LAL 14-24

UNIVERSITÉ PARIS-SUD

ÉCOLE DOCTORALE D'INFORMATIQUE

THÈSE DE DOCTORAT

Soutenue le 20 Février 2014 par

Djalel BENBOUZID

Sequential prediction for budgeted learning

Applications to trigger design

Discipline : Informatique
Préparée au Laboratoire de l'Accélérateur Linéaire de l'Université (Paris-Sud XI), dans les équipes APPSTAT et TAO

sous la direction de Balázs KÉGL

Jury

Rapporteurs	Ludovic DENOYER	Université Paris VI
	Kilian WEINBERGER	Washington University in St.Louis
Directeur	Balázs KÉGL	Université Paris-Sud & CNRS
Examinateurs	Florence D'ALCHÉ-BUC	Université d'Evry-Val d'Essonne
	Damien ERNST	Université de Liège
	Vladimir GLIGOROV	CERN
	Michèle SÉBAG	Université Paris-Sud & CNRS
Invité	Guy WORMSER	Université Paris-Sud & CNRS

Last modified on February 28, 2014

Prédiction Séquentielle pour l'Apprentissage Budgeté
Application à la Conception de *Trigger*

Djalel Benbouzid

Introduction

Cette thèse aborde le problème de la classification sous un nouvel angle, en y incorporant une dimension séquentielle. Dans la classification séquentielle, la prédiction ne consiste plus en une opération "atomique" exploitant l'ensemble des attributs d'une observation donnée, à l'instar des algorithmes classiques tels que les Séparateurs à Vaste Marge[1] (SVM) ou les réseaux de neurones, mais consiste en une série d'opérations simplifiées, portant sur un sous-ensemble d'attributs et pouvant se terminer prématurément.

Nous nous intéressons, en particulier, au cas de l'apprentissage à contraintes de budget (ou apprentissage *budgété*) où l'objectif est de concevoir un "classifieur" qui, tout en apportant des prédictions correctes, doit gérer un budget computationnel qui est consommé au fur et à mesure que les différents attributs sont acquis ou évalués. Les attributs peuvent avoir des coûts d'acquisition différents et il arrive souvent que les attributs les plus discriminatifs soient les plus coûteux. Le diagnostic médical et le classement de pages web sont des exemples typiques d'applications de l'apprentissage budgété. Pour le premier, l'objectif est de limiter le nombre et le coût des tests médicaux que le patient doit endurer et, pour le second, le classement doit se faire dans un temps assez court pour ne pas faire fuir l'usager.

Notre principale motivation réside dans la conception de classifieurs qui répondent à des contraintes computationnelles, telles que la vitesse de classification, ou à des contraintes de budget plus complexes et plus atypiques où le coût d'acquisition d'une variable varie en fonction de plusieurs facteurs, comme c'est le cas dans la conception de déclencheurs (ou *trigger*) en physique des particules.

Les triggers sont un type de classifieurs rapides, temps-réel et sensibles aux coûts qui ont pour objectif de filtrer les données massives que les accélérateurs de particules produisent et d'en retenir les événements les plus susceptibles de contenir le phénomène physique étudié, avant d'être enregistrés pour des analyses ultérieures. La conception de trigger impose des contraintes computationnelles strictes lors de la classification mais, surtout, exhibe des schémas complexes de calcul du coût de chaque attributs. Certains attributs sont dépendants d'autres attributs et nécessitent de calculer ces derniers en amont, ce qui a pour effet d'augmenter le coût de la classification. De plus, le coût des attributs peut directement dépendre de leur valeur concrète. On retrouve ce cas de figure lorsque les extracteurs d'attributs améliorent la qualité de leur sortie avec le temps mais peuvent toujours apporter des résultats préliminaires. Enfin, les observations sont

[1] La traduction littérale du nom anglais d'origine, Support Vector Machines, serait Machine à Vecteur de Support.

regroupées en *sacs* et, au sein du même sac, certaines observations partagent le calcul d'un sous-ensemble d'attributs. Toutes ces contraintes nous ont amenés à formaliser la classification sous un angle séquentiel.

Répondant à la nécessité de concevoir des classifieurs précis et, dans le même temps, assez rapides pour être utilisés en temps réel et dans des environnements embarqués, des algorithmes de classification séquentielle ont été proposés, notamment à travers la famille des classifieurs en cascade, cependant, là où ces derniers se limitent, le plus souvent, à une réponse ad-hoc à la problématique de départ, l'approche proposée dans ce manuscrit se veut plus formelle en combinant apprentissage supervisé et apprentissage par renforcement; l'apprentissage supervisé apportant des garanties en terme de propriétés de généralisation et l'apprentissage par renforcement offrant un cadre de formalisation pour les problème de prises de décisions séquentielles.

De part la motivation de départ et les nombreuses similitudes avec les cascades de classifieurs, notre approche peut être considérée comme une généralisation du principe de cascade, cependant, elle constitue dans le même temps une contribution à une famille de modèles émergents visant à aborder le problème de prédiction à travers une approche intrinsèquement séquentielle.

Cette thèse se veut une contribution au domaine de l'informatique et à l'apprentissage statistique en particulier, cependant, du fait de l'environnement scientifique dans lequel ces travaux ont été effectués ainsi que leur motivation première, s'inscrivant directement dans le domaine de la physique des particules, nous avons taché de rendre ce manuscrit accessible à la communauté de la physique expérimentale.

Description des chapitres

Le manuscrit s'articule autour de 5 chapitres, les chapitres 3 et 4 en présentent les principales contributions. Le chapitre 3 introduit le principal algorithme de cette thèse, MDDAG (Benbouzid et al., 2012), acronyme anglais pour Graphe Acyclique Orienté de Décisions Markoviennes. Le chapitre 4 illustre l'application de cet algorithme à la conception de déclencheurs. Dans ce chapitre, une brève description de la problématique en physique des particules est décrite ainsi la formalisation des cas atypiques de classification avec contraintes de budget qui en résultent.

Chapitre 1

Le premier chapitre introduit succinctement le domaine de l'apprentissage statistique ainsi que ses principaux paradigmes. Très sommairement, l'apprentissage statistique, dit aussi apprentissage artificiel ou automatique, fait référence à l'étude et l'analyse des algorithmes avec pour entrée un ensemble de données et ayant pour objectif d'inférer des inconnus à partir des données connues. On y distingue généralement trois principaux paradigmes, selon le type de données en entrée, à savoir,

- l'apprentissage supervisé,
- l'apprentissage non supervisé
- et l'apprentissage par renforcement.

L'objectif en apprentissage supervisé est d'inféré -ou d'apprendre- une fonction d'un espace $\mathcal{X}$ (souvent, $\mathcal{X} = \mathbb{R}^d$) vers un espace $\mathcal{Y}$. Selon que $\mathcal{Y}$ soit un espace discret ou continue, nous parlerons respectivement de classification ou de régression. Les données en entrée se présentent sous forme de n couples $(\mathbf{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}, i = 1, \ldots, n$. Le label y_i, *i.e*, la "bonne réponse" de chaque observation $\mathbf{x}_i$ fait donc partie des données en entrée, dites données d'entraînement, et l'on cherche à apprendre une fonction $f(\mathbf{x}) \in \mathcal{Y}$ qui *généralise* la prédiction à des observations dont on ne connaît pas le label.

Lorsque l'on n'a pas accès à des données avec labels, on parle d'apprentissage non supervisé. L'estimation de densité, la réduction de dimensionnalité, la séparation des données en clusters sont des applications types d'apprentissage non supervisé.

Enfin, l'apprentissage par renforcement s'intéresse à l'apprentissage par interaction avec un environnement. L'objectif est d'apprendre la politique optimale d'un agent qui interagit avec un environnement à travers un ensemble d'actions. L'environnement renvoi à l'agent une évaluation de chaque action à un instant donné, à travers une valeur qu'on appelle le *gain*[2], mais en aucun cas ne l'informe directement sur l'optimalité ou non de l'action effectuée. Ainsi, l'agent apprend par exploration/exploitation en maximisant l'espérance de la sommes des gains obtenus durant un épisode, c-à-d, au long de la séquence de décisions.

Chapitre 2

Le deuxième chapitre fait l'état de l'art d'une famille d'algorithmes d'apprentissage supervisé, celle des méthodes de *boosting*. Le principe du boosting consiste à combiner des classificateurs dits "faibles", ayant un taux d'erreur à peine inférieur à celui d'une réponse aléatoire, pour former un classifieur global (ou classifieur fort) avec un taux d'erreur arbitrairement faible.

Historiquement, le premier algorithme de boosting à exhiber une complexité polynomiale fût ADABOOST (Freund and Schapire, 1997). Depuis, l'algorithme a été exhaustivement étudié, autant théoriquement qu'empiriquement, et a fait l'objet de nombreuses applications en apprentissage statistique (Schapire, 2003; Meir and Rätsch, 2003).

Les algorithmes de boosting sont particulièrement adéquat lorsqu'il s'agit d'imposer des contraintes computationnelles à la prédiction. En effet, leur nature additive permet de contrôler avec précision le nombre de classifieurs faibles à combiner et donc la complexité du classifieur final. De plus, la complexité du classifieur faible peut aussi être indépendamment contrôlée.

[2]On parlera de *coût* si l'on cherche à pénaliser l'action de l'agent avec une valeur négative.

Ces propriétés intéressantes en matière de contrôle explicite de la complexité du classifieur final ne sont pas étrangères au fait que les algorithmes de boosting aient contribué au premier détecteur de visage en temps réel, implémenté sous forme de classifieurs en cascade Viola and Jones (2001).

Une cascade de classifieurs est une série de classifieurs, ou étapes, à complexité croissante et politique d'arrêt prématuré. À la fin de chaque étape, une partie des observations est classifiée et le reste est envoyé à l'étape suivante pour une classification plus complexe. Ainsi, les observations "faciles" sont classifiées dans les premières étapes avec une faible complexité et les observations plus difficiles peuvent atteindre des étapes plus complexes. Les dernières étapes, ne recevant qu'un petit sous-ensemble des observations peuvent être arbitrairement complexes. Une architecture en cascade permet de traiter des problèmes à temps réel ou à données massives et, de ce fait, on trouve une assez vaste littérature visant à améliorer les algorithmes de constructions de cascades. Nous faisons une revue de littérature des différentes approches qui ont été proposées dans le cadre des cascades de classifieurs.

Le chapitre 2 introduit aussi MULTIBOOST, un cadriciel auquel nous avons contribué et qui implémente des algorithmes de boosting de manière modulaire, de sorte à rendre l'implémentation de nouveaux algorithmes relativement aisée.

Notre contribution principale, présentée au chapitre 3, s'inspire des cascades de classifieurs et en partage les motivations initiales en terme de classification rapide ainsi que l'aspect séquentiel de l'inférence. Cependant, nous abordons une approche différente dans la construction du classifieur séquentiel en le traduisant directement comme un problème d'apprentissage par renforcement. Ceci nous permet de palier les difficultés liées à l'apprentissage des cascades classiques tout en apportant une niveau de flexibilité qui permet d'adapter notre approche à des problèmes sous contraintes de budget plus complexes.

Chapitre 3

Ce chapitre présente la contribution principale de la thèse, à savoir un cadriciel pour la classification séquentielle sous contrainte de budget. Le chapitre commence par introduire les classifieurs "parcimonieux au cas par cas" (*instance-dependent sparsity*) qui consiste en la sélection d'un sous-ensemble d'attributs à exploiter en fonction de l'observation en entrée. Cela contraste avec la définition des classifieurs parcimonieux classiques qui, une fois le modèle parcimonieux trouvé, l'appliquent à toutes les observations de manière équivalente. Un tel modèle présente une trivialité en terme de minimisation de l'erreur empirique. Cependant, en imposant une contrainte de séquentialité, le problème n'est plus trivial et l'apprentissage consiste à séquentiellement prendre des décision de classification en fonction de l'instance courante mais aussi de décisions passées. Une fois le cadriciel présenté, nous introduisons MDDAG, une instanciation de cette approche qui aborde particulièrement les problèmes de classification rapide et/ou à contraintes de budget.

MDDAG traduit l'apprentissage en un processus de décision Markovien où l'objectif est d'apprendre la politique optimal d'un agent ayant pour entrée une séquence de classifieur et qui itère sur cette séquence afin d'en sélectionner un sous-ensemble à évaluer pour une observation donnée. Les classifieurs peuvent être binaires ou multi-classes, hétérogènes actant sur des attributs continus ou discrets, et peuvent aussi être des classifieurs faibles. Dans la première partie expérimentales, d'ailleurs, les classifieurs sont obtenus grâce à ADABOOST.MH, une extension multi-classe de ADABOOST. Dans l'application à la conception de déclencheurs au chapitre 4, chaque classifieur est un classifieur fort, aussi obtenu avec ADABOOST.MH. Pour chaque classifieur en entrée, l'agent choisit une action parmi les trois suivantes,

- l'évaluer, ce qui aura pour conséquence d'en avoir la prédiction,
- le négliger, et donc sauter directement au classifieur suivant,
- ou arrêter la classification et faire une prédiction en fonction des classifieurs évalués tout au long de l'épisode.

Nous proposons aussi une définition de l'espace des états de l'agent tractable, évitant ainsi l'espace de départ dont les variables croissent avec le nombre de classifieurs en entrées et le nombre de classes.

Dans ce chapitre, nous faisons aussi une revenue de littérature des modèles séquentiels appliqués à la prédiction. En effet, ce type d'approche connaît un intérêt particulier depuis quelques années. C'est pourquoi, nous proposons une mise en perspective et une taxonomie des différentes approches en mettant en exergue leurs points de rapprochement et de divergence.

Enfin, ce chapitre se termine par une partie expérimentale, mettant MDDAG en application sur divers problèmes standards en apprentissage statistique. Les résultats montrent un gain consistent en matière de performance dès lors que la complexité du classifieur finale doit être bornée. Il arrive aussi que MDDAG obtienne les mêmes performances que la combinaison englobant tous les classifieurs en entrée, mais avec une complexité inférieure.

Chapitre 4

Dans ce chapitre, nous appliquons MDDAG à un problème concret d'apprentissage à contraintes de budget, à savoir, la conception de déclencheurs en physique des particules, notre motivation initiale. Nous utilisons des données de l'expérimentation LHCb[3]

Nous présentons d'abord les simples modifications qu'il suffit d'apporter au cadriciel de départ afin de satisfaire les différentes contraintes qu'impose la conception de déclencheurs. Contrairement au problèmes budgétés classiques, où chaque attribut est associé à un coût d'acquisition constant, le coût des attributs auxquels un déclencheur a accès dépendent de plusieurs facteurs, que nous pouvons distinguer en trois types,

[3]Les données proviennent de simulation simplifiées.

- les attributs avec un coût immédiat, correspondant aux cas classique,
- les attributs dont le coût dépend de leur valeur actuelle. Cela résulte dans le cas de l'expérience LHCb du fait que certaines particules énergétiques nécessitent moins de temps de calcul que d'autres particules, moins énergétiques. D'une façon générale, ce cas se produit lorsqu'un attribut résulte d'un traitement anytime, i.e, un algorithme qui améliore ses résultats avec le temps mais qui peut à chaque instant délivrer des résultats intermédiaires.
- Les attributs dont le coût dépend d'un sous-ensemble d'observation. Ceci résulte du fait que les observations arrivent en forme de sacs d'observation qui partagent certains calculs d'attributs. En d'autres termes, si deux observations appartenant au même sac partage un attribut, il suffit de calculer la valeur de cet attribut une seule pour les deux observations.

À ces contraintes atypique s'ajoute un dernier niveau de complexité dans le calcul des coûts: les attributs sont interdépendants. Afin de calculer un attribut, il faut, en amont, calculer tous les attributs dont il dépend et, par conséquent, accumuler le coût d'acquisition de tous les attributs calculés. (Figure 4.1 dans le manuscrit).

Nous montrons donc que MDDAG permet très simplement de prendre en compte toutes ces contraintes complexes à travers des modifications mineures de l'algorithme.

Enfin, nous proposons aussi une visualisation du classifieur final. Ceci a pour avantage de clairement montrer la nature sous forme de graphe de ce dernier. Bien que les classifieurs en entrée soient parcourus séquentiellement, les "chemins" que peuvent prendre les différentes observations forment un graphe acyclique orienté et montre, par la même occasion, la nature "data-dependent" de notre approche. C'est en effet cette propriété qui permet de réduire drastiquement la complexité moyenne de la classification tout en gardant une précision raisonnable.

Chapitre 5

Le dernier chapitre clôt la dissertation en mettant en relief les différentes connexions que peut avoir l'approche proposée avec différents domaines de l'apprentissage statistique. En effet, MDDAG aborde l'apprentissage supervisé sous un angle d'apprentissage par renforcement, de plus, le classifieur final étant parcimonieux permet rapproche notre algorithmes des méthodes parcimonieuses. Enfin, la nature en graphe et "data-dependent" soulève d'intéressantes questions en matière de représentation des données et de codification. Le chapitre présente aussi différentes perspectives et directions de recherche dans les modèles séquentiels pour la classification.

References

Benbouzid, D., Busa-Fekete, R., and Kégl, B. (2012). Fast classification using sparse decision DAGs. In *Proceedings of the 29th International Conference on Machine Learning*.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139.

Meir, R. and Rätsch, G. (2003). An introduction to boosting and leveraging. *Advanced lectures on machine learning*, pages 118–183.

Schapire, R. E. (2003). The boosting approach to machine learning: An overview. In *Nonlinear estimation and classification*, pages 149–171. Springer.

Viola, P. and Jones, M. (2001). Fast and robust classification using asymmetric adaboost and a detector cascade. *Proc. of NIPS01*.