

Apprentissage incrémental par sélection de données dans un flux pour une application de sécurité routière

Saunier N., Midenet S., Grumbach A.

Actes de Conférence Cap'04, Montpellier, France, in : Conférence d'Apprentissage Cap 2004, Liquiere M. et Sebban M. (Eds), Presses Universitaires de Grenoble (PUG), pp 239-251, 2004

---

# Apprentissage incrémental par sélection de données dans un flux pour une application de sécurité routière \*

Nicolas Saunier<sup>1,2</sup>, Sophie Midenet<sup>2</sup>, Alain Grumbach<sup>1</sup>

<sup>1</sup> Département Informatique et Réseaux, Ecole Nationale Supérieure des  
Télécommunications,  
46 rue Barrault, 75013 Paris.  
{saunier, grumbach}@infres.enst.fr

<sup>2</sup> Unité de Recherche Génie des Réseaux de Transport et Informatique Avancée,  
Institut National de la Recherche sur les Transports et leur Sécurité,  
2 avenue du Général Malleret-Joinville, 94114 Arcueil  
{saunier, midenet}@inrets.fr

**Résumé** : Cet article s'inscrit dans le contexte applicatif de l'évaluation de stratégies de gestion de feux de circulation. L'évaluation s'appuie sur l'estimation de la sévérité d'interactions entre véhicules. Des véhicules sont en interaction s'ils sont proches dans le temps et l'espace. La sévérité de leur interaction augmente avec leur proximité spatio-temporelle. Cet article décrit comment nous avons traité le problème de l'apprentissage de la sévérité à partir d'images du carrefour.

Nous disposons d'une base de données constituées d'images vidéo et de mesures d'occupation surfacique du carrefour extraites de ces images. Les données sont bruitées et complexes, et la seule manière d'évaluer la sévérité est de confier cette tâche à un expert humain.

Dans ce contexte, nous avons développé une approche qui s'appuie sur un «apprentissage actif». Les caractéristiques des données et de leur traitement évoquées ci-dessus nous ont amenés à prolonger l'apprentissage actif dans les directions suivantes : flux séquentiel de données, possibilité pour l'expert d'étiquetage flou et pour l'apprenant de sélectionner les exemples d'apprentissage, et combinaison d'hypothèses par vote.

Après une présentation détaillée du contexte justifiant notre démarche (partie 1), nous décrivons l'approche à travers les algorithmes principaux mis en oeuvre (partie 2), puis nous proposons une évaluation à partir de résultats expérimentaux (partie 3).

**Mots-clés** : Apprentissage Incrémental, Sélection de Données, Flux de Données.

---

\*Le travail présenté dans cet article fait partie d'une thèse cofinancée par la région Ile de France.

# 1 Introduction

Le travail décrit dans cet article répond aux besoins d'une application réelle. Afin d'analyser la sécurité dans les carrefours à feux, nous développons de nouveaux indicateurs de risques pour les usagers (voir (Saunier *et al.*, 2003) pour plus de détails). Nous détectons automatiquement certaines catégories d'interactions entre véhicules. Des véhicules sont en interaction s'ils sont «proches» dans le temps et l'espace. Nous voulons estimer la sévérité de ces interactions, sévérité définie comme la distance entre l'interaction et l'accident potentiel. Il existe de nombreuses méthodes de calcul de la sévérité, fondées sur des grandeurs mesurables comme des distances entre véhicules et des vitesses, et le jugement calibré d'experts humains (van der Horst, 1990). Cet article traite de l'apprentissage automatique de la sévérité à partir d'images du trafic sur l'intersection.

Les données d'entrée proviennent de mesures de l'occupation surfacique du carrefour, fournies par plusieurs capteurs vidéo. Elles sont bruitées et complexes (cf figure 1). Le trafic et les données d'un carrefour réel ont été enregistrés pendant 8 mois. Il n'existe pas encore à notre connaissance de système automatique d'estimation des indicateurs de sévérité. Dans notre application, seul un expert humain peut fournir les étiquettes de sévérité en observant les séquences vidéo correspondant aux données. Du fait de la contrainte de navigation dans des bandes vidéo VHS, nous accédons aux données séquentiellement. De cette façon, nous pourrions aussi utiliser les méthodes développées dans un cadre temps réel.

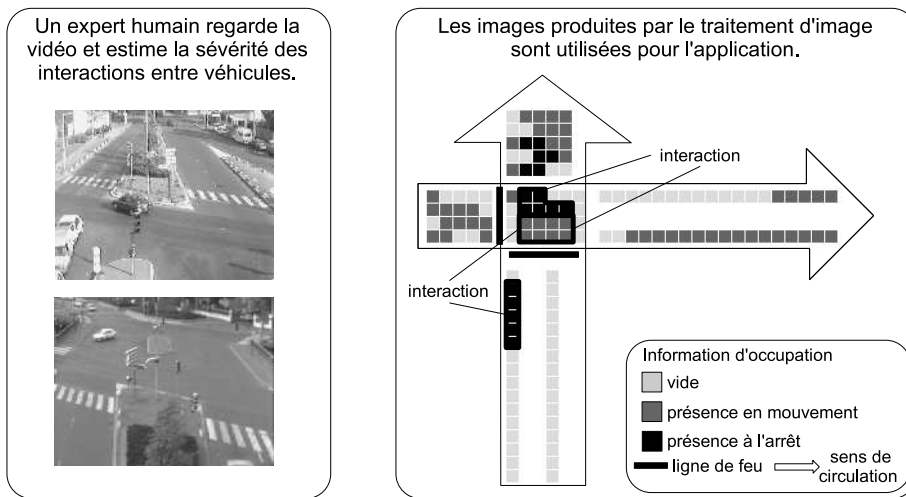


FIG. 1 – Images vidéo et données en entrée, contenant 2 interactions détectées. Les données sont simplifiées pour être lisibles en noir et blanc. Chaque unité peut en fait prendre 6 états, au lieu des 3 représentés dans la figure : la présence en mouvement dans la figure couvre les états trace rapide de présence, début de présence, fin de présence et présence en mouvement.

Pour nommer les entités en jeu dans ce problème, nous nous inspirons de (Cornuéjols *et al.*, 2002). Trois entités interagissent, l'*expert* (ou oracle, superviseur, professeur), l'*apprenant* et la ou les *hypothèse(s)* apprise(s). L'apprenant est un système automatique capable de construire une hypothèse selon un algorithme donné, i.e. une fonction approchant autant que possible le concept-cible, ici la sévérité d'interactions entre véhicules. Dans le cadre de l'apprentissage classique, l'apprenant reçoit passivement des exemples du concept-cible étiquetés par un expert (cf figure 2).



FIG. 2 – L'apprenant est passif dans un problème d'apprentissage classique.

Pour juger la sévérité, l'expert humain ne peut distinguer que quelques niveaux discretisés. À côté des erreurs inhérentes à l'étiquetage par un expert humain (Yan *et al.*, 2003), la tâche est compliquée pour des exemples proches des frontières entre les classes. Il semble plus juste de raisonner avec des frontières graduelles. Nous pouvons utiliser des fonctions d'appartenance pour décrire les classes identifiées par l'expert. Afin d'éviter que l'expert ne choisisse arbitrairement une classe lorsqu'il hésite, il peut étiqueter l'exemple comme appartenant de façon égale aux classes concernées : nous parlons dans ce cas d'«étiquette floue».

En outre, après avoir étudié et étiqueté une partie des données, nous avons remarqué que les données des différentes classes se chevauchent : deux exemples peuvent être très proches et pourtant étiquetés différemment sans hésitation par l'expert. Il faut aussi noter que les tailles des classes ne sont pas équilibrées. Ces caractéristiques des données et de l'étiquetage constituent un problème d'apprentissage difficile. Les performances sont mauvaises lors d'un apprentissage classique «batch», dans lequel l'apprenant passif utilise un ensemble fini de données d'apprentissage pour construire une hypothèse.

L'apprentissage batch utilise un ensemble d'apprentissage choisi aléatoirement parmi les données, en faisant généralement l'hypothèse simplificatrice d'une distribution indépendante et identique des données (Cornuéjols *et al.*, 2002; Duda & Hart, 2000). Dans le cadre non-batch, i.e. incrémental, l'apprentissage se fait en plusieurs étapes, en utilisant des ensembles de données d'apprentissage différents. Dans le cas d'un flux de données, l'apprenant peut mettre à jour l'hypothèse à chaque nouvelle donnée. L'apprentissage incrémental permet de faire une sélection «intelligente» des données, sans avoir besoin de déterminer la distribution de ces données. Afin d'améliorer les résultats de notre application, nous voulons sélectionner les données qui apportent le plus d'information, pour en particulier préciser les frontières des classes proches. Pour atteindre ce but, nous laissons le contrôle du processus d'apprentissage à l'apprenant pour construire incrémentalement une hypothèse. Cette définition rentre dans le cadre de l'apprentissage actif.

Les recherches dans le domaine de l'apprentissage se sont principalement concentrées sur des méthodes dans lesquelles l'apprenant interagit avec l'expert et peut lui poser



FIG. 3 – Un apprenant actif peut poser des questions à l’expert.

des questions, par exemple lui demander d’étiqueter certains exemples (protocole de requête d’appartenance, «membership queries») (cf figure 3).

La plupart des algorithmes d’apprentissage actif travaillent dans le cadre fondé sur un réservoir de données («pool-based») (Yan *et al.*, 2003; Baram *et al.*, 2003; Tong, 2001; Roy & McCallum, 2001; Iyengar *et al.*, 2000; Abe & Mamitsuka, 1998; Freund *et al.*, 1997; Seung *et al.*, 1992; Lewis & Catlett, 1994), dans lequel un ensemble fixe d’exemples non-étiquetés et complètement accessibles est présenté à l’apprenant. A chaque étape, l’apprenant choisit un ou plusieurs exemples dans le réservoir de données pour qu’ils soient étiquetés. L’expert les étiquette et l’apprenant construit une hypothèse à partir de tous les exemples étiquetés jusque là. Dans notre application, les données sont accessibles séquentiellement, ce qui nous conduit à considérer une variante, fondée sur un flux continu de données non-étiquetées («stream-based»), étudiée dans (Freund *et al.*, 1997; Cesa-Bianchi *et al.*, 2003). A l’arrivée de chaque exemple, l’apprenant doit choisir s’il demande à l’expert de l’étiqueter. Par ailleurs, nous introduisons par rapport aux algorithmes d’apprentissage actif existants la possibilité de ne pas utiliser en apprentissage les données étiquetées par l’expert. Cette possibilité n’est pas particulière au cadre avec flux de données. Les différences entre ces deux cadres sont résumées dans la figure 4.

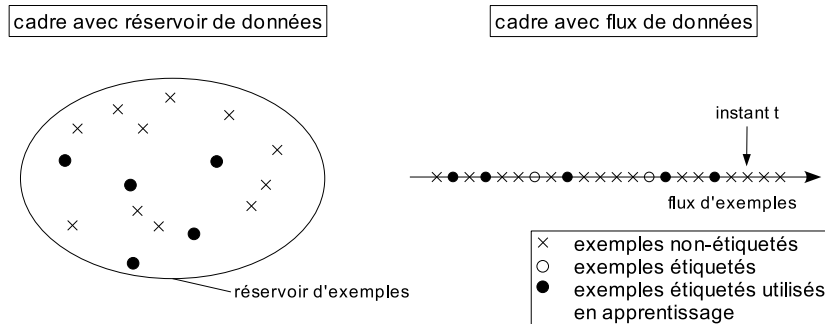


FIG. 4 – A l’instant  $t$ , l’apprenant peut sélectionner *n’importe quel* exemple dans le cadre avec réservoir de données, alors que dans le cadre avec flux de données, l’apprenant est limité à l’exemple *courant*.

Nous travaillons dans le cadre de l’apprentissage actif dans la mesure où nous sélectionnons un sous-ensemble du flux des données pour les utiliser en apprentissage et

augmenter les performances du système. Nous ne cherchons pas à réduire le nombre de données étiquetées, mais plutôt à améliorer les performances de l'apprenant à nombre d'exemples étiquetés constant. Ne pas utiliser toutes les données étiquetées en apprentissage peut sembler contre-intuitif. Nous montrons cependant que cela peut apporter des gains pour une application dans laquelle les classes se chevauchent.

De plus, ces méthodes d'apprentissage incrémental permettent d'envisager l'obtention de résultats intermédiaires (algorithme «anytime») pendant l'apprentissage et fournissent des méthodes de mise à jour d'une hypothèse dans le cas d'évolution temporelle de la cible d'apprentissage. Nous pensons explorer ces directions dans des développements ultérieurs de ce travail. Dans la suite, nous décrivons l'approche à travers les algorithmes principaux mis en oeuvre (partie 2), puis nous proposons une évaluation à partir de résultats expérimentaux (partie 3).

## 2 Algorithmes incrémentaux de sélection de données dans un flux

### 2.1 Description générique

---

**Algorithme 1:** Algorithme incrémental de sélection de données.

---

**Entrées :** Soit  $A$  l'algorithme d'apprentissage utilisé  
 Soit  $x_t \in X$  l'exemple disponible à l'instant  $t$   
 Soit  $y_t \in Y$  la classe (inconnue) de  $x_t$   
 Soit  $F = [x_{1 \leq t \leq N}]$  un flux d'exemples  
 Soit  $E_i = \{(x_t, y_t)\}$  l'ensemble d'apprentissage après la sélection de  $i$  exemples étiquetés  
 Soit  $h_i : X \rightarrow Y$  l'hypothèse apprise selon  $A$  sur  $E_i$

**Sorties :** Hypothèse finale  $h_{finale}$ .

**début**

Etiqueter  $t_0$  exemples de  $F$  et les mettre dans  $E_0$

Appliquer  $A$  à  $E_0$  pour construire  $h_0$

$t = t_0 + 1$ ;  $i = 0$

**répéter**

**si**  $CritèreSélection(x_t, h_i, E_i)$  satisfait **alors**

$i = i + 1$

$E_i = E_{i-1} \cup \{(x_t, y_t)\}$  ( $y_t$  est obtenue dans  $CritèreSélection$ )

Appliquer  $A$  à  $E_i$  pour construire  $h_i$

$t = t + 1$

**jusqu'à**  $CritèreArrêt$

**fin**

---

Le principe générique des algorithmes incrémentaux de sélection des données dans un flux est décrit dans l'algorithme 1. Le coeur de ce type d'algorithme est le critère de sélection des données. Une première possibilité consiste à sélectionner une première

fois les données avant leur étiquetage. Une idée naturelle dans ce cas serait d'adapter dans le cadre incrémental les idées des algorithmes d'apprentissage actif avec réservoir de données en ajoutant un seuil. En effet ces algorithmes essaient d'optimiser différentes mesures. L'échantillonnage par incertitude («*Uncertainty Sampling*») sélectionne les exemples pour lesquels l'hypothèse courante a le moins de certitude (Lewis & Catlett, 1994). L'algorithme requête par comité («*Query-by-Committee*») sélectionne les exemples qui maximisent le désaccord parmi un comité d'hypothèses (Abe & Mamitsuka, 1998; Freund *et al.*, 1997; Seung *et al.*, 1992). D'autres algorithmes ont pour but de réduire la taille de l'espace des versions (Schohn & Cohn, 2000; Tong, 2001). Aucune de ces méthodes n'optimise directement la métrique selon laquelle l'apprenant sera finalement évalué, l'espérance de l'erreur de l'apprenant sur un futur ensemble de test, qui peut être estimée en utilisant les prédictions de l'hypothèse courante (Roy & McCallum, 2001).

Dans ces algorithmes d'apprentissage actif avec réservoir de données sont sélectionnés l'exemple ou les exemples minimisant ou maximisant ces mesures. L'adaptation de ces critères dans le cadre d'un flux de données est direct en ajoutant un seuil : par exemple, sélectionner l'exemple courant  $x_t$  si la certitude de l'hypothèse courante dans sa prédiction est en-dessous d'une certaine valeur. Nous préférons éviter ce type de méthodes car l'ajustement de tels seuils présente des problèmes de robustesse. Il est aussi possible de sélectionner les données une fois qu'elles sont étiquetées. Nous proposons dans la partie suivante un critère fondé sur les exemples mal-classés.

Outre le critère de sélection des données, le critère d'arrêt et la construction de l'hypothèse finale sont importants. Ces deux éléments sont liés. Si la qualité des hypothèses  $h_i$  apprise sur l'ensemble d'apprentissage  $E_i$  est bonne, le critère d'arrêt n'est pas crucial et l'hypothèse finale est une des hypothèses  $h_i$ . Si ce n'est pas le cas, il faut trouver soit un bon critère d'arrêt, soit une méthode pour construire des hypothèses finales de bonne qualité à partir des nombreuses hypothèses  $h_i$ . Nous privilégions dans ce travail l'amélioration de la qualité de l'hypothèse finale par combinaison des  $h_i$ . Ce point est discuté plus loin.

## 2.2 Critère de sélection des données

Tout d'abord, les exemples auxquels l'expert attribue une étiquette fautive ne sont pas utilisés dans le travail présenté dans cet article. Ils peuvent appartenir à plus d'une classe et apportent donc moins d'information que ceux appartenant clairement à une classe.

Nous nous sommes concentrés sur l'amélioration des performances à ensemble de données étiquetées constant plutôt que sur la réduction du nombre d'exemples étiquetés. Nous avons choisi de demander les étiquettes de tous les exemples à l'expert, et de sélectionner les données mal-classées par l'hypothèse courante (cf algorithme 2). L'apprentissage sur les données mal-classées a été étudié dans le cadre de l'apprentissage batch supervisé avec la technique de *Windowing* (Fürnkranz, 1998). Les auteurs attribuent l'intérêt d'apprendre sur les données mal-classées à la déformation de la distribution réelle des données, et évoquent la possibilité d'améliorer les performances avec ce type de sélection de données d'apprentissage. C'est aussi le but visé par les techniques de *Boosting* (Freund, 1995; Schapire, 1990) qui ré-échantillonnent plusieurs

fois l'ensemble des exemples en se concentrant sur les plus difficiles à apprendre. Notre travail suit les mêmes principes que le Boosting, et peut se voir comme une extension de la technique de Windowing pour traiter un flux de données. Il y a aussi eu des tentatives dans le cadre de l'apprentissage actif avec réservoir de données (Iyengar *et al.*, 2000), à ceci près que les vraies étiquettes ne sont pas connues. A leur place sont utilisées les prédictions d'une hypothèse apprise sur les exemples étiquetés.

Nous sélectionnons donc les exemples mal-classés par l'hypothèse courante pour déformer la distribution réelle des données vers les zones difficiles, i.e. les zones dans lesquelles des exemples de différentes classes sont très proches, et les zones inconnues. Une hypothèse apprise sur un ensemble dans lequel sont sur-représentés les exemples situés dans les zones frontières doit permettre de mieux préciser les frontières entre classes qu'une hypothèse apprise sur l'ensemble des données. Cela implique de n'utiliser qu'une partie des données étiquetées.

---

**Algorithme 2:** Critère de sélection des données.

---

**Entrées :** Soit  $(x, y) \in X \times Y$  le couple de l'exemple disponible et sa classe (inconnue)  
 Soit  $Y_{expert} = Y \cup \{\text{étiquettes floues}\}$  ensemble des étiquettes que l'expert peut attribuer à  $x$   
 Soit  $E = \{(x, y)\}$  l'ensemble courant de données étiquetées pour l'apprentissage  
 Soit  $h : X \rightarrow Y$  l'hypothèse courante apprise sur  $E$

**Sorties :** Sélection de  $x$ .

**début**

Demander à l'expert d'étiqueter  $x$  : soit  $y_{expert}$  cette étiquette

**si**  $y_{expert} \in Y$  et  $h(x) \neq y_{expert}$  **alors**

Retourner *vrai*

**sinon**

Retourner *faux*

**fin**

---

### 2.3 Critère d'arrêt et hypothèse finale

Une fois qu'un ensemble de données étiquetées a été sélectionné, comment construire l'hypothèse finale susceptible d'être utilisée ensuite en exploitation, dont la performance sera mesurée sur un ensemble de test indépendant ? La plupart des algorithmes d'apprentissage actif avec réservoir de données choisissent l'une des hypothèses apprises durant le processus d'apprentissage actif. Puisque leurs performances augmentent avec la taille de l'ensemble d'apprentissage, ce choix est raisonnable. Dans ce cas, le critère d'arrêt n'est pas crucial. Les auteurs représentent la croissance des performances en fonction du nombre de données étiquetées, et mettent en avant la réduction du nombre d'exemples nécessaires pour atteindre un niveau de performance donné. Le critère d'arrêt et les méthodes pour estimer les performances réelles d'une hypothèse, telle qu'elle peut être mesurée sur un ensemble de test indépendant, sont abordés dans la lit-



térature. (Schohn & Cohn, 2000; Baram *et al.*, 2003) mettent en évidence le fait que la performance réelle d'une hypothèse n'est pas disponible. Même des méthodes standards comme la validation croisée ou «leave-one-out» ne permettent pas d'obtenir une estimation fiable de l'erreur réelle d'une hypothèse obtenue par apprentissage actif, car la distribution de l'ensemble d'apprentissage est biaisée par le processus de l'apprentissage actif vers les exemples difficiles à apprendre, différente de la vraie distribution des données. (Baram *et al.*, 2003) introduit un estimateur empirique, de maximisation de l'entropie de classification, et (Schohn & Cohn, 2000) utilise des caractéristiques des SVM, la famille d'hypothèses utilisée, à savoir le nombre d'exemples devenant vecteurs de support.

---

**Algorithme 3:** Algorithme incrémental de sélection des données fondé sur les exemples mal-classés, avec le critère de sélection des données décrit dans l'algorithme 2 et les notations de l'algorithme 1.

---

**Entrées :** Soit  $Vote_{i_1, i_2}(h_j)$  l'hypothèse obtenue par vote majoritaire parmi les hypothèses de  $\{h_j \mid i_1 \leq j \leq i_2\}$   
 Soit  $n$  le nombre d'hypothèses combinées par vote

**Sorties :** Hypothèse finale  $h_{finale} = Vote_{max(0, i-n), i}(h_j)$ .

**début**

Étiqueter  $t_0$  exemples de  $F$  et les mettre dans  $E_0$

Appliquer  $A$  à  $E_0$  pour construire  $h_0$

$t = t_0 + 1$ ;  $i = 0$

**répéter**

**si**  $CritèreSélection(x_t, Vote_{max(0, i-n), i}(h_j), E_i)$  **satisfait alors**

$i = i + 1$

$E_i = E_{i-1} \cup \{(x_t, y_t)\}$  ( $y_t$  est obtenue dans  $CritèreSélection$ )

Appliquer  $A$  à  $E_i$  pour construire  $h_i$

$t = t + 1$

**jusqu'à**  $CritèreArrêt$

**fin**

---

Cependant, dans le cas où les performances ne croissent pas simplement en fonction du nombre de données sélectionnées dans l'ensemble d'apprentissage, un bon critère d'arrêt ou une méthode pour construire une meilleure hypothèse finale est nécessaire. Comme nous venons de le voir, évaluer la performance de l'hypothèse courante durant le processus d'apprentissage est un problème difficile. Une solution naturelle serait d'utiliser un ensemble de validation indépendant. Nous privilégions l'amélioration de la qualité des hypothèses produites, qualité qui devrait être croissante en fonction du nombre de données ajoutées. Une solution est fournie par les méthodes de combinaisons d'hypothèses. (Schapire *et al.*, 1997) montre que ce type de méthode est plus robuste et précis. Par exemple, les méthodes de *Bagging* et de *Boosting* comme *Adaboost* sont utilisées en conjonction avec des requêtes par comité dans (Abe & Mamitsuka, 1998). Dans notre cas, de nombreuses hypothèses sont apprises durant le processus, et peuvent être combinées pour former l'hypothèse finale. Nous suggérons de la former en tant que *Vote* majoritaire sur les prédictions de tout ou partie des hypothèses apprises

durant le processus d'apprentissage. Nous utilisons les dernières hypothèses apprises car elles sont apprises sur le plus grand nombre de données d'apprentissage. Pour les mêmes raisons de robustesse, l'hypothèse utilisée dans le critère de sélection est une combinaison par Vote des hypothèses précédemment apprises. La nouvelle version de l'algorithme incrémental de sélection des données est présentée dans l'algorithme 3. Le choix du nombre d'hypothèses combinées par Vote  $n$  sera discuté dans le cadre de l'application.

### 3 Evaluation sur des données réelles

#### 3.1 Données expérimentales

Nous avons étudié des algorithmes incrémentaux de sélection des données d'apprentissage dans un flux dans un contexte applicatif précis d'évaluation automatique de la sévérité des interactions entre véhicules dans une intersection. Dans le cadre de cet article, nous travaillons précisément sur un indicateur de sévérité lié à la vitesse, mais l'approche développée est directement applicable à tout indicateur de sévérité.

Nous disposons d'une grande base de données constituée durant une expérience de 8 mois sur un site réel. Les données d'entrée du système proviennent de mesures de l'occupation spatiale d'une intersection. Sa surface est discrétisée et chaque unité discrète indique la présence et la dynamique des véhicules sur la période d'une seconde. Ces unités ont 6 états possibles (cf figure 1). Les vecteurs d'entrée composés de ces unités, i.e. l'image de l'intersection, ont des tailles différentes selon l'intersection. Pour ce travail, nous avons utilisé les données provenant d'une intersection, et les vecteurs d'entrée sont de taille 80 unités. L'expert a identifié trois niveaux de vitesse, minimale, moyenne et maximale (MIN, MOY, et MAX), et deux frontières graduelles entre les classes MIN et MOY, MOY et MAX, correspondant respectivement à des étiquettes floues «MIN-MOY» et «MOY-MAX». La classe MOY est nettement plus nombreuse que les autres et la classe MAX est la plus difficile à discriminer. Cette classe des interactions les plus sévères est la plus intéressante pour notre application de sécurité routière.

Nous avons fait les évaluations présentées dans la suite avec trois ensembles différents. Tout d'abord nous avons étiqueté complètement une séquence de données  $F$ , d'environ 30 minutes, fournissant 510 exemples, parmi lesquels 103 ont des étiquettes floues et ne sont pas utilisés pour l'apprentissage. Nous avons construit un ensemble de test composé de 4 séquences de 10 minutes, sous des conditions de trafic différentes, 2 en trafic dense et 2 en trafic fluide, fournissant 541 exemples, parmi lesquels 170 ont des étiquettes floues et ne sont pas utilisés en test. Enfin, nous tirons au hasard dans un troisième ensemble 3 exemples, un de chaque classe, pour l'ensemble initial d'apprentissage  $E_0$ . Il s'agit donc d'un apprentissage local par rapport à l'ensemble de la base de données acquises sur 8 mois. D'autres techniques, en particulier de suivi des performances, devront être envisagées pour étendre ce travail à l'ensemble des données.

Nous utilisons un classifieur bayésien naïf comme hypothèse car ses performances en apprentissage batch sont les meilleures parmi les modèles testés, même si l'hypothèse d'indépendance des attributs n'est pas respectée. Son algorithme d'apprentissage est rapide et incrémental. Nous travaillons avec la boîte à outils Weka (Witten & Frank,

2000).

### 3.2 Résultats expérimentaux

Nous avons d'abord travaillé sur l'ajustement du paramètre  $n$  (cf algorithme 3). Des tests nous ont montrés que les performances sont assez sensibles à la valeur de  $n$ . Nous avons choisi  $n = 7$ , et remarqué que les performances se dégradent lorsque l'on s'éloigne de cette valeur.

Nous comparons les performances de l'hypothèse construite par l'algorithme 3 à chaque instant  $t$ , appelé MC (sélection des données Mal-Classées), aux performances d'hypothèses apprises sur des exemples de l'ensemble  $F_t = \{x_{t'} \in F | t' \leq t\}$ , ensemble des exemples vus jusqu'à  $t$  :

«**algorithme**» **BATCH** : l'hypothèse est apprise simplement sur  $F_t$ ,

«**algorithme**» **BAGGING** l'hypothèse est apprise par Bagging sur  $F_t$ , à base de  $n$  classifi eurs bayésiens naïfs appris chacun sur un sous-ensemble d'exemples tiré aléatoirement dans  $F_t$ , de même taille que l'ensemble  $E_i$  des données sélectionnées par MC,

«**algorithme**» **BATCH-EQ** : l'hypothèse est apprise sur un sous-ensemble de  $F_t$  dont la distribution des classes est équilibrée,

«**algorithme**» **BATCH-EQ-MC** : l'hypothèse est apprise sur un sous-ensemble de  $F_t$  dont la distribution des classes est la même que celle de  $E_i$ .

En comparant BATCH et notre algorithme MC, nous voulons montrer qu'il est possible d'avoir de meilleures performances en sélectionnant un sous-ensemble des données pour l'apprentissage. BAGGING construit une agrégation d'hypothèses comparable à notre hypothèse finale. Nous voulons ainsi montrer que MC sélectionne effectivement des données plus informantes que ce qui pourrait être obtenu par échantillonnage aléatoire. Nous voulons aussi analyser le fonctionnement de MC, en particulier préciser si son fonctionnement est fondé sur un ré-équilibrage de la distribution des classes. Nous le comparons pour cela à BATCH-EQ dans lequel on égalise simplement les proportions des classes, et à BATCH-EQ-MC, qui n'est pas un algorithme à proprement parler puisqu'il copie la distribution des classes de l'ensemble d'apprentissage construit par MC.

Nous traçons l'évolution du pourcentage d'exemples bien classés au cours de l'apprentissage, en fonction du nombre d'exemples sélectionnés par MC, pour les algorithmes MC et ses concurrents, BATCH et BAGGING, dans la figure 5. Notre algorithme a la meilleure performance parmi les algorithmes testés pour le dernier tiers de l'apprentissage (au-delà de 80 exemples). En revanche, ses performances sont moins bonnes pour un faible nombre de données d'apprentissage que celle des autres algorithmes. Il est intéressant de noter l'augmentation quasi-monotone de la performance de notre algorithme après un démarrage très variable. Comme notre but est d'améliorer les performances pour toutes les classes, nous présentons aussi les résultats pour la classe MAX dans la figure 5. Le taux d'exemples bien classés par l'hypothèse apprise par notre algorithme est meilleur que celui des hypothèses produites par les autres al-

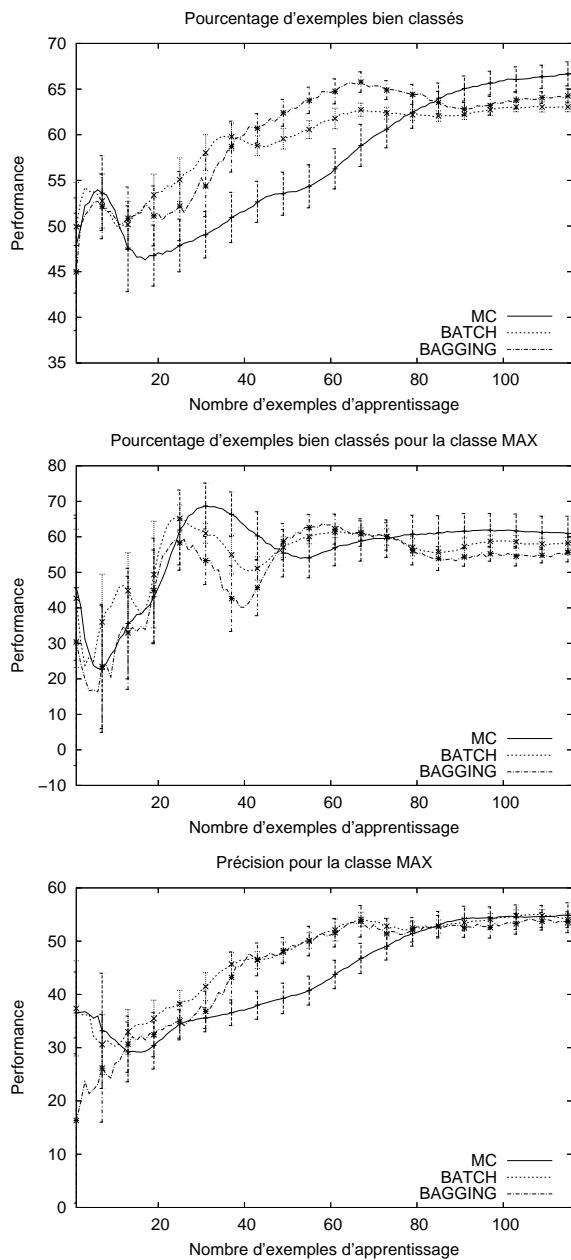


FIG. 5 – Courbes d'apprentissage des 3 algorithmes MC, BATCH et BAGGING en fonction du nombre d'exemples sélectionnés par MC ; **Haut** : Pourcentage d'exemples correctement classés pour les 3 algorithmes ; **Milieu** : Pourcentage des exemples de la classe MAX correctement classés pour les 3 algorithmes ; **Bas** : Précision par rapport à la classe MAX pour les 3 algorithmes. Toutes ces grandeurs sont mesurées sur l'ensemble de test, moyenné sur 50 essais, avec des barres d'écart-type (espacées pour la lisibilité).

algorithmes pour le dernier tiers de l'apprentissage. La précision<sup>1</sup> est aussi la meilleure, mais plus proche des autres algorithmes. Notre algorithme MC montre de bonnes performances à partir d'un certain nombre d'exemples sélectionnés.

	BATCH	BAGGING	MC
Exemples bien classés	63,3 ± 0,4	65,9 ± 0,7	<b>67,3 ± 1,6</b>
Exemples bien classés (MIN)	<b>83,1 ± 0,8</b>	82,5 ± 1,0	74,9 ± 2,5
Précision (MIN)	51,2 ± 0,7	56,6 ± 1,5	<b>64,9 ± 3,3</b>
Exemples bien classés (MOY)	59,5 ± 0,4	65,4 ± 1,0	<b>69,0 ± 2,4</b>
Précision (MOY)	<b>76,4 ± 0,4</b>	75,2 ± 0,8	75,4 ± 1,8
Exemples bien classés (MAX)	58,9 ± 1,1	56,5 ± 1,9	<b>59,2 ± 4,3</b>
Précision (MAX)	55,2 ± 0,5	<b>57,6 ± 1,3</b>	55,1 ± 2,8

	BATCH-EQ	BATCH-EQ-MC	MC
Exemples bien classés	58,5 ± 2,2	63,3 ± 1,1	<b>67,3 ± 1,6</b>
Exemples bien classés (MIN)	<b>84,2 ± 1,7</b>	82,8 ± 1,7	74,9 ± 2,5
Précision (MIN)	47,3 ± 2,2	53,0 ± 1,9	<b>64,9 ± 3,3</b>
Exemples bien classés (MOY)	48,0 ± 4,5	56,7 ± 1,8	<b>69,0 ± 2,4</b>
Précision (MOY)	<b>78,0 ± 2,9</b>	78,0 ± 1,1	75,4 ± 1,8
Exemples bien classés (MAX)	64,0 ± 4,9	<b>64,7 ± 2,2</b>	59,2 ± 4,3
Précision (MAX)	49,4 ± 2,9	53,7 ± 1,5	<b>55,1 ± 2,8</b>

TAB. 1 – Performances (en pourcentages) à la fin de la séquence  $F$ , globales et pour chaque classe (moyennée sur 50 essais avec initialisation aléatoire de  $E_0$ ). Le meilleur score de chaque ligne apparaît en gras. **Haut** : Comparaison de MC avec les algorithmes concurrents BATCH et BAGGING ; **Bas** : Comparaison de MC avec BATCH-EQ et BATCH-EQ-MC

Nous présentons aussi dans le tableau 1 les performances finales des hypothèses produites par les cinq algorithmes. Pour cela, nous laissons le processus d'apprentissage épuiser le flux  $F$ . Le nombre moyen d'exemples sélectionnés est de  $136 \pm 5$ , parmi 407 exemples vus. Dans les deux tableaux, notre algorithme MC a le meilleur taux de classification global, et la meilleure performance dans 3 autres critères par classe. Il est proche du meilleur pour 2 autres, et nettement moins bon dans un cas (pourcentage d'exemples bien classés de la classe MIN). Cependant, les autres algorithmes sont meilleurs dans ce cas car ils négligent la précision pour cette classe, bien plus faible que celle de notre algorithme. La tâche reste difficile, mais nous observons que notre algorithme MC a des gains sur presque tous les critères par rapport à ses concurrents. Nous montrons en particulier qu'il est possible d'améliorer les performances en sélectionnant efficacement un sous-ensemble des données. Equilibrer simplement la distribution des classes est une idée insuffisante, et la distorsion de la distribution des classes n'ex-

<sup>1</sup>La précision d'une hypothèse sur un ensemble de test pour une classe est définie comme le nombre d'exemples de cette classe correctement classés, sur le nombre d'exemples classés par l'hypothèse dans cette classe. Ce critère est complémentaire du taux d'exemples de la classe correctement classés, car ce dernier peut être très élevé, mais la précision très faible si l'hypothèse classe tous les exemples dans la classe.

plique pas seule les performances de notre algorithme. Tous les exemples n'apportent pas autant d'information et un critère de sélection comme le notre donne des résultats intéressants.

## 4 Conclusion

Nous avons présenté dans cet article des algorithmes incrémentaux pour la sélection de données d'apprentissage dans un flux de données. Nous avons proposé un algorithme générique, puis une solution fondée sur les données mal-classées. Nous suggérons de combiner des hypothèses apprises pendant le processus pour des raisons de robustesse et de performance.

Nous avons testé notre algorithme sur une application réelle, l'évaluation de la sévérité des interactions entre véhicules dans une intersection. Nous montrons que notre algorithme améliore globalement les performances, générales et par classe.

Nous pensons explorer maintenant l'importance de l'initialisation de l'ensemble d'apprentissage. Nous envisageons d'utiliser en apprentissage les exemples auquel l'expert attribue une étiquette floue, ainsi que la métrique des classes de sévérité. Nous sommes également intéressés par d'autres méthodes de combinaison d'hypothèses, comme le Boosting dont les principes sont proches de ceux de notre algorithme.

## Références

- ABE N. & MAMITSUKA H. (1998). Query learning strategies using boosting and bagging. In *Proceedings of the Fifteenth International Conference on Machine Learning*.
- BARAM Y., EL-YANIV R. & LUZ K. (2003). Online choice of active learning algorithms. In *Proc. of The Twentieth International Conference on Machine Learning*, p. 19–26.
- CESA-BIANCHI N., CONCONI A. & GENTILE C. (2003). Learning probabilistic linear-threshold classifiers via selective sampling. *Lecture Notes in Computer Science*, **2777**, 373–387.
- CORNUÉJOLS A., MICLET L. & KODRATOFF Y. (2002). *Apprentissage Artificiel*. Eyrolles.
- DUDA R. O. & HART P. E. (2000). *Pattern Classification*. Wiley-Interscience.
- FREUND Y. (1995). Boosting a weak learning algorithm by majority. *Information and Computation*, **121**(2), 256–285.
- FREUND Y., SEUNG H. S., SHAMIR E. & TISHBY N. (1997). Selective sampling using the query by committee algorithm. *Machine Learning*, **28**(2-3), 133–168.
- FÜRNKRANZ J. (1998). Integrative windowing. *Journal of Artificial Intelligence Research*, **8**, 129–164.
- IYENGAR V. S., APTE C. & ZHANG T. (2000). Active learning using adaptive resampling. In *Sixth ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, p. 92–98.
- LEWIS D. D. & CATLETT J. (1994). Heterogeneous uncertainty sampling for supervised learning. In W. W. COHEN & H. HIRSH, Eds., *Proceedings of ICML-94, 11th International Conference on Machine Learning*, p. 148–156, New Brunswick, US : Morgan Kaufmann Publishers, San Francisco, US.

- ROY N. & MCCALLUM A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning*, p. 441–448 : Morgan Kaufmann, San Francisco, CA.
- SAUNIER N., MIDENET S. & GRUMBACH A. (2003). Automatic detection of vehicle interactions in a signalized intersection. In *16th International Cooperation on Theories and Concepts in Traffic Safety Workshop*, Soesterberg, Holland.
- SCHAPIRE R. E. (1990). The strength of weak learnability. *Machine Learning*, **5**(2), 197–227.
- SCHAPIRE R. E., FREUND Y., BARTLETT P. & LEE W. S. (1997). Boosting the margin : a new explanation for the effectiveness of voting methods. In *Proc. 14th International Conference on Machine Learning*, p. 322–330 : Morgan Kaufmann.
- SCHOHN G. & COHN D. (2000). Less is more : Active learning with support vector machines. In *Proc. 17th International Conf. on Machine Learning*, p. 839–846 : Morgan Kaufmann, San Francisco, CA.
- SEUNG H. S., OPPER M. & SOMPOLINSKY H. (1992). Query by committee. In *Computational Learning Theory*, p. 287–294.
- TONG S. (2001). *Active Learning : Theory and Applications*. PhD thesis, Department of Computer Science of Stanford University.
- VAN DER HORST R. (1990). *A time-based analysis of road user behavior in normal and critical encounter*. PhD thesis, Delft University of Technology.
- WITTEN I. H. & FRANK E. (2000). *Data Mining : Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco.
- YAN R., YANG J. & HAUPTMANN A. (2003). Automatically labelling video data using multi-class active learning. In *Proceedings. Ninth IEEE International Conference on Computer Vision, 2003*, p. 516–523.