z_n) \right|$,

$$\boldsymbol{v}^{\mu}_{x,n} = (1-\mu)\boldsymbol{v}^{\mu}_{x,n-1} + \mu \boldsymbol{k}^n_x \text{ et } \boldsymbol{\pi}^{\eta}_n = (1-\eta)\boldsymbol{\pi}^{\eta}_{n-1} + \eta\delta_{a\alpha}$$

où $\eta = \gamma, \mu$ ou ζ. Les expressions en x sont valables pour y en remplaçant x par y et μ par ζ. Les vecteurs $\boldsymbol{\pi}^{\eta}_n$ dépendent des facteurs d'oublis $\eta = \gamma, \mu$ ou ζ, et réalisent en quelque sorte une estimation adaptative des mesures de probabilités des variables X, Y et du couple.

3 Illustrations

Pour illustrer le comportement des algorithmes et la perte de performance limitée de HSIC parcimonieux, nous travaillons sur des données bidimensionnelles : la première composante X étant distribuée suivant une gaussienne centrée réduite, et la deuxième Y suivant une loi exponentielle bilatérale de paramètre 1. X et Y sont indépendantes, et la dépendance est obtenue par une rotation. Les statistiques suivantes sont calculées par méthode Monte-Carlo en utilisant 500 réalisations des processus, pour des tailles d'échantillons de $n = 3000$ (sauf les temps de calculs qui sont évalués jusque n=10^4 échantillons en moyennant sur 100 réalisations). L'ensemble des résultats est synthétisé par la figure (1). Nous utilisons le noyau gaussien $\exp(-\|x-y\|^2/1.2)$ pour lequel la valeur 1.2 est choisie empiriquement. HSIC parcimonieux est évalué pour $\mu_0 = 0.85, 0.9, 0.95$. La figure A montre des traces typiques et un zoom après convergence, qui illustre la faible perte de performance en fonction de μ_0 quand on compare HSIC parcimonieux à HSIC. La perte décroît avec μ_0 croissant, ce qui est attendu puisque $\mu_0 = 1$ correspond à HSIC usuel. Pour quantifier cette perte, on trace l'erreur quadratique moyenne (la valeur vraie de HSIC étant inconnue, on choisit la valeur de HSIC estimée). Au delà de la confirmation d'une convergence au taux usuel $1/n$, la figure (1B) montre encore une fois une très faible perte pour $\mu_0 = 0.95$. Ceci est confirmé dans l'insert où nous montrons les valeurs finales moyennes estimées μ_0 de 0.85 à 1, ainsi que des intervalles $\pm 2\sigma$ (de confiance à 95 % sous hypothèse gaussienne). Les moyennes sont très proches de la moyenne de HSIC, et les variances très comparables.

Ces observations prennent toute leur importance avec C et D. En C, nous traçons les temps de calculs (en unités arbitraires) en fonction de n pour $\mu_0 = 0.85, 0.9, 0.95, 0.99$ et 1 (HSIC). En D nous traçons la taille moyenne des dictionnaires obtenus en fonction de μ_0 (et les intervalles $\pm 2\sigma$). Pour $\mu_0 = .99$, le gain en temps de calcul est comparable à HSIC (même plus mauvais pour les n petits). Mais dès $\mu_0 \leq 0.95$, les temps sont divisés par au moins 2 pour des tailles de données très grandes. De plus, en examinant D, on voit que la la taille du dictionnaire est très réduite, puisque le taux de compression est de 25 pour $\mu_0 = 0.95$!

Pour illustrer l'algorithme adaptatif, on l'applique pour détecter un saut brutal dans la structure de dépendance et pour suivre une évolution continue de cette structure. Dans le premier cas, les variables aléatoire utilisées précédemment deviennent soudainement indépendantes, puis à nouveau dépendantes avec une rotation toutefois différente. Pour le deuxième cas, l'angle de rotation suit une courbe gaussienne sur l'intervalle d'observation, son maximum étant atteint au milieu de l'intervalle. La figure (1E) illustre la poursuite, et la figure (1F) le comportement pour les ruptures. Les facteurs d'oublis choisis sont $\gamma = 0.05, \mu = \zeta = 0.1$ et le paramètre de cohérence $\mu_0 = 0.95$. Ces figures illustrent les comportements usuels de l'adaptivité : compromis vitesse de convergence-variance, retard en poursuite. L'algorithme reste à être étudié.

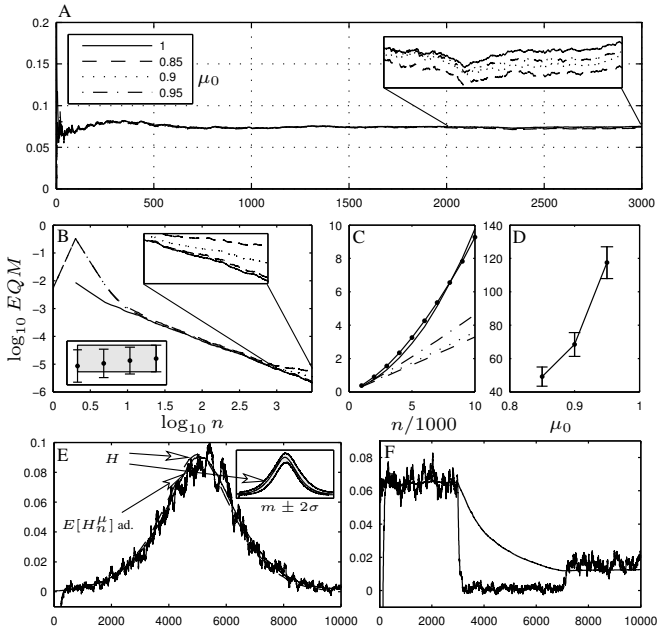

FIGURE 1 – A : traces typiques pour différentes valeurs de μ_0 et un zoom. B : EQM en échelle loglog et un zoom pour différentes valeurs de μ_0. L'insert bas montre la moyenne finale $\pm 2\sigma$ pour les trois valeurs de μ_0 et HSIC ($\mu_0 = 1$). C : temps de calcul (u. a.). Les points épais correspondent à $\mu_0 = 0.99$. D : taille moyenne $\pm 2\sigma$ du dictionnaire pour $n = 3000$. E : Illustration en poursuite. Insert : bande moyene $\pm 2\sigma$ et paramètre H à poursuivre. F : Versions adaptative et récursive face à des changements brusques.

Références

[1] Y. Engel, S. Mannor, and R. Meir. The kernel recursive least-squares algorithm. *IEEE Trans. on Signal Processing*, 52(8) :2275–2284, 2004.

[2] S. Fine and K. Scheinberg. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2 :243–264, 2001.

[3] R. Fortet. *Vecteurs, fonctions et distributions aléatoires dans les espaces de Hilbert* . Hermès, 1995.

[4] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf. Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6 :2075–2129, 2005.

[5] A. Rényi. On measures of dependence. *Acta Math. Acad. Sci. Hungar.*, 10(2) :441–451, september 1959.

[6] C. Richard, J. C. M. Bermudez, and P. Honeine. Online prediction of time series data with kernels. *IEEE Trans. on Signal Processing*, 57(3) :1058–1067, Mar 2009.

[7] B. Schölkopf and A. J. Smola. *Learning with kernels*. MIT Press, Cambridge, Ma, USA, 2002.

[8] L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13 :1393–1434, 2012.

[9] I. Steinwart and A. Christmann. *Support vector machines*. Springer, 2008.

[10] R. Vicente, M. Wibral, M. Lindner, and G. Pipa. Transfer entropy–a model-free measure of effective connectivity for the neurosciences. *Journal of Computational Neuroscience*, 30(1) :45–67, 2011.