2018

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale MathSTIC

présentée par

Yann Dauxais

préparée à l'unité de recherche UMR 6074 IRISA
Institut de recherche en informatique et systèmes aléatoires
UFR Informatique et électronique (ISTIC)

Extraction de chroniques discriminantes

Thèse soutenue à Rennes
le 13 avril 2018

devant le jury composé de :

Sandra BRINGAY
Professeur à l'université Paul Valéry/rapporteur

Panagiotis PAPAPETROU
Professor at Stockholm University/rapporteur

Florent MASSEGLIA
CR Inria Montpellier/examinateur

David GROSS-AMBLARD
Professeur à l'université de Rennes 1 / directeur de thèse

Thomas GUYET
MC agrocampus-ouest/co-directeur de thèse

André HAPPE
Ingénieur CHU Rennes/examinateur

Résumé en français

Jan 19 12:12:52 DEBUG repo: using cache for: fedora
Jan 19 12:12:52 DEBUG not found updateinfo for: Fedora 25 - x86_64
Jan 19 12:12:53 DEBUG repo: using cache for: rpmfusion-free-updates
Jan 19 12:12:53 DEBUG not found deltainfo for: RPM Fusion for Fedora 25 - Free - Updates
Jan 19 12:12:53 DEBUG not found updateinfo for: RPM Fusion for Fedora 25 - Free - Updates
...
Jan 22 10:51:50 DDEBUG Command: dnf -y update VirtualBox-5.1
Jan 22 10:51:50 DDEBUG Installroot: /
Jan 22 10:51:50 DDEBUG Releasever: 25
Jan 22 10:51:50 DDEBUG Base command: upgrade
Jan 22 10:51:50 DDEBUG Extra commands: ['VirtualBox-5.1']
Jan 22 10:51:50 DEBUG repo: using cache for: virtualbox
Jan 22 10:51:50 DEBUG not found deltainfo for: Fedora 25 - x86_64 - VirtualBox
Jan 22 10:51:50 DEBUG not found updateinfo for: Fedora 25 - x86_64 - VirtualBox

Table 1: Quelques enregistrements produits par DNF, le gestionnaire de paquet de Fedora. Ces enregistrements concernent l'utilisation de DNF au travers de quelques types d'évènements. Cet exemple d'enregistrement est simple : chaque ligne est décomposable en un triplet (date, type d'évènement générique, description de l'évènement).

De nos jours, de nombreuses informations sont enregistrées informatiquement. Un exemple de telles informations est donné par le tableau 1. Les enregistrements de cet exemple sont ceux du gestionnaire de paquet DNF pour la distribution linux Fedora. Chaque enregistrement, représenté par une ligne, est composé de quelques caractères. Chacune de ces lignes est composée d'une date, un type d'évènement et de la description de l'évènement. Un usage évident de tels enregistrements concerne l'activité d'un système informatique. Leur intérêt est le même que pour la videosurveillance : détecter des activités malveillantes ou, du moins, enregistrer chaque activité en cas de malveillance. L'activité de sites web tels qu'Amazon, Google ou Facebook pour ne citer que les plus gros et l'activité de composants informatiques tels que les processeurs et les disques durs sont deux exemples de tels enregistrements. Dans le cadre de l'activité d'un site web, l'enregistrement classique des pages visitées correspond au format $\langle UID, date, page \rangle$. Le tableau 2 illustre de tels enregistrements produits par Apache2. Pour cet exemple, l'UID, c'est-à-dire l'identifiant utilisateur, est l'adresse IP utilisée pour la requête. Pour un utilisateur, chaque log correspond à date a été demandée une certaine page. Ces informations permettent de reconstruire le parcours et les habitudes de chaque utilisateur. De ce fait, les parcours menant à des comportements profitables, comme des achats pour l'exemple d'un site d'e-commerce, peuvent être étudiés à travers ces données et aider les décideurs à réorganiser leurs sites web ou proposer des offres commerciales. Une remarque intéressante concernant cet exemple est que les systèmes d'enregistrements ont été créés dans le but de gérer l'activité de serveurs informatiques et non pour des raisons commerciales. C'est un exemple d'usage secondaire qu'il peut être fait des données. De telles analyses sont de plus en plus communes du fait de l'important volume de données stocké dans les serveurs

37.59.178.104 - - [04/Feb/2018:12:33:47 +0000] "POST /xmlrpc.php HTTP/1.1"
37.59.178.104 - - [04/Feb/2018:12:33:49 +0000] "POST /xmlrpc.php HTTP/1.1"
157.55.39.150 - - [04/Feb/2018:12:58:45 +0000] "GET /robots.txt HTTP/1.1"
207.46.13.145 - - [04/Feb/2018:12:58:49 +0000] "GET / HTTP/1.1"
115.231.219.32 - - [04/Feb/2018:16:57:47 +0000] "GET /manager/html HTTP/1.1"
115.231.219.32 - - [04/Feb/2018:16:58:02 +0000] "GET /manager/html HTTP/1.1"
115.231.219.32 - - [04/Feb/2018:16:58:34 +0000] "GET /manager/html HTTP/1.1"
207.46.13.145 - - [04/Feb/2018:17:01:47 +0000] "GET / HTTP/1.1"
37.59.222.30 - - [04/Feb/2018:17:33:14 +0000] "GET /wp-login.php HTTP/1.1"
37.59.222.30 - - [04/Feb/2018:17:33:14 +0000] "GET /administrator/index.php HTTP/1.1"
37.59.222.30 - - [04/Feb/2018:17:33:14 +0000] "GET /admin.php HTTP/1.1"

Table 2: Quelques enregistrements produits par le serveur HTTP Apache2. Ces enregistrements correspondent à des connexions distantes à un site web. Ces enregistrements ont été simplifiés pour l'exemple et ne contiennent plus que les informations essentielles : l'adresse IP de connexion, la date et la page web accédée.

d'entreprises et de la transformation qu'opère les techniques de fouille de données sur l'industrie et le commerce.

D'autres types de données peuvent être enregistrées sans être liées à l'activité informatique. Un exemple de telles données est celui des bases de données médico-administratives. Une part importante de la population française est assurée par l'assurance maladie et les remboursements effectués par cette assurance sont enregistrés dans de tels bases. Ces remboursements peuvent concerner des délivrances de médicaments, des séjours hospitaliers mais aussi des consultations médicales. De tels enregistrements conduisent à la production de très grands volumes de données qui requièrent une infrastructure importante pour les maintenir et les analyser. De telles données sont initialement collectées pour des raisons administratives mais le SNDS[1] a été créé dans le but de promouvoir leur utilisation dans le cadre de recherches médicales.

Le type des données enregistrées est de plus en plus complexe. Les données du tableau 2 concernant l'activité d'un site web pourraient, par exemple, être enrichies par de nouvelles informations telles que le temps passé sur chaque page, l'aire de la page affichée à l'écran ou le type d'appareil utilisé pour accéder à cette page. Ces données pourraient aussi être enrichies par des enregistrements extérieurs concernant, par exemple, la mise à jour des profils utilisateurs. Une des dimensions de grand intérêt parmi ces données est le temps. Des évènements tels que les délivrances de médicaments, les utilisations d'un processeur ou les accès à une page web sont datés et l'information temporelle est essentielle pour décrire des différences entre, par exemple, une régularité et un pic de consommation. De telles données sont appelées séquentielles si le temps est décrit par l'ordre des occurrences ou temporelles si le temps est décrit numériquement. Dans cette thèse, seules les données représentées par un ensemble de séquences temporelles sont considérées. Chaque élément de cet ensemble, c'est-à-dire chaque séquence, représente les enregistrements décrivant une certaine entité. Par exemple, pour des enregistrements d'un site web, une séquence concernerait un utilisateur. Le tableau 3 illustre la représentation sous forme d'ensemble de séquences des enregistrements du tableau 2. Dans cet exemple, nous faisons l'hypothèse que les requêtes provenant de l'adresse IP 37.59.222.30 sont malveillantes car trois requêtes provenant de cette adresse tentent d'accéder au panneau d'administration et ce, exactement à la même date. Un exemple de motif intéressant pourrait décrire de telles requêtes malveillantes en les différenciant d'autres requêtes comme, par exemple, celles provenant de l'adresse IP 207.46.13.145 qui concernent le crawler web Bingbot (utilisé par Microsoft pour le moteur de recherche Bing).

Un grand nombre d'informations est contenu dans ces bases de données et un défi d'intérêt est

[1] SNDS: Système National des Données de Santé `https://www.snds.gouv.fr/SNDS/Accueil`

IP address	Sequence
37.59.178.104	(/xmlrpc.php, 12:33:47), (/xmlrpc.php, 12:33:49)
157.55.39.150	(/robots.txt, 12:58:45)
207.46.13.145	(/, 12:58:49), (/, 17:01:47)
115.231.219.32	(/manager/html, 16:57:47), (/manager/html, 16:58:02), (/manager/html, 16:58:34)
37.59.222.30	(/wp-login.php, 17:33:14), (/administrator/index.php, 17:33:14), (/admin.php, 17:33:14)

Table 3: Représentation sous forme de séquences temporelles du tableau 1.2. Dans cet exemple, une séquence est composée des évènements concernant les requêtes d'une même adresse IP. Un évènement est une requête associée à une date.

donc d'exploiter ces données pour en extraire de nouvelles connaissances. Ce défi se décompose en deux questions : Comment exploiter de grands ensembles de données et qu'est-ce qui est cherché dans ces données ?

L'analyse de ces données peut être faite à l'aide de modèles statistiques pour lesquels le but est de montrer qu'un comportement, par exemple une stratégie commerciale, a un impact significatif sur les données. Cette approche est utile pour valider une hypothèse, c'est-à-dire, une stratégie donnée. Par exemple, des cliniciens pourraient vouloir tester si un nombre significatif de patients traités a été guéri. Ces mêmes cliniciens pourraient aussi avoir une intuition sur l'effet secondaire d'un médicament, écrire cette intuition sous forme d'hypothèse statistique pour finalement tester sa significativité sur les données. Par exemple, un traitement peut être bon pour la population générale mais les cliniciens voudraient valider qu'il l'est aussi, ou non, pour des populations spécifiques définies par leur genre, leur âge, etc.

Dans certains cas, aucune hypothèse statistique n'a été établie pour expliquer les comportements contenus dans les données. Un traitement des données plus important est donc nécessaire pour identifier de tels comportements. Extraire des comportements intéressants apparaissant dans les données et satisfaisant un ensemble de contraintes utilisateurs semble être une solution efficace pour ce problème. Dans ce cas, l'automatisation de cette tâche est nécessaire. En effet, ce n'est pas raisonnable d'explorer manuellement les données du fait de leur volume. Les algorithmes dédiés à cette tâche seraient donc utilisés en tant qu'assistant pour la génération d'hypothèses.

Le choix de cette thèse est donc d'explorer le champ des approches d'extraction de motifs pour extraire des hypothèses concernant les comportements intéressants apparaissant dans les données. Les motifs sont des objets qui décrivent des comportements satisfaisant certaines contraintes et qui apparaissent dans les données. La contrainte classique utilisée dans ce cas est la contrainte de fréquence minimale. Un motif satisfaisant cette contrainte est extrait si et seulement si ce motif apparait au moins x fois dans les données. Dans ce cas, le paramètre x doit être défini par l'utilisateur. Les exemples classiques d'analyse de paniers de consommateurs, l'analyse de ce qui est acheté par les consommateurs dans les magasins, utilise les *itemsets* comme modèle de motifs. Un itemset est un motif ne considérant que les co-occurrences d'évènements (appelés items dans ce cas). Un exemple d'itemset extrait dans ce cas d'application serait *(abricot, baguette, chocolat)* qui décrit que *abricot*, *baguette* et *chocolat* sont achetés dans le même panier. Pour les enregistrements séquentiels et temporels, de nombreux modèles de motifs ont été proposés tels que les motifs séquentiels [Agrawal and Srikant, 1995], les épisodes [Mannila et al., 1997] ou les chroniques [Dousson and Duong, 1999]. Le choix de modèle de motifs dépend du type de données et de la représentation des connaissances désirée par l'utilisateur.

Afin de proposer un algorithme d'extraction de motifs utilisable par des experts de domaines variés et donc des données variées, le modèle de motifs extraits par un tel algorithme doit être

générique, expressif, extractible et interprétable.

Comme introduit plus tôt, la contrainte utilisateur dédiée à l'extraction de motifs la plus communément utilisée est la contrainte de fréquence minimale. La fréquence d'un motif est la proportion des données dans laquelle ce motif apparaît. Le terme de support est utilisé dans le cas où le nombre absolu d'occurrences est calculé et non la proportion de celles-ci. A partir de ces définitions, l'ensemble des comportements fréquents apparaissant dans les données peut être retourné à l'utilisateur. Pour déterminer si un motif est fréquent ou non, la démarche classique est de demander à l'utilisateur de définir lui-même le seuil de fréquence. Par exemple, si l'utilisateur définit ce seuil comme étant égal à 20% des données, un motif fréquent est un motif apparaissant au moins dans 20% des données. Ce choix du seuil de fréquence permet de régler la généricité du modèle de motifs.

Dans le cas des séquences temporelles, l'approche classique pour définir la fréquence est de considérer le nombre de séquences temporelles dans lesquelles un motif est trouvé. Par exemple, un motif concernant la délivrance de médicaments possédant une fréquence de 20% des séquences décrit un comportement apparaissant chez 20% des assurés. Un autre exemple de motif fréquent serait un motif fréquent caractérisant une requête malveillante. Dans ce cas, connaître ce type de motif aiderait à reconnaître de futures requêtes malveillantes et, par exemple, bannir les adresses IP associées.

L'ensemble des motifs fréquents peut pourtant être si volumineux qu'il serait difficile de l'extraire ou d'en tirer une information utile. Par exemple, pour des séquences de délivrances de médicaments, une proportion importante des motifs fréquents impliquerait l'aspirine ou le paracétamol. Détecter que de tels médicaments sont fréquemment délivrés n'est pas réellement utile pour les cliniciens. Un grand nombre de motifs inintéressants est ainsi généré par les algorithmes d'extraction de motifs fréquents. En outre, l'ensemble des motifs fréquents peut contenir des connaissances déjà connues du fait de l'existence de comportements réguliers inintéressants. De ce fait, certains travaux ont été réalisés sur d'autres types de contraintes utilisateurs qui leur permettraient de contraindre l'ensemble des motifs extraits afin d'être plus intéressants.

La contrainte étudiée dans cette thèse est la contrainte de croissance minimale. Cette contrainte requière un jeu de données dont les séquences sont labellisées. Ces labels sont utilisés pour définir des classes de séquences et donc, un motif discriminant, c'est-à-dire un motif satisfaisant la contrainte de croissance minimale, est un motif qui apparait plus souvent dans les séquences d'une classe que dans les autres. Par exemple, pour des séquences associées à des personnes, les classes pourraient être relatives à leur âge ou leur genre. Utiliser de tels labels permet d'extraire des comportements spécifiques associés uniquement, par exemple, aux femmes ou aux jeunes.

Les motifs discriminants sont plus intéressants pour l'utilisateur car ils représentent de potentielles réponses aux questions de celui-ci. Par exemple, les séquences du tableau 3 pourraient être labellisées par l'utilisateur comme requête malveillante ou non. A partir de ces labels, les motifs discriminants extraits concerneraient ce qui est récurrent parmi les requêtes malveillantes mais ne l'est pas, ou moins, parmi les autres.

Pour des ensembles de séquences labellisées, les motifs discriminants sont plus intéressants que les motifs fréquents car ils :

- sont utilisables dans un outil de détection automatique (par exemple de comportements malveillants)
- permettent de donner un aperçu de ce que pourrait être un comportement spécifique (par exemple un comportement malveillant)

Ce second point est l'atout majeur des motifs discriminants. Contrairement aux meilleurs algorithmes de classification, tels que les réseaux de neurones profonds, la classification faite par des

motifs est explicable. De ce fait, les motifs discriminants peuvent être utilisés pour retourner des hypothèses de travail aux décideurs ou utilisés dans un processus de classification supervisée par l'utilisateur. Dans le contexte de la pharmaco-épidémiologie, les labels peuvent être associés aux résultats des traitements. Dans ce cas, le but est d'extraire les motifs qui discriminent l'un de ces résultats, par exemple la guérison de patients, par rapport aux autres. Les cliniciens peuvent ainsi découvrir et comprendre pourquoi un traitement est efficace dans certains cas et non dans d'autres.

Les motifs discriminants sont toutefois moins extractibles que les motifs fréquents. La contraint de croissance minimale n'est pas anti-monotone comme l'est celle de fréquence minimale et les astuces des algorithmes classiques ne peuvent donc pas être utilisées pour extraire ce type de motifs. De plus, peu de travaux ont tentés d'extraire des motifs discriminants incluant une information temporelle. Choisir où séparer la dimension temporelle afin d'obtenir des motifs discriminants n'est pas simple. Par exemple, la séquence associée à l'adresse IP 37.59.222.30, est considérée malveillante entre autres choses car elle contient trois évènements apparaissant exactement à la même date. Mais il est envisageable de considérer la séquence associée à l'adresse IP 115.231.219.32 comme malveillante car elle contient trois évènements apparaissant dans un intervalle de temps très court. Dans cet exemple concis, ces deux séquences peuvent être distinguées manuellement de la séquence associée à l'adresse IP 207.46.13.145 mais une telle distinction deviendrait plus compliquer pour des milliers de séquences. Extraire effectivement des motifs temporels discriminants est donc un défi important du fait de sa difficulté et de l'intérêt de la dimension temporelle à différencier des motifs. C'est donc à ce défi que cette thèse s'est attaqué.

Contributions

La contribution initiale de cette thèse a été le choix du modèle de chroniques pour représenter des comportements temporels. Une chronique est un couple $(\mathscr{E}, \mathscr{T})$ tel que $\mathscr{E}$ est un multi-ensemble d'évènements et $\mathscr{T}$ un ensemble de contraintes temporelles s'appliquant à $\mathscr{E}$. La figure 1 illustre plusieurs exemples de chroniques. Pour toutes ces chroniques le multi-ensemble d'évènements $\mathscr{E}$ correspond à *Abricot*, *Baguette* et *Chocolat*. Pour la chronique $\mathscr{C}_1$, l'arrête allant de *Chocolat* à *Baguette* signifie que *Baguette* apparaît au moins 1 unité de temps après *Chocolat*. L'unité de temps n'est pas définie ici mais elle pourrait typiquement être fixée au jour.

L'intérêt de ce choix de modèle est de représenter des comportements intéressants dans le contexte de la pharmaco-épidémiologie. Ce choix cible particulièrement l'expressivité temporel du modèle de chronique et les précédents usages des chroniques de le domaine médical. L'expressivité temporelle des chroniques est importante pour le contexte de la pharmaco-épidémiologie du fait de l'importance du temps dans le domaine médicale. Par exemple, qu'une délivrance de médicaments soit hebdomadaire ou mensuelle ne conduira pas aux mêmes résultats et ces deux régularités de délivrance doivent donc être considérées différentes. Le modèle de chronique a donc été adopté par les cliniciens et autres membres du projet pour son expressivité.

Les contributions de cette thèse, centrées sur le modèle de chroniques, se regroupent en trois parties :

- L'extraction de chroniques discriminantes
- La généralisation des chroniques discriminantes et l'étude de leur interprétabilité
- L'application de l'extraction de chroniques discriminantes sur un cas d'étude de pharmaco-épidémiologie

La première contribution de cette thèse a donc été de proposer la tâche d'**extraction de chroniques discriminantes**. L'extraction de chroniques fréquentes ou d'autres motifs discriminantes ont été étudiés avant ce travail mais pas la combinaison des deux.

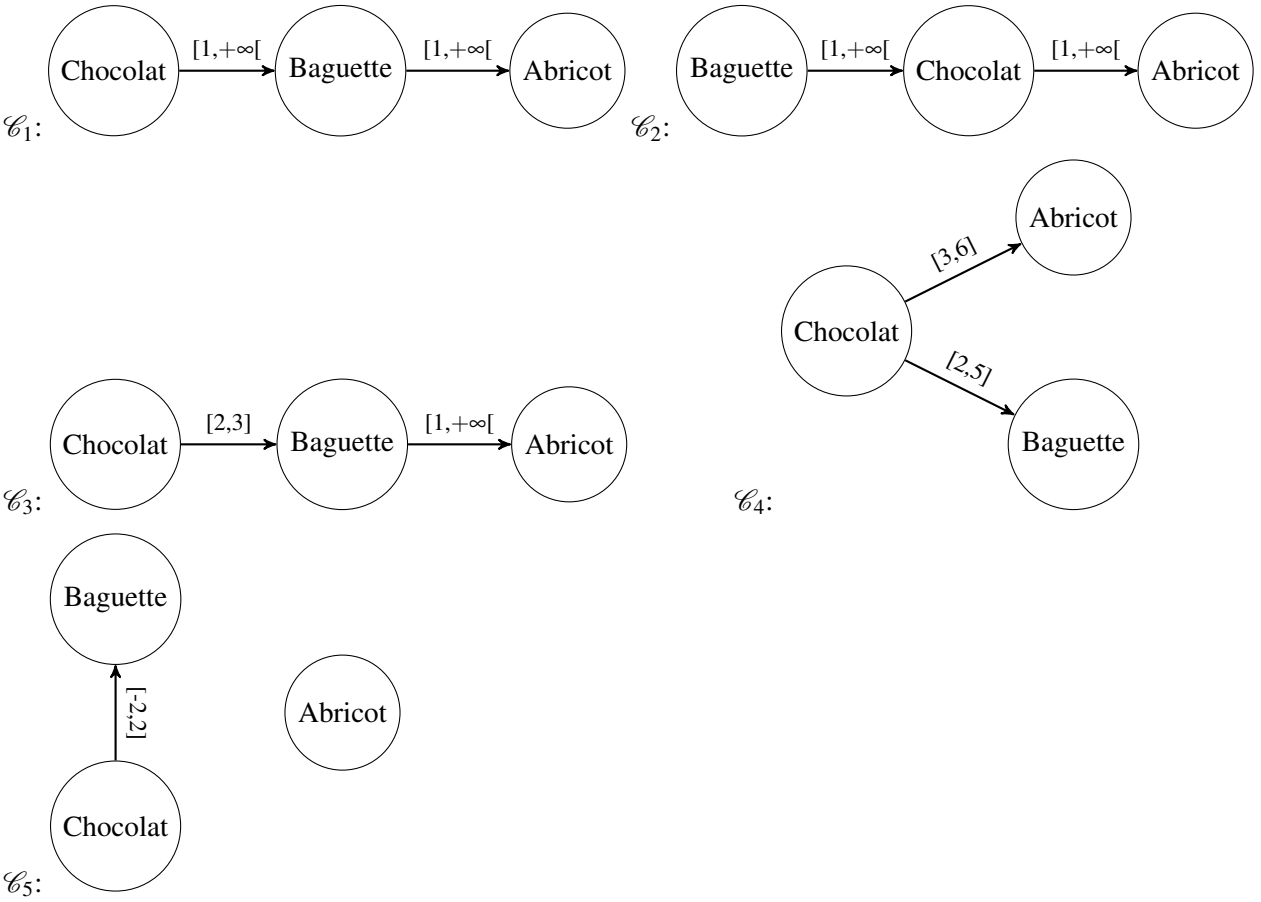


Figure 1: Illustrations de chroniques représentant plusieurs comportements temporels basés sur les évènements concernant les achats : Abricot, Baguette et Chocolat.

L'avantage du modèle de chronique est d'être plus expressif que des motifs plus standard tels que les motifs séquentiels. La différence principale entre ces deux modèles de motifs est la façon de représenter la dimension temporelle. Là où les motifs séquentiels représentent la dimension temporelle uniquement par de la séquentialité, c'est-à-dire par l'ordre temporel des évènements, les chroniques représentent le temps numériquement et ne contraignent pas forcément les évènements à apparaître dans un ordre séquentiel donné. Par exemple, le comportement d'une personne achetant du chocolat puis une baguette puis des abricots peut être représenté par un motif séquentiel. L'ordre séquentiel décrit la différence avec d'autres motifs tels que, par exemple, une baguette puis du chocolat et enfin des abricots. Ces deux comportements sont aussi représentables par des chroniques. Les chroniques $\mathscr{C}_1$ et $\mathscr{C}_2$ de la figure 1 représentent respectivement ces deux comportements. Mais il est aussi possible de représenter des comportements tels qu'acheter une baguette entre 2 et 3 jours après avoir acheté du chocolat puis ensuite acheter des abricots. Ce comportement est illustré par la chronique $\mathscr{C}_3$. Pour ce comportement, une information temporelle numérique a été ajouté à la séquentialité des évènements. Un autre exemple, serait l'achat d'une baguette entre 2 et 5 jours après l'achat de chocolat et l'achat d'abricots entre 3 et 6 jours après l'achat du chocolat. Ce comportement est illustré par la chronique $\mathscr{C}_4$. Dans ce cas, il n'y a aucune séquentialité entre les évènements concernant l'achat d'une baguette et l'achat d'abricots. Un dernier exemple serait l'achat d'une baguette entre 2 jours avant l'achat de chocolat et 2 jours après. Dans ce cas une information temporelle est donnée et l'occurrence de ces deux évènements est contrainte, mais aucune séquentialité n'a été imposée. L'achat d'une baguette peut tout à fait arriver avant ou après l'achat de chocolat. Ce comportement est illustré par la chronique $\mathscr{C}_5$.

L'expressivité du modèle de chronique peut donc permettre de décrire des comportements discriminants que les motifs séquentiels ne peuvent pas décrire. Cela rend d'autant plus intéressant l'extraction de chronique discriminante mais c'était un défi important d'extraire effectivement de tels motifs. En effet, la représentation numérique du temps rend plus difficile le choix des bornes des contraintes temporelles. Les combinaisons de toutes possibilités bornes ne peuvent pas être énumérées en temps raisonnable pour d'autres données que des données d'exemple et une heuristique efficace doit donc être utilisée pour n'extraire qu'un-sous ensemble intéressant de ces combinaisons.

L'algorithme d'extraction de chroniques discriminantes DCM proposé dans cette thèse s'appuie sur l'heuristique incomplète de l'algorithme d'apprentissage de règles numériques $Ripper_k$ pour extraire efficacement des contraintes temporelles discriminantes. Pour un multi-ensemble d'évènements donné, il est possible de représenter l'ensemble de ses occurrences temporelles sous la forme de données tabulaires qu'un algorithme d'apprentissage de règles numériques pourra ensuite utiliser. L'algorithme DCM énumère donc l'ensemble des multi-ensembles d'évènements fréquents puis retourne l'ensemble de leurs occurrences temporelles à $Ripper_k$ afin d'obtenir des règles numériques que seront ensuite traduites en contraintes temporelles discriminantes. L'association des multi-ensembles fréquents et de ces contraintes temporelles discriminantes permet finalement de générer un ensemble de chroniques discriminantes.

L'efficacité de DCM à effectivement extraire les chroniques discriminants un ensemble de séquences temporelles a été évaluée sur données synthétiques. Des séquences labellisées positives ou négatives ont été générées à partir de chroniques discriminants ces deux labels. L'algorithme DCM a effectivement extrait les chroniques discriminants les séquences positives et négatives. Cette évaluation nous a ainsi permis de montrer que l'heuristique de $Ripper_k$, quoique incomplète, extrait effectivement les contraintes temporelles permettant de discriminer les ocurrences d'un multi-ensemble.

L'intérêt de l'expressivité des chroniques discriminantes a ensuite été évaluée en comparant les résultats de chroniques discriminantes extraites en matière de précision, utilisées pour classifier de nouvelles séquences, avec les résultats obtenus à partir de motifs séquentiels discriminants.

Cette évaluation a été effectuée sur données réelles. Les résultats ont montré que l'utilisation des chroniques discriminantes pour classifier des séquences temporelles était plus intéressante pour certains jeux de données que l'utilisation de motifs séquentiels discriminants. Cette évaluation permet donc de justifier l'intérêt de la dimension temporelle pour la classification de séquences et l'intérêt des chroniques à représenter cette dimension.

La seconde contribution de cette thèse concerne la **généralisation des chroniques** et l'étude de leur **interprétabilité**. Le choix du modèle de chroniques a été fait pour son expressivité et son interprétabilité mais l'interprétabilité des chroniques n'a jamais été clairement étudié précédemment. Une généralisation du modèle de chronique discriminante a donc été proposé et la définition de différents types de chroniques généralisées a permis de comparer leur interprétabilité. L'idée de la généralisation des chroniques est de remplacer $Ripper_k$ par n'importe quel autre algorithme de machine learning. Dans le cas d'algorithmes de machine learning construisant des modèles plus complexes que des règles numériques, les motifs obtenus ne seraient donc plus des chroniques mais une forme généralisée de celles-ci. Un algorithme de machine learning plus expressif extrait des chroniques généralisées plus expressives mais impacte négativement leur interprétabilité. Le compromis entre, ce gain en expressivité et cette perte d'interprétabilité, est comparé pour plusieurs types de chroniques généralisées. Ce compromis est d'autant plus important pour des chroniques discriminantes puisqu'elles peuvent être utilisées dans un contexte de classification. Grâce à ce contexte de classification, il est donc possible d'évaluer le gain en expressivité d'un type de chronique généralisée en se basant sur la précision de celui-ci à classifier de nouvelles séquences. La comparaison des différents résultats en matière de précision obtenus par plusieurs types de chroniques généralisées a donc pu être effectuée sur un ensemble de jeux de données réels et synthétiques.

Pour terminer, la troisième contribution de cette thèse a été d'extraire des comportements liés à l'épilepsie grâce aux chroniques discriminantes sur un **cas d'étude de pharmaco-épidémiologie** : GENEPI. Le jeu de données associé à cette étude concerne des personnes épileptiques pour lesquelles un traitement stable a duré au moins un an, mais qui ont finalement été hospitalisées pour au moins une crise d'épilepsie.

Ce travail fait parti du projet PEPS (Pharmaco-Epidémiologie des Produits de Santé) porté par le CHU de Rennes et financé par l'ANSM (Agence Nationale de Sécurité du Médicament). Le but principal de ce projet est de mener des études de pharmaco-épidémiologie. L'étude GENEPI en est l'une d'elles. Le but initial de cette étude était de valider l'association entre une hospitalisation pour crise d'épilepsie et une substitution d'un médicament anti-épileptique princeps à un médicament anti-épileptique générique. Un nombre important de pré-traitement avait à l'origine dû être fait pour tester statistiquement cette seule hypothèse. Parmi ces pré-traitements, les plus notables sont la qualification de période de stabilité et de substitution de médicaments princeps à générique. Nous avons donc décider d'extraire des chroniques discriminantes à partir de ce jeu de données :

- Pour ne confirmer qu'aucune chronique discriminante relative à l'hypothèse précédente n'a été extraite (les résultats de l'étude l'ont rejetée).
- Pour explorer plus profondément le jeu de données en extrayant des comportements temporels discriminants qui pourraient intéresser les cliniciens.

Les chroniques extraites ont finalement identifié des substitutions entre médicaments anti-épileptiques mais pas entre princeps et génériques. Ces chroniques ont de plus discriminé ces substitutions au travers de leur dimension temporelle. Une telle précision dans la description d'un comportement est plus difficile à obtenir au travers du processus classique d'étude de pharmaco-épidémiologie mais est pourtant obligatoire pour répondre aux questions les plus complexes des

cliniciens. De plus, l'automatisation de l'exploration de l'espace de recherche nous a permis de généraliser le contexte de l'étude originale. Les chroniques extraites ont finalement été validées par les cliniciens et des pistes d'améliorations de l'extraction et d'extraction d'autres types de comportements ont été proposées. Les résultats obtenus confirment l'utilité d'une telle approche dans la réalisation d'étude de pharmaco-épidémiologie.

Cette thèse contribue finalement à l'extraction de motifs temporels et en souligne l'importance. En effet, les hypothèses de travail données par les chroniques discriminantes extraites aux experts sont très importantes. De telles hypothèses de travail sont difficilement obtenable en utilisant des méthodes classiques de classification de séquences telles que les réseaux de neurones profonds. Par exemple, les motifs extraient à partir du jeu de données GENEPI sont compréhensibles par les cliniciens. Un algorithme de classification dit "boîte noire" pourrait bien sûr obtenir une meilleure précision. Mais, dans cette thèse, l'idée défendue est qu'une faible amélioration en matière de précision ne peut justifier une perte de l'interprétabilité du modèle.

Remerciements

Je m'étais dit que je travaillerais mes remerciements, mais je m'y suis pris à la dernière minute donc je vais être concis. Pour changer.

Merci à tous les employés de l'IRISA d'en faire un lieu de travail convivial, et plus particulièrement, merci à tous les membres de l'équipe Lacodam, tous très sympathiques et prêts à se sacrifier pour faire une pause et aller boire un café ensemble. D'ailleurs, si l'un d'entre vous tombe sur cette page, je veux bien amener un gâteau pour le partager avec vous. Merci à Marie-Noëlle d'avoir fait en sorte que les tâches administratives liées à ces trois années de thèse soient totalement transparentes. Merci à mes directeurs de thèse de m'avoir soutenu et encadré pendant trois ans, et notamment à Thomas qui était en première ligne et continue à m'encourager et m'accompagner dans mon travail de recherche, même après ma soutenance. Merci aussi aux membres de mon jury d'avoir accepté de relire cette thèse et d'avoir bravé les grèves pour assister à ma soutenance. Je ne les ai sûrement pas assez remerciés sur le moment mais, pour ma défense, je n'avais aucune idée de ce qui se passait après la délibération. Et enfin, merci à Mélusine qui m'a supporté pendant ces trois années et compte, parmi ses hauts faits, le sauvetage de mon pot de thèse.

J'ai particulièrement apprécié ces trois années de thèse dont je ne garderai que de bons souvenirs. Merci à tous !

Discriminant chronicle mining

Contents

1 Introduction **19**

I State of the art **27**

2 Temporal pattern mining **29**
2.1 Sequential pattern mining 30
2.1.1 Standard sequential patterns 30
2.1.2 Partially ordered patterns 39
2.2 Mining of patterns including numerical temporal information 40
2.2.1 Chronicle definition 41
2.2.2 Chronicle mining approaches 43

3 Supervised pattern mining **45**
3.1 Discriminant rule mining 45
3.2 Discriminant temporal pattern mining 48
3.3 Pattern-based classification 48
3.4 Positioning 50

II Contributions **51**

4 Discriminant chronicle mining **53**
4.1 Discriminant chronicles 54
4.2 DCM algorithm 55
4.2.1 Multiset mining 56
4.2.2 Temporal constraint mining 56
4.3 Dealing with the multiple instance issues 59
4.3.1 Multiple instance problem 59
4.3.2 Witness selection for $Ripper_k$ 60
4.3.3 A multiple instance rule learning algorithm: MITI 61
4.4 Decision based on discriminant pattern sets 61
4.5 Benchmark 62
4.5.1 Synthetic data 62
4.5.2 *BIDE-D* comparison 65
4.6 Conclusion 67

5 Interpretable temporal pattern-based classification **69**
5.1 Generalized chronicles . . . 71
5.1.1 Discriminant generalized chronicles . . . 72
5.1.2 Classification based on generalized chronicles . . . 74
5.1.3 Temporal pattern interpretability . . . 77
5.2 Experiments . . . 80
5.2.1 Experimental setup . . . 80
5.2.2 Results . . . 84
5.3 Conclusion and perspectives . . . 91

6 Case study **95**
6.1 Pharmaco-epidemiology based on the SNDS . . . 95
6.2 GENEPI . . . 98
6.3 Discriminant chronicle mining for pharmaco-epidemiology . . . 100
6.4 Positive and negative datasets construction . . . 101
6.5 Experiments and results . . . 102
6.5.1 Taking into account time in brand-to-generic substitution . . . 103
6.5.2 Example of a complex chronicle leading to new hypotheses . . . 104
6.6 Conclusion . . . 105

7 Conclusion **107**
7.1 Discussion and perspectives . . . 108
7.1.1 Application . . . 108
7.1.2 Discriminant chronicle mining algorithm . . . 109
7.1.3 Generalized discriminant chronicles . . . 111
7.1.4 Measures of interpretability . . . 112

Glossary **115**

A DCM implementation **117**

B ATC descriptions **123**

C Publications **127**

Chapter 1

Introduction

Jan 19 12:12:52 DEBUG repo: using cache for: fedora
Jan 19 12:12:52 DEBUG not found updateinfo for: Fedora 25 - x86_64
Jan 19 12:12:53 DEBUG repo: using cache for: rpmfusion-free-updates
Jan 19 12:12:53 DEBUG not found deltainfo for: RPM Fusion for Fedora 25 - Free - Updates
Jan 19 12:12:53 DEBUG not found updateinfo for: RPM Fusion for Fedora 25 - Free - Updates
...
Jan 22 10:51:50 DDEBUG Command: dnf -y update VirtualBox-5.1
Jan 22 10:51:50 DDEBUG Installroot: /
Jan 22 10:51:50 DDEBUG Releasever: 25
Jan 22 10:51:50 DDEBUG Base command: upgrade
Jan 22 10:51:50 DDEBUG Extra commands: ['VirtualBox-5.1']
Jan 22 10:51:50 DEBUG repo: using cache for: virtualbox
Jan 22 10:51:50 DEBUG not found deltainfo for: Fedora 25 - x86_64 - VirtualBox
Jan 22 10:51:50 DEBUG not found updateinfo for: Fedora 25 - x86_64 - VirtualBox

Table 1.1: Some lines in the log file of DNF, the package manager of Fedora. This log file is dedicated to record the usage of DNF by recording some types of event. This is an example of logs were each record is simple. Each line can be decomposed as a timestamp, a generic event type and a string describing in more detail what happened.

Nowadays, a lot of information is recorded into logs. An example of such logs recorded by the DNF package manager of Fedora is given by Table 1.1. Each record, represented by a line, is composed by few characters. On each line, we can find a timestamp, an event type and a description of the event. One obvious usage of such log is done to record computer activity. The purpose of such record is the same as video surveillance: detect malicious activity or, at least, save information in case of malicious activity. The activity of websites like Amazon, Google or Facebook for the biggest ones and the usage of the hardware components like CPU or hard drive disks are two examples of such recorded activities. For a website activity, the standard logs tracking the pages visited by users on the website could have the format $\langle UID, timestamp, where \rangle$. The Table 1.2 illustrates such type of logs produced by Apache2. In this case the UID is a user ID if the user is logged in but it can also be IP addressed as in Table 1.2 or information that could allow to track the user. For this user, each log corresponds to where (s)he was on the site for a specific timestamp *i.e.* when (s)he was on the website and from where (s)he came. This information can allow to rebuild the browsing and habits of this user. Thereby, the paths leading to profitable behaviors like, for example, purchases, can be studied through those data and support stakeholders to reorganize the websites or make commercial offers. One interesting remarks about this example is that web log collection system has been created for server management purposes and not for commercial reasons. This is an example of secondary usage of data. Such kind of analysis is more

37.59.178.104 - - [04/Feb/2018:12:33:47 +0000] "POST /xmlrpc.php HTTP/1.1"
37.59.178.104 - - [04/Feb/2018:12:33:49 +0000] "POST /xmlrpc.php HTTP/1.1"
157.55.39.150 - - [04/Feb/2018:12:58:45 +0000] "GET /robots.txt HTTP/1.1"
207.46.13.145 - - [04/Feb/2018:12:58:49 +0000] "GET / HTTP/1.1"
115.231.219.32 - - [04/Feb/2018:16:57:47 +0000] "GET /manager/html HTTP/1.1"
115.231.219.32 - - [04/Feb/2018:16:58:02 +0000] "GET /manager/html HTTP/1.1"
115.231.219.32 - - [04/Feb/2018:16:58:34 +0000] "GET /manager/html HTTP/1.1"
207.46.13.145 - - [04/Feb/2018:17:01:47 +0000] "GET / HTTP/1.1"
37.59.222.30 - - [04/Feb/2018:17:33:14 +0000] "GET /wp-login.php HTTP/1.1"
37.59.222.30 - - [04/Feb/2018:17:33:14 +0000] "GET /administrator/index.php HTTP/1.1"
37.59.222.30 - - [04/Feb/2018:17:33:14 +0000] "GET /admin.php HTTP/1.1"

Table 1.2: Some logs of the HTTP server Apache2. The logs are simplified here to be only IP addresses, timestamps and accessed web pages.

and more common due to the tremendous amount of data that is hidden in company servers and that the data mining revolution may transform as business nuggets.

Other data can be recorded without being firstly linked to computer activity. Another example is the medico-administrative databases. A great part of the French population is insured by the national healthcare program, and the reimbursements are recorded in such database. Those reimbursements are done for drug deliveries, hospital stays but also medical consultations. Some care pathway described by this information can be seen as a patient log. This leads to big sets of data requiring a big infrastructure to manage and analyze them. These data are collected for administrative reasons but the French SNDS[1] has been created in order to promote their use for medical research purposes.

The type of the recorded data is more and more complex. From the previous example on website activities, we can enrich the logs with new information like the time spend on the page, the page subarea displayed on the screen or the device used to access the website. The data can also be enriched with exogenous logs like logs concerning the update of the user profiles. A highly valuable dimension of these data is the time. Events like drug deliveries, CPU usages or web connections are timestamped and the temporal information is essential to describe some differences between, for example, regularity and consumption peak. We call such type of data *sequential* traces if the time is described by the order of occurrence of the events or *temporal* if the time is described numerically. In this thesis we consider only data represented by a set of temporal sequences. Each element of this set represents records that describe an independent entity. For example, in website logs, a way to represent the log set for which each element concerns a user. A temporal sequence is a temporal trace concerning a website user or an insured person. Table 1.3 illustrates a sequential representation of the logs in Table 1.2. In this example, we can assume that request from 37.59.222.30 are malicious because three requests from this address attempt to access the administrator panel at the same timestamps. An example of interesting pattern could describe such malicious requests to discriminate them from, for example, requests from 207.46.13.145 which are requests from the web crawler Bingbot (used for the Microsoft search engine Bing).

A lot of information is then contained in these record bases, and a great challenge is to exploit these data to discover new knowledge. The challenge is twofold: How can we process those big sets of data and what is searched in those data?

Data analysis can be done with statistical models for which the goal is to show that a given strategy or behavior is good or bad. This approach is useful to validate a *hypothesis i.e.* the given strategy. For example, clinicians would like to test whether the significant number of patients

[1] SNDS: Système National des Données de Santé (National System for Healthcare Data) `https://www.snds.gouv.fr/SNDS/Accueil`

IP address	Sequence
37.59.178.104	(/xmlrpc.php, 12:33:47), (/xmlrpc.php, 12:33:49)
157.55.39.150	(/robots.txt, 12:58:45)
207.46.13.145	(/, 12:58:49), (/, 17:01:47)
115.231.219.32	(/manager/html, 16:57:47), (/manager/html, 16:58:02), (/manager/html, 16:58:34)
37.59.222.30	(/wp-login.php, 17:33:14), (/administrator/index.php, 17:33:14), (/admin.php, 17:33:14)

Table 1.3: Sequential representation of the logs in Table 1.2. In this example, a sequence is composed by the events of an IP address. An event is a web page request at a specific timestamp.

treated by a drug was healed. They can also have an intuition about adverse drug reaction, transform it as a hypothesis to evaluate it statistically and check such intuition on the data. A treatment can be good for the global population but we want to validate that this is the same for more specific population defined by their gender, their age, etc.

In some cases, no statistical hypothesis is known to explain some behaviors in datasets. A heavy process is so needed to identify those behaviors. Extracting interesting behaviors existing in a dataset satisfying a set of user constraints seems to be an efficient solution for this problem. Automation is needed in this case. It is not reasonable to explore the data manually due to their large size. Such algorithm will be used as an assistant for hypothesis generation.

The choice of this thesis is to explore the field of pattern mining to extract hypotheses of interesting behaviors from data. Patterns are objects that describe behaviors satisfying some constraints and that are found in datasets. The standard constraint used for this is the minimal frequency constraint where a pattern is extracted if and only if this pattern occurred at least x times in the data. In this case, the parameter x has to be defined by the user. The basic example of market-basket analysis, the analysis of what is bought in specific shops, uses *itemset* as pattern model. An itemset is a pattern only considering co-occurrence of events (or items in this case). For the example of market-basket analysis *(bread, butter, chocolate)* is an itemset that describes that *bread*, *butter* and *chocolate* are bought together. For sequential and temporal logs, several behavior models have been proposed such that sequential patterns [Agrawal and Srikant, 1995], episodes [Mannila et al., 1997] or chronicles [Dousson and Duong, 1999]. The choice of the pattern model depends on the data type and the knowledge representation expected by the user.

To propose an algorithm for various experts analyzing various types of temporal trace, the behavior model extracted by such algorithm is needed to be (i) generic, (ii) expressive, (iii) extractable and (iv) interpretable.

Generic The model genericity is the ability of a model to represent knowledge for various data domain. It concerns not directly the use of an algorithm by the experts but its robustness. Indeed, an expert extracting behaviors for his/her type of traces does not need a generic model but only a model fitting his/her data. An ad hoc model fitting the data can then be better than a generic model for the expert criteria. However, a generic model fits more likely new data and then the research done for it can be reused for new data. Furthermore, a generic model proven to be good for previous data type can be seen more robust than an ad hoc model.

Extractable The extractability of a model concerns the property of a model to be extracted by an efficient algorithm. This is mainly related to the number of possible behaviors that the model can represent and so, the expressiveness of the model. Let's compare two models m_1 and m_2, if m_1

generates theoretically less combinatorial possibility than m_2, it used to practically generates less behaviors than m_2 for a specific dataset and so the extraction is simpler.

As said for the expressiveness, too complex a model would be hardly extractable, in terms of computational time or memory space. Indeed, the number of possible behaviors generated through the model would be so big that the extracted behaviors require more memory than the initial data.

Interpretable The interpretability of a model is a complex property that is discussed with more details in Section 5.1. This property is related to the understandability of the extracted behaviors done by the expert and the intelligibility of the knowledge represented by those behaviors. Similar to the extractability, if a model is too complex, the description of the extracted behaviors could be difficultly interpretable.

Expressive The expressiveness of a model is its capacity to discriminate behaviors. It is an important trait as it often requires a trade-off with the extractability and interpretability of the model. If an expert needs to extract new knowledge from a dataset, the model dedicated to extract this knowledge needs to be complex enough to describe it. The use of too simple a model would lead to the extraction of too simple knowledge, potentially already known by the expert. On the other hand, the choice of a too complex model could limit the extractability and interpretability of the model. A model has to be chosen to be expressive enough to represent the knowledge in the data.

The most common user constraint dedicated to extract such patterns is the minimal frequency. The frequency of a pattern is the proportion of data in which this pattern occurs. The term of support is used for the absolute number of occurrences and not the proportion. Based on these definitions, all the frequent behaviors found in the data will be returned to the expert. To determine if a pattern is frequent or not, the standard way is to ask to the expert to provide a threshold. For example, a frequent pattern is a pattern occurring at least in 20% of the data. This is a way to tune the genericity of the model.

In the case of temporal sequences, the standard way to consider the frequency is to consider the number of temporal sequences in which a pattern is found. For example, a pattern concerning drugs delivered with a frequency of 20% described a behavior occurring for 20% of the insured population. Another example of frequent pattern could be a frequent malicious request pattern. In this case, knowing such patterns could help to recognize future similar malicious requests and, for example, blacklist the IP address.

But the frequent pattern set can be so big that the pattern set will be difficultly extractable and useless. For example, in drug delivery sequences, a big part of the frequent sequential patterns involves aspirin or paracetamol. Knowing that such drugs are frequently delivered is not really useful for clinicians. A lot of uninteresting patterns will be generated by frequent pattern mining algorithms. Furthermore, the frequent pattern set can contain already known knowledge due to very common or not interesting regular behaviors. Thereby, some works have been done on generic user constraints that could allow him/her to constrain extracted patterns to be more interesting for him. It can be simple constraints like maximal frequency where the too frequent patterns are not presented to the user. Some other constraints concern the complexity of the extracted behaviors like the length of the extracted patterns where the number of events concerned by the pattern is restricted by a lower and upper bounds. Some constraints are specified on the pattern set, for example, the maximal or closed patterns. With the latter constraint, a pattern is extracted if there is no larger pattern containing it with the same frequency. Such pattern set constraint reduces the information redundancy.

The constraint that is studied in this thesis is the discriminancy. This constraint requires dataset of labeled sequences. Labels are used to define class of sequences and a discriminant pattern is

a pattern that occurs more likely for a class than another. For example, for sequences concerning people, the class label can be the gender or age categories. Using those labelsn we can then extract the behaviors concerning, for example, women or young people.

Discriminant patterns are more interesting for users because they represent potential answers to the user question. For example, the sequences in Table 1.3 could be labeled by the user to be malicious request or not. From those labels, the extracted discriminant patterns will concern what is recurrent in malicious requests and not in the other.

For datasets of labeled sequences, discriminant patterns are more interesting than frequent patterns:

- to be used in an automatic detection tool of malicious behaviors
- to give insights about what a malicious behavior is

This second point is the main upside of discriminant patterns. In contrary to the best approaches of classification like deep neural networks, the classification made by a pattern is explicable. Thereby, discriminant patterns can be used to give insights to stakeholders or in classification process supervised by humans. In the pharmaco-epidemiology context, the class labels can also be the treatment results. In this case, we want to extract the patterns that discriminate an outcome, for example the patient healing, from other possible outcomes. In such a way, the clinician can discover and understand why the treatment was successful in some cases and not in some others.

Discriminant patterns are, however, less extractable than frequent patterns. Discriminant constraint is not anti-monotonic and standard algorithmic tricks of frequency constraints does not hold. In contrary to the frequent patterns, it is not possible to extend a discriminant pattern until it is no more discriminant to extract them all. Such property is discussed later in chapter 4. Furthermore, few works have tried to extract discriminant patterns including temporal information. Choosing where to split the temporal dimension to obtain discriminant patterns is not simple. For example, the sequence with IP address 37.59.222.30, is considered malicious among other things because the three events occurred at the same timestamp. But we could also think that the sequence with IP address 115.231.219.32 is malicious because the three events occurred within a short period. In this small example, we can split those two previous sequences manually with the sequence with IP address 207.46.13.145 but it becomes more difficult with thousands of sequences and as many different cases. There is so a challenge to effectively extract discriminant temporal patterns because the time is in many cases very important to represent patterns.

Contributions

The initial contribution of this thesis was to choose to work on the chronicle model to represent temporal behaviors. A chronicle is a couple $(\mathscr{E}, \mathscr{T})$ such that $\mathscr{E}$ is a multiset of events and $\mathscr{T}$ a set of temporal constraints applied to the multiset $\mathscr{E}$. Figure 1.1 illustrates several chronicles. Chronicles are more formally defined in section 2.2.1. The aim of this choice was to represent interesting behavior for the pharmaco-epidemiological context. This choice was focused on the temporal expressiveness of the chronicle model and the previous usage of chronicles in medical domain. The temporal expressiveness of the chronicle is important for the pharmaco-epidemiological context because the temporal dimension of the medical data is very important. For example, a weekly drug delivery or a monthly one will not have the same outcome on the health and have to be considered differently. The chronicle model had been adopted by the clinicians and other members of the project for this expressiveness.

The following contributions of this thesis are threefold:

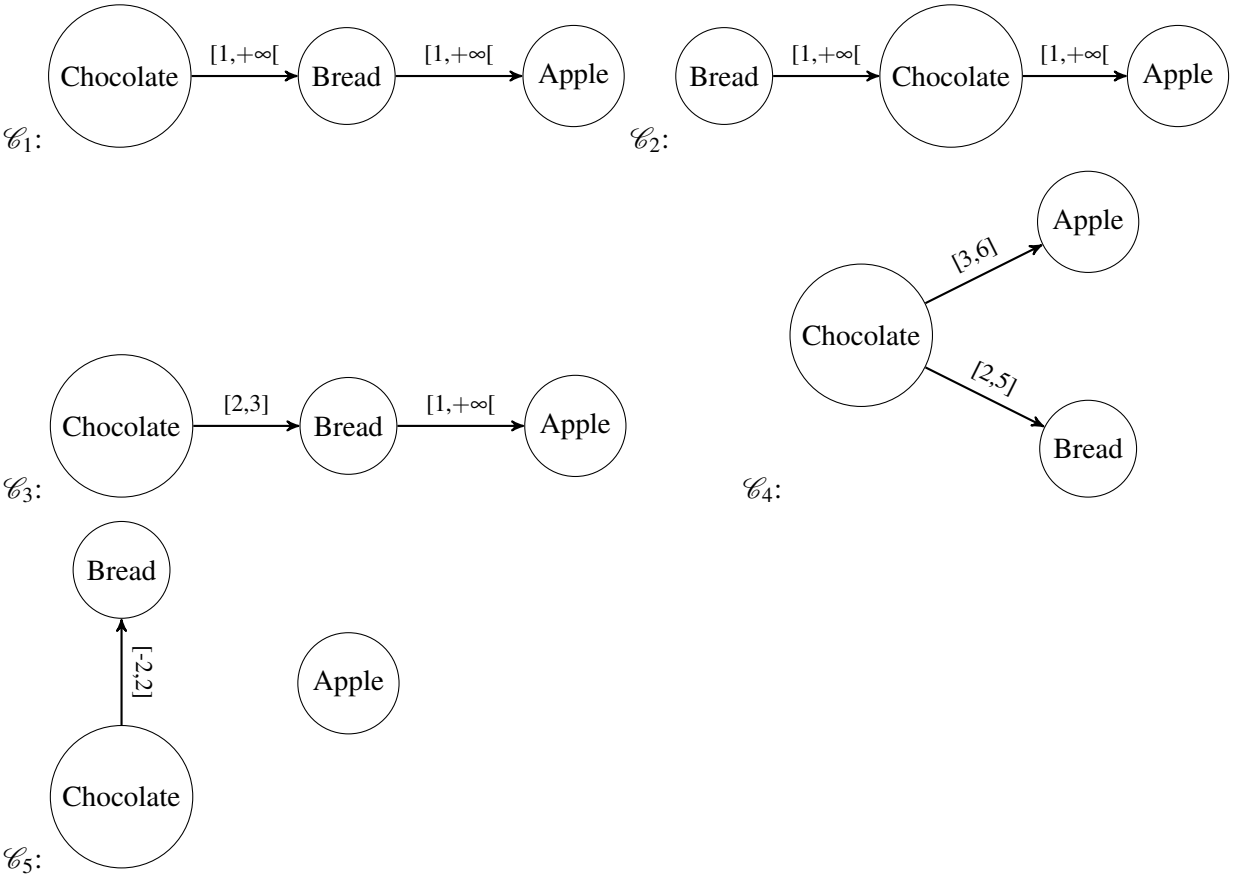


Figure 1.1: Chronicle illustrations of several temporal behaviors based on the buying events Apple, Bread and Chocolate.

1. the discriminant chronicle mining task
2. the study of patterns interpretability
3. a pharmaco-epidemiology case study

The first contribution is to propose the **discriminant chronicle mining task**. The extraction of frequent chronicles and some discriminant temporal patterns were previously studied but not this combination.

The upside of the chronicle model is to be more expressive than standard patterns like sequences. The main difference between these two models of pattern is the way to model the temporal dimension. Where the sequential patterns represent the temporal dimension only by sequentiality, *i.e.* what is the temporal order of the events, chronicles represent the time numerically and so events are not constrained to occur in a sequential order but to occur with timestamps satisfying temporal constraints. For example, the behavior of a person buying chocolate then bread and then apple can be represented by a sequential pattern. The sequential order describes the difference between another pattern that could be bread and then chocolate and then apple. These two behaviors can be represented by chronicles. They are illustrated by chronicles $\mathscr{C}_1$ and $\mathscr{C}_2$ on Figure 1.1, respectively. But we can also represent behaviors like buying bread between 2 or 3 days after chocolate and then apple. This behavior is illustrated by the chronicle $\mathscr{C}_3$ on Figure 1.1. There is in this behavior a temporal information in addition to the sequentiality. We can also describe non-sequential behavior with patterns like buying bread between 2 and 5 days after chocolate and

apple between 3 and 6 days after chocolate. This behavior is illustrated by the chronicle $\mathscr{C}_4$ on Figure 1.1. In this case bread and apple occurs after chocolate but there is no sequentiality between the last two events. Finally, a temporal constraint between two events can also be non-sequential. The chronicle model can describe the behavior of buying bread between 2 days before and 2 days after chocolate. This behavior is illustrated by the chronicle $\mathscr{C}_5$ on Figure 1.1.

The expressiveness of the chronicle model describes discriminant behaviors that sequential patterns cannot describe. This make discriminant chronicle mining interesting but it is an important challenge to effectively extract such patterns.

The second contribution concerns the **interpretability of the chronicle model**. The choice of the chronicle model was made for its expressiveness and its interpretability but the interpretability of chronicles was not studied before. The interpretability of the chronicle model is so studied and a generalization of the chronicle model was proposed to position the chronicle interpretability in a set of temporal patterns. This interpretability is highlighted for discriminant patterns because we can use them in a classification context. Thanks to this classification context, we compare the interpretability of several sets of patterns by taking into consideration their different accuracy to classify benchmark datasets. A trade-off between, the interpretability of the pattern model, and, the accuracy, is finally highlighted through those comparisons. This leads to study this trade-off considering the interpretability of the pattern themselves and, on the other side, the interpretability of the pattern set as a whole.

The third contribution is the extraction of epileptic behaviors through the **discriminant chronicle mining task on a case study** dataset: GENEPI. This dataset concerns epileptic people and is made of patients with a stable regular treatment during at least 1 years but were hospitalized for at least one epileptic seizure.

This work is a part of the PEPS project. The main goal of this project is to conduct pharmaco-epidemiology studies. The GENEPI study was one of them. The goal of this study was to further assess the association between seizure-related hospitalization and generic substitution of anti-epileptic drugs. A lot of work was done to process the dataset in order to statistically assess this hypothesis. We decide then to extract discriminant chronicles from this dataset:

- To confirm that no chronicle relative to the previous hypothesis is extracted (previous result was negative).
- To explore more deeply the dataset by extracting discriminant temporal behaviors that are interesting for clinicians.

Extracted results are finally discussed and ways to improve the extraction or extract other types of behaviors are proposed.

The following sections are gathered into two chapters: the state of the art and the contributions. State of the art of temporal and supervised pattern mining will be firstly presented. The three contributions are then presented and finally discussed.

Part I

State of the art

Chapter 2

Temporal pattern mining

One of the domains to which belongs the discriminant chronicle mining task is the temporal pattern mining. This part presents the different notions and works related to chronicles. In this chapter, the expressiveness of the chronicle model is detailed. This expressiveness is more important than the expressiveness of the other state-of-the-art patterns. Such difference of expressiveness is earned thanks to the management of the temporal dimension.

Temporal pattern mining is a research field that proposes and studies algorithms to extract interesting behaviors from temporal data. This is a specialization of the pattern mining domain where the temporal dimension is considered important to describe interesting behaviors.

Since there are various forms of temporal dataset for which specific algorithms have been proposed, this chapter is focused on the mining of some sequence sets. A temporal sequence, in this case, is a set of timestamped events. Additional information can be added to sequences that did not directly concern the event format like a label information.

More formally, let $\mathbb{E}$ be a set of event types and $\mathbb{T}$ be a temporal domain where $\mathbb{T} \subseteq \mathbb{R}$. An **event** is a couple (e,t) such that $e \in \mathbb{E}$ and $t \in \mathbb{T}$. We assume that $\mathbb{E}$ is totally ordered by $\leq_{\mathbb{E}}$. A **sequence** is a tuple $\langle SID, \langle (e_1,t_1),(e_2,t_2),...,(e_n,t_n)\rangle, L\rangle$ where SID is the sequence index, $\langle (e_1,t_1),(e_2,t_2),...,(e_n,t_n)\rangle$ a finite event sequence and $L \in \mathbb{L}$ where $\mathbb{L}$ is a label set. Sequence events are ordered by $\leq_{\mathbb{T}}$ defined as $\forall i,j \in [1,n],\ (e_i,t_i) \leq_{\mathbb{T}} (e_j,t_j) \Leftrightarrow t_i < t_j \vee (t_i = t_j \wedge e_i <_{\mathbb{E}} e_j)$.

SID	Sequence	Label
1	$(A,1)$, $(B,3)$, $(A,4)$, $(C,5)$, $(C,6)$, $(D,7)$	$+$
2	$(B,2)$, $(D,4)$, $(A,5)$, $(C,7)$	$+$
3	$(A,1)$, $(B,4)$, $(C,5)$, $(B,6)$, $(C,8)$,$(D,9)$	$+$
4	$(B,4)$, $(A,6)$, $(E,8)$	$-$
5	$(B,1)$, $(A,3)$, $(C,4)$	$-$
6	$(C,4)$, $(B,5)$, $(A,6)$, $(C,7)$, $(D,10)$	$-$

Table 2.1: Set of six sequences labeled with two classes $\{+,-\}$.

Table 2.1 represents a set of six sequences containing five event types (A, B, C, D and E) and labeled with two different labels $\mathbb{L} = \{+,-\}$. A concrete example of real dataset using this format could be a dataset of care pathways. In such dataset, the event types represent real medical events like drug deliveries, hospitalization or medical consultation. Let the event type A to be the delivery of drug A. The event $(A,1)$ is then the delivery of the drug A at the timestamp 1. Let the time unit be days. A sequence made of $(A,1)$, $(B,3)$ corresponds to the drug delivery of A followed two days later by the drug delivery of B. In such type of dataset, the most common use of the SID is to represent each patient by a sequence. Thus, the sequence 1 in Table 2.1 beginning by

$(A,1)$, $(B,3)$ corresponds to a patient for whom we know that a drug A was delivered and two days later a drug B was delivered. If this dataset concerns patients that were exposed to a disease, the $+$ label could correspond to a positive outcome and the $-$ label to a negative one. We then know that patient 1 was treated with specific drugs at specific timestamps and was healed. The label information is presented here with the format of sequences but will be really used in the next chapter for supervised pattern mining (chapter 3).

This representation of sequences is not the only existing but is adapted to the use of temporal pattern mining done in the next part of this thesis. As an example of other representation of sequences, the sequences are represented in the format $\langle(A),(B,C),(D)\rangle$ in [Agrawal and Srikant, 1995] where the time dimension is reduced as an order between the events. In this example $(A),(B,C)$ means that A occurs before (B,C) and (B,C) means that B and C occurs at the same time.

2.1 Sequential pattern mining

Sequential patterns only take the temporal order of events into account. Reviews [Masseglia et al., 2004, Mabroukeh and Ezeife, 2010, Mooney and Roddick, 2013] did the state of the art of the sequential pattern mining, the different associated algorithmic approaches and their applications. This section is improved by these previous reviews while focusing on the mining of sequence sets. All those algorithms are grouped together because they extract subsequence. In their works, sequences are both data and the pattern shape. I will refer to these sequences as *standard sequential patterns* in order to dissociate the sequences as data and sequences as patterns.

According to [Mooney and Roddick, 2013], the sequential pattern mining problem was first addressed by [Agrawal and Srikant, 1995] and was defined as follows:

> "Given a database of sequences, where each sequence consists of a list of transactions ordered by transaction time and each transaction is a set of items, sequential pattern mining is to discover all sequential patterns with a user-specified minimum support, where the support of a pattern is the number of data-sequences that contain the pattern."

We could then highlight two families of algorithms dedicated to extract such patterns: Apriori-based and pattern growth algorithms. The main difference between these two families introduced by the pattern growth algorithms is the candidate generation step. In practice, it is the search space exploration strategy that differentiate those families. The Apriori-based algorithms use mainly a breadth-first search strategy what leads to an important candidate generation step. On the other hand, the pattern growth algorithms use mainly a depth-first search strategy.

This taxonomy is similar to that in the other review [Mabroukeh and Ezeife, 2010]. In the first review a distinction is made for the Apriori-based algorithms by separating the horizontal and the vertical database formats. On the other hand, the review made in [Mabroukeh and Ezeife, 2010] focus more on the difference with a third algorithm family: the early-pruning algorithms.

2.1.1 Standard sequential patterns

We will call standard sequential patterns, patterns for which each pair of events or itemsets of the pattern is constrained to occur in a specific order. This is the type of pattern introduced in the first paper on sequential pattern mining [Agrawal and Srikant, 1995] and more commonly called sequence. In this section we then present the definition of standard sequential patterns and some algorithms dedicated to extract such patterns with the same taxonomy as [Mooney and Roddick, 2013].

2.1.1.1 Definition

In this part we will use the standard definition of sequential patterns [Masseglia et al., 1998] and adapt it to our previous definition of dataset illustrated in Table 2.1.

Definition 1. Let $\mathscr{I} = \{i_1, i_2, ..., i_m\}$ be a set of literals called items. An **itemset** is a non-empty set of items. A **sequence** s is a list of itemsets ordered according to their timestamp. It is denoted by $\langle s_1, s_2, ..., s_n \rangle$ where s_j is an itemset. A k-sequence is a sequence of k items (or of length k). A sequence $\langle s_1, s_2, ..., s_n \rangle$ is a subsequence of another sequence $\langle s'_1, s'_2, ..., s'_n \rangle$ if there exist integers $i_1 < i_2 < ... < i_n$ such that $s_1 \subseteq s'_{i_1}, s_2 \subseteq s'_{i_2}, ..., s_n \subseteq s'_{i_n}$. For short, we denote that s is a subsequence of s' by $s \subseteq s'$.

The original Definition 1 describes what is a sequence and so in this section, a standard sequential pattern. The set of literals $\mathscr{I}$ corresponds to our set of event types $\mathbb{E}$. It is the vocabulary used to generate the dataset. From this vocabulary we can compose complex events that are called itemset. These itemsets are sets of events constrained to occur at the same timestamp. An itemset of events A, B and C is denoted (A,B,C). Two events $(A,1)$ and $(B,1)$ will so be represented as the itemset (A,B). It is worth noticing that itemsets are sets and not multisets *i.e.* an event can occur at most once per timestamp.

A standard sequential pattern is a totally ordered set of those itemsets. For example, the sequence $s_1 = \langle (A,B,C),(D) \rangle$ describes a case where observe A, B and C at the same timestamp and then D *i.e.* $t_A = t_B = t_C < t_D$. An event may occur several times in a standard sequential pattern if it occurs in several itemsets. An example of such pattern is $\langle (A,B,C),(A) \rangle$.

A subsequence is a subset of events preserving sequential constraints. For example, $\{A,B,D\}$ is a subset of $\{A,B,C,D\}$ and so if we preserve the sequential constraints $t_A = t_B < t_D$ we obtain $\langle (A,B),(D) \rangle$ a subsequence of s_1. On the other hand $s_2 = \langle (A,B,D) \rangle$ is not a subsequence of s_2 because, for s_2, $t_A = t_D$ but for s_1, $t_A < t_D$.

From the definition of subsequence, the **support** $supp(s)$ of a pattern s is defined as the number of sequences in which s occurs. If a pattern s is a subsequence of a sequence s' of the dataset, we can say that s occurs in s'. Then the support of s in a sequence set $\mathscr{S}$ can be formally defined as $supp(s) = |\{s' \in \mathscr{S} | s \subseteq s'\}|$. We can use this formal definition in our case by transforming our sequences in the sequential pattern format. For a sequence s, for each timestamp t_j for which we observe at least one event (e_i, t_j), we can create an itemset regrouping all event types with same timestamp. We then call a pattern s **frequent** if $supp(s) \geq \sigma_{min}$ where σ_{min} is specified by the user. Such parameter can also be called σ [Masseglia et al., 1998] or *min_supp* [Mooney and Roddick, 2013] in the literature. Alternatively we call frequency the proportion of sequences in which the pattern occurs.

2.1.1.2 Apriori-based algorithms

This section presents the different algorithms based on a candidate generation step to extract the frequent set of standard sequential patterns in a sequence set.

The first algorithms of this family were based on horizontal databases. A horizontal database is a database for which the sequences are ordered using the temporal dimension. Table 2.1 illustrating such database type.

The first Apriori algorithms dedicated to mine frequent sequential patterns are proposed in [Agrawal and Srikant, 1995]. In the problem statement of the paper, the definition of a maximal sequence is given. A sequence s is said **maximal** if s is not contained in any other frequent sequence. Using the previous definition of a subsequence, a maximal sequence s is defined such that $\nexists s',$, s' is frequent, $s \subseteq s'$ and $s' \neq s$. This definition is then used to reduce the number of extracted patterns.

Customer Id	Transaction time	Items Bought
1	June 25 '03	30
1	June 25 '03	90
2	June 10 '03	10,20
2	June 15 '03	30
2	June 20 '03	40,60,70
3	June 25 '03	30,50,70
4	June 25 '03	30
4	June 30 '03	40,70
4	July 25 '03	90
5	June 12 '03	90

Customer Id	Customer Sequence
1	$\langle(30),(90)\rangle$
2	$\langle(10,20),(30),(40,60,70)\rangle$
3	$\langle(30,50,70)\rangle$
4	$\langle(30),(40,70),(90)\rangle$
5	$\langle(90)\rangle$

Table 2.2: On the left, the customer transaction database example proposed in [Agrawal and Srikant, 1995]. On the right, the transformed customer sequence version.

Customer Id	Transformed Customer Sequence	Mapped Transformer Customer Sequence
1	$\langle\{(30)\},\{(90)\}\rangle$	$\langle\{1\},\{5\}\rangle$
2	$\langle\{(30)\},\{(40),(70),(40,70)\}\rangle$	$\langle\{1\},\{2,3,4\}\rangle$
3	$\langle\{(30),(70)\}\rangle$	$\langle\{1,3\}\rangle$
4	$\langle\{(30)\},\{(40),(70),(40,70)\}\rangle$	$\langle\{1\},\{2,3,4\}\rangle$
5	$\langle\{(90)\}\rangle$	$\langle\{5\}\rangle$

Table 2.3: The transformed custom sequences as presented in the original example. The dataset before the transformation is that one shown in Table 2.2. The itemset set used for the transformation is $\{(30),(40),(70),(40,70),(90)\}$ mapped with $\{1,2,3,4,5\}$.

The proposed algorithms are all split in five phases: (i) Sort Phase, (ii) Litemset Phase, (iii) Transformation Phase, (iv) Sequence Phase and (v) Maximal Phase. The first phase is a processing of the data. The goal of this phase is to process a customer transaction database like illustrated by Table 2.2 and to generate a sequence set. Table 2.2 illustrated such database and the associated sequence set. This phase is optional if the data are already sequence sets.

The second phase is called Litemset Phase for large itemset phase. In this paper, a large itemset or by extension a large sequence is simply an itemset or a sequence satisfying the minimal support constraint. It is commonly called a frequent itemset or a frequent sequential pattern. In this phase, the algorithm enumerates the set of all the frequent itemsets $\mathscr{F}_1$ and by extension the set of all the frequent 1-sequences since this is $\{\langle f\rangle | f \in \mathscr{F}_1\}$. The set of all frequent itemsets $\mathscr{F}_1$ is denoted L in the original paper but the notation is modified here to prevent the confusion with the label L in the sequence definition. The example given in the original paper sets the minimal frequency

Algorithme 1 The $next(k)$ function heuristic given for *AprioriSome*.

Require: k : integer, $|\mathscr{C}_k|$: candidate k-sequence set size, $|\mathscr{F}_k|$: frequent k-sequence set size
1: $hit_k \leftarrow |\mathscr{F}_k|/|\mathscr{C}_k|$
2: **if** $hit_k < 0.666$ **then**
3: **return** $k+1$
4: **if** $hit_k < 0.75$ **then**
5: **return** $k+2$
6: **if** $hit_k < 0.80$ **then**
7: **return** $k+3$
8: **if** $hit_k < 0.85$ **then**
9: **return** $k+4$
10: **return** $k+5$

σ_{min} to 25% of the number of sequences. It is similar to say that the patterns must occur in at least two sequences to be frequent because the sequence set contains five sequences and the support of a sequential pattern is inevitably a positive integer. The frequent itemsets extracted from the sequence set in Table 2.2 with such σ_{min} are (30), (40), (70), (90) and $(40,70)$. To match those itemsets in the data in the following step, they are all associated with contiguous integers. The goal of such procedure is to optimize the matching of a specific itemset in the following phase. The example proposes to associate (30), (40), (70), $(40,70)$, (90) with $1,2,3,4$ and 5 respectively.

The third phase, the Transformation Phase, uses frequent itemsets to transform the database before the mining phases. Table 2.3 shows the transformation of Table 2.2 with the previously extracted itemsets (*i.e.* the frequent ones). The transformation is done by replacing each itemset i of a sequence by the set of frequent itemsets included in i. For example, it can be noticed on Table 2.3, that the second sequence of Table 2.2 $\langle(10,20),(30),(40,60,70)\rangle$ becomes $\langle\{(30)\},\{(40),(70),(40,70)\}\rangle$ and then $\langle\{1\},\{2,3,4\}\rangle$ using the associated integers. The advantage of this phase is twofold:

- The infrequent items are directly pruned.
- Each itemset can be easily matched because it is now represented as an item.

The transformation can also be done using the mapping integers to match efficiently the itemsets occurring or not in each transaction. This transformation is also shown for the example on table 2.3.

The fourth phase, the Sequence Phase, is the mining phase. Three versions of this phase are given to produce three algorithms: *AprioriAll*, *AprioriSome* and *DynamicSome*. The difference between *AprioriAll* and the two others is that it is designed to compute all the frequent sequences while the others are designed to only compute the maximal sequences. The Maximal Phase is so designed for *AprioriAll* and is the pruning of all the non-maximal sequences from the extracted frequent sequence set.

- *AprioriAll*: This version of the algorithm is based on a recursive candidate generation. Each candidate k-sequence set is generated from the frequent $(k-1)$-sequence set. The mining procedure is initialized from the frequent 1-sequences computed in the Litemset Phase until a k for which none of the candidate sequences are frequent. A k-sequence is generated as a candidate if all it $(k-1)$-subsequences are frequent. For example, the sequence $\langle1,2,3,4\rangle$ is a candidate if $\langle1,2,3\rangle$, $\langle1,2,4\rangle$ and $\langle2,3,4\rangle$ are frequent. The support of a candidate sequence is finally tested by a pass on the data.

- *AprioriSome*: This version of the algorithm try to improve the mining of maximal sequential patterns done by *AprioriAll*. This is done by using a $next(k)$ function where $next(k) = k+1$ is the default function used by *AprioriAll*. This function will choose for which candidate k-sequences the algorithm will traverse the data to compute the support. The frequent k-sequence is denoted $\mathscr{F}_k$ and the candidate k-sequence set is denoted $\mathscr{C}_k$.

  For example, using the function $next(k) = k+2$ means that $\mathscr{C}_2$ and $\mathscr{C}_3$ are computed from $\mathscr{F}_1$ but only $\mathscr{F}_3$ is computed by a pass on the data. The candidate set $\mathscr{C}_2$ is not pruned by a pass on the data. Thereby, $\mathscr{C}_3$ is potentially bigger if $\mathscr{F}_2$ would be computed but it saves at least an iteration on the data. The next k for which $\mathscr{F}_k$ will be computed will then be 5.

  When this forward generation stops, all the sets $\mathscr{F}_k$ for which the sets $\mathscr{C}_k$ were computed have to be computed too. The backward generation going from the maximal k to 1 begins then. As only the maximal sequences have to be extracted, the candidate sequences of $\mathscr{C}_{k-1}$ that are subsequences of at least one sequence of $\mathscr{F}_k$ can be pruned without traversing the

data. For example, if $\mathscr{F}_3$ is known and $\mathscr{F}_2$ as to be computed, all the sequences of $\mathscr{C}_2$ that are subsequences of at least one sequence of $\mathscr{F}_3$ can be pruned. This can be done using the definition of a maximal sequential pattern. As a maximal sequential pattern is a frequent pattern that is not a subsequence of another frequent pattern, the subsequences of a maximal pattern are not maximal. Then, the subsequences of $\mathscr{F}_k$ included in $\mathscr{C}_{k-1}$ cannot be maximal and so are not interesting for the task of maximal sequential pattern mining.

This step is recursively repeated until all the $\mathscr{F}_k$ sets are extracted. The intuition behind the use of a forward step for some $\mathscr{F}_k$ and a backward step for the others is to balance the trade-off between the time wasted in enumerating non-maximal sequences versus enumerating extensions of small candidate sequences. The $next(k)$ function used by the authors in their experiments is given by algorithm 1.

- *DynamicSome*: The idea of *DynamicSome* is the same as for *AprioriSome* but without generating the $\mathscr{C}_k$ sets for which the $\mathscr{F}_k$ sets are not computed. The example given is based on a $step$ parameter set up to 3. It is equivalent to use the function $next(k) = k+3$ in *AprioriSome*. Unlike for *AprioriSome*, the algorithm starts with an initialization step. For $step = 3$, the initialization computes $\mathscr{C}_2$ and $\mathscr{C}_3$ and then $\mathscr{F}_3$. This frequent initial sequence set will then be used for each candidate generation step. The forward step is then similar to the *AprioriSome* forward step but all the candidate sets are not generated: to compute $\mathscr{F}_{k+step}$, $\mathscr{C}_{k+step}$ is computed by joining sequences of $\mathscr{C}_k$ and $\mathscr{C}_{step}$. Finally, the backward step differs from *AprioriSome* in the computation of $\mathscr{C}_k$, that has to be computed if missing. The authors reported in the experiments section that this algorithm generates too many candidates and ran out of memory.

These algorithms are not the most efficient in terms of memory consumption and computational complexity and more recent algorithms are preferred to extract standard sequential patterns. However, they highlight the difficulty encountered when trying to explore the search space of frequent sequential patterns.

The algorithm *GSP* [Srikant and Agrawal, 1996] is proposed by the same authors to overcome some limitations they observed in their previous algorithms. The extensions of the first Apriori algorithms into *GSP* included time constraints: minimal and maximal gap constraints, sliding windows constraint and item taxonomies. To introduce taxonomies in the algorithm, the authors propose the new definition 2 for the items.

Definition 2. Let $\mathscr{I} = \{i_1, i_2, ..., i_m\}$ be a set of literals, called items. Let $\mathscr{T}$ be a directed acyclic graph on the literals. An edge in $\mathscr{T}$ represents an *is-a* relationship, and $\mathscr{T}$ represents a set of taxonomies. If there is an edge in $\mathscr{T}$ from p to c, p is called a *parent* of c and c a *child* of p (p represents a generalization of c.) The taxonomy can be modeled as a directed acyclic graph (DAG) rather than a tree to allow for multiple taxonomies. A node $\widehat{x}$ is an *ancestor* of x (and x a *descendant* of $\widehat{x}$) if there is an edge from $\widehat{x}$ to x in transitive-closure($\mathscr{T}$).

The definition of the support is also extended. A transaction T (an itemset in a sequence) *contains* an item $x \in \mathscr{I}$ if x is in T or x is an ancestor of some item in T. By extension, a transaction T contains an itemset $y \subseteq \mathscr{I}$ if T contains every item of y.

The sliding window proposed in *GSP* is a generalization of the itemset definition. The items of an itemset are no more constrained to occur at the same time but in a sliding window. The parameter *window-size* defined by user is the maximal size of those windows. An itemset $y = (i_1, i_2, ...i_n)$ satisfies this constraint if the difference between the timestamps of its first item t_{i_1} and its last item t_{i_n} is lower or equal to *window-size*. More formally $t_{i_n} - t_{i_1} \leq$ *window-size*. For example, the pattern $\langle (A,B), C \rangle$ occurs in sequence 1 of Table 2.1 with *window-size* $= 2$ because there is a mapping

with $(A,1),(B,3),(C,5)$ where $t_B - t_A \leq$ *window-size* so A and B can be considered to occur at the same time and $t_C > t_B$ what makes C following B.

The minimal and maximal gap constraints are similar to the sliding window constraint but they do not compare the first and last item timestamps of the same itemset but of two following itemsets. Let $s = \langle s_1, s_2, ..., s_n \rangle$, *start-time*$(s_i)$ the timestamp of the first item of s_i and *end-time*(s_i) the timestamp of the last item of s_i, the minimal gap constraint is defined as *start-time*(s_i) − *end-time*$(s_{i-1}) >$ *min-gap*, $2 \leq i \leq n$. The maximal gap constraint is defined as *start-time*(s_i) − *end-time*$(s_{i-1}) \leq$ *max-gap*, $2 \leq i \leq n$. It is worth noticing that the minimal gap constraint is a strict inequality but not the maximal constraint. It is a convenient definition to set *min-gap* $= 0$ for no minimal gap constraint but still avoiding the overlap of two itemsets *i.e.* the last item of the first itemset occurs before the first item of the second itemset. In the example with the pattern $\langle (A,B),C \rangle$ and the sequence 1 of table 2.1 the pattern matches the mapping $(A,1),(B,3),(C,5)$ for *max-gap* $= 2$ because $t_C - t_B \leq 2$ but not for *min-gap* $= 2$ because it does not satisfy because $t_C - t_B > 2$.

Those new definitions imply then more computational time to check if a pattern occurs in a sequence.

The mining phase works then like for the previous Apriori algorithms. Candidate sequences are generated in a first step and then pruned if they are detected infrequent. To adapt this paradigm to the sliding window and gap constraints, the notion of *contiguous subsequence* is introduced. This notion is introduced by definition 3.

Definition 3. Given a sequence $s = \langle s_1, s_2, ..., s_n \rangle$ and a subsequence c, c is a *contiguous* subsequence of s if any of the following conditions holds:

- c is derived from s by dropping an item from either s_1 or s_n.
- c is derived from s by dropping an item from an element s_i which has at least 2 items.
- c is a contiguous subsequence of c', and c' is a contiguous subsequence of s.

For example, consider the sequence $s = \langle (1,2),(3,4),(5),(6) \rangle$. The sequences $\langle (2)(3,4)(5) \rangle$, $\langle (1,2),(3),(5),(6) \rangle$ and $\langle (3),(5) \rangle$ are some of the contiguous subsequences of s. However, $\langle (1,2),(3,4),(6) \rangle$ and $\langle (1),(5),(6) \rangle$ are not.

Based on this definition, the candidate k-sequences set is generated by joining the frequent $(k-1)$-sequences and then pruning the candidate for which a contiguous subsequence is not frequent. The interest of this type of subsequence is that if the candidate sequence satisfies the gap constraints, the contiguous subsequences satisfy it too. So if a contiguous subsequence is not frequent, neither is the candidate sequence.

More constraints and techniques to extract standard sequential patterns were studied after this paper. Some monotone and anti-monotone constraints were listed and reviewed in [Pei and Han, 2002]. Those properties are important in algorithm optimization as it allows to efficiently prune the pattern search space. Those properties are linked to the pattern lattice as presented latter for the SPADE algorithm [Zaki, 2001].

The PSP [Masseglia et al., 1998] algorithm achieve the exact same task as GSP. The difference between the two algorithms is the organization of the candidate sequences. A prefix-tree is used to optimize the memory space and the access time to the candidate sequence set.

The SPIRIT [Garofalakis et al., 1999] algorithm was proposed to tackle the pattern flood problem and allow the user to control the shape of the extracted pattern set. The purpose of SPIRIT is to mine sequential patterns constrained by regular expressions. The user uses SPIRIT to extract sequential patterns satisfying a regular expression that (s)he gives as parameters. The algorithm

is build as GSP and the use of the regular expression constraint in the mining process is similar as for the gap constraints. The algorithm is proposed with different heuristics for checking the satisfaction of the constraint during the candidate generation step. Those different heuristics are used to balance the trade-off between the complexity of the constraint checking and generated a great number of candidates.

A different use of the databases is handled with MFS [Zhang et al., 2001]. When the previous algorithms only consider extracting once the frequent sequence set of a database, MFS is designed to extract the frequent sequence set using a previous extraction. The idea is that if the user wants to mine the frequent patterns periodically, the frequent pattern set can be different if the database is updated but will remain similar as the previous extraction. The algorithm then has to be used each time the user wants to update the frequent pattern set but all the previous extracted knowledge has not to be forgotten. The algorithm generates so, candidates using the previous extracted set. This allows to generate fewer candidates and so to do less data scans in comparison with GSP. The major source of efficiency improvement presented by the authors is the early checking of supports of the long sequences.

Like PSP for GSP, the RE-Hackle [Albert-Lorincz and Boulicaut, 2003] try to improve the efficiency of the SPIRIT algorithms by using more efficient structure. In this case, RE-Hackle uses what the authors called a Hackle-tree to represent the regular expression. Those RE-Hackle trees are trees for which the nodes are operators or subsequences called atomic sequences. This representation allows a more efficient evaluation of the potential satisfaction of the regular expression by a pattern. They finally compare the proposed heuristic associated with such trees with two heuristics of SPIRIT and conclude that RE-Hackle boosts the performance of sequential pattern mining under regular expressions.

MSPS [Luo and Chung, 2008] extends the improvement of GSP introduced by PSP. To improve the breadth-first search strategy used by the first Apriori algorithm family, a sampling step is done to generate local maximal frequent patterns. The goal of this sampling is to forbid the candidate generation of the subsequences of those maximal patterns. Indeed, if a pattern is frequent, all its subsequences are frequent and then, their supports not have to be checked.

SID	Sequence	Label
1	$(A,1)$, $(A,4)$, $(B,3)$, $(C,5)$, $(C,6)$, $(D,7)$	$+$
2	$(A,5)$, $(B,2)$, $(C,7)$, $(D,4)$	$+$
3	$(A,1)$, $(B,4)$, $(B,6)$, $(C,5)$, $(C,8)$,$(D,9)$	$+$
4	$(A,6)$, $(B,4)$, $(E,8)$	$-$
5	$(A,3)$, $(B,1)$, $(C,4)$	$-$
6	$(A,6)$, $(B,5)$, $(C,4)$, $(C,7)$, $(D,10)$	$-$

Table 2.4: Vertical format of Table 2.1.

A new approach to extract the pattern set is proposed with the algorithm SPADE [Zaki, 2001]. The SPADE algorithm is based on a lattice-based approach. The hyper-lattice is defined by the subsequence relation on the sequence set. Examples of such lattice are given by Figure 2.1 or by Figure 2.2 for a sublattice bounded by a given maximal pattern. This representation shows that, as explained for AprioriAll, all subsequences of a frequent sequence are frequent. But it also makes apparent that the search has not to be restricted to a purely bottom-up search. The authors propose then to use other strategies to extract the frequent sequences than the breath-first search like the depth-first search. Consequently, it improves memory management because the whole candidate sequence set has not to be saved.

The second contribution of the paper is to represent the data in a vertical format. The horizontal

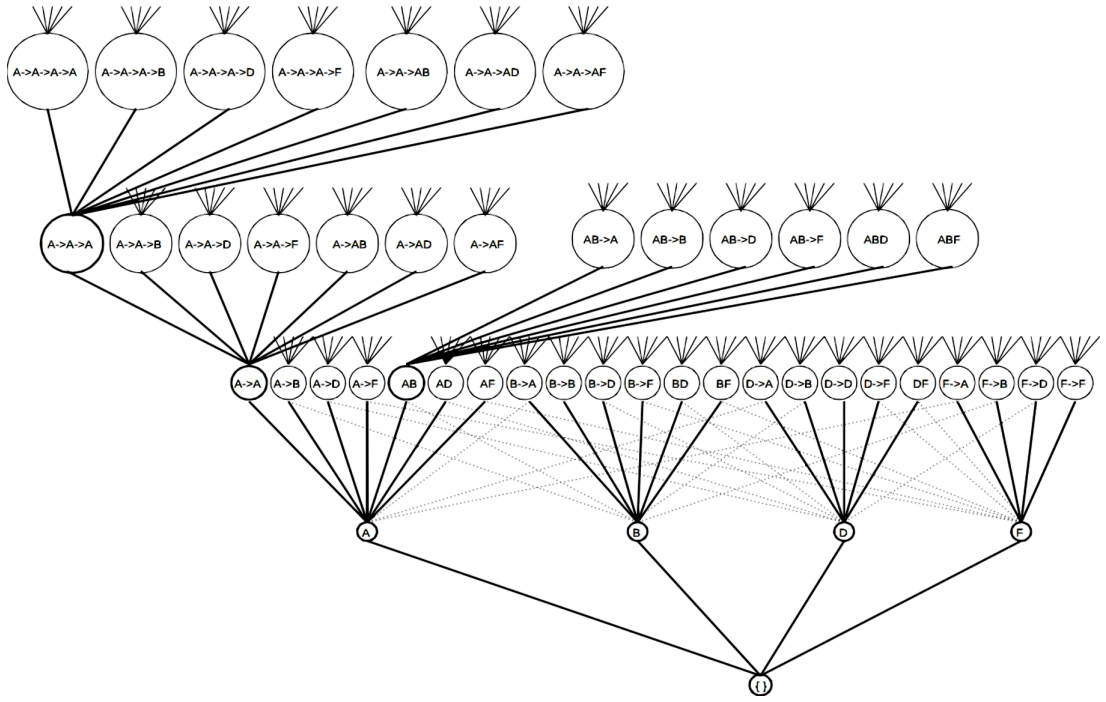


Figure 2.1: An example of hyper-lattice of sequential patterns. The order is defined by the subsequence relation. This figure was extracted from [Zaki, 2001].

format is the format illustrated by Table 2.1. In this format the events contained by a sequence are ordered firstly by their timestamp and then by their symbols. It is the opposite of the vertical format: the events are firstly ordered by their symbols and then by their timestamps. Table 2.4 illustrates the vertical format of Table 2.1.

In this format, all the events in a sequence are not ordered firstly by their timestamp but by their symbol.

Based on this algorithm, the authors also proposed cSPADE [Zaki, 2000] that allows to mine sequential patterns under constraints. The constraints proposed by cSPADE are the followings: (i) sequence length limitation, (ii) minimum or maximum gap constraints, (iii) time windows, (iv) item constraints and (v) sequences predictive of one or more classes. The time window constraint is different with the sliding window of GSP because it is a window in which the whole sequence have to occur. The predictive sequences can only be mined from dataset where each sequence has a label. This last constraint will be discussed with more details in Section 3.2.

The SPAM [Ayres et al., 2002] algorithm can be seen as an improvement of SPADE like PSP is an improvement of GSP. SPAM uses a depth-first search with pruning mechanisms for optimization and a vertical representation of the databases. One of the improvements is done in the database design that uses a bitmap representation. One limitation of this approach is that it is only useful if all the data fit into the main memory.

The CCSM [Orlando et al., 2004] algorithm is an improvement of SPADE only dedicated for the breadth-first search strategy. To overcome many problems associated with this type of strategy, CCSM uses a cache system that stores immediate id-lists for future reuse.

The LAPIN [Yang et al., 2005] and LAPIN-SPAM [Yang and Kitsuregawa, 2005] algorithms are based on the observation that if the last position of an item α is smaller than, or equal to, the position of the last item in a sequence s, then item α cannot be appended to s as a $(k+1)$-sequence extension in the same sequence. To exploit this observation, a specific table contains the last position information for each item in each sequence. The LAPIN-SPAM combines this

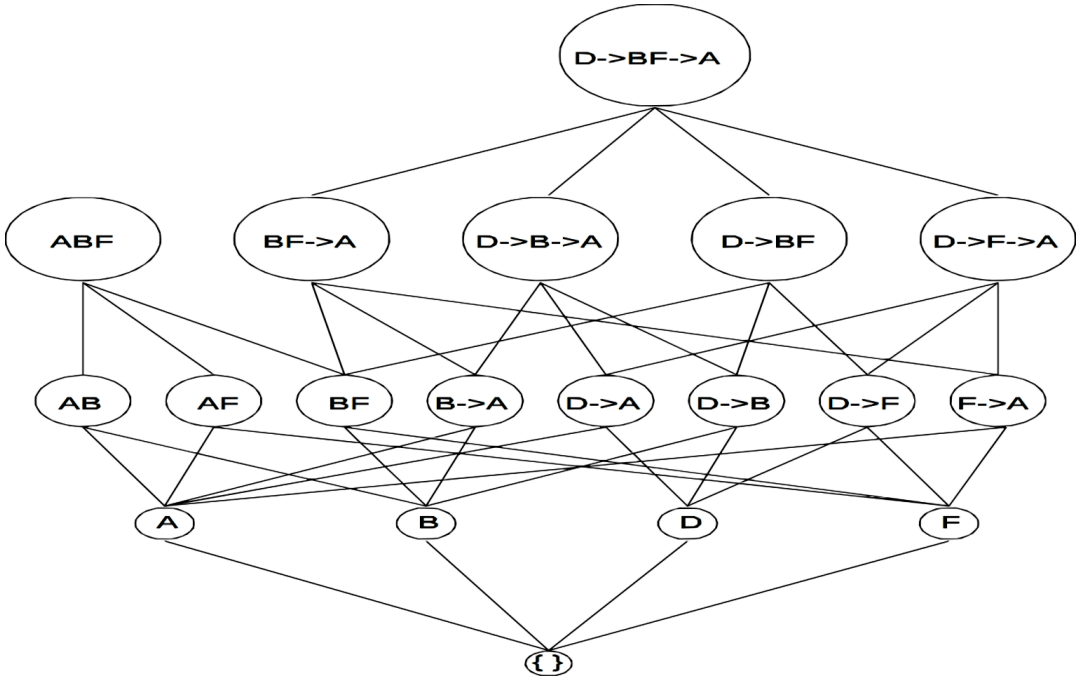


Figure 2.2: An example of sublattice bounded by the maximal sequential pattern $\langle (D), (BF), (A) \rangle$. This figure was extracted from [Zaki, 2001].

improvement with the SPAM algorithm.

2.1.1.3 Pattern growth algorithms

Many limitations of AprioriAll were overcome with the next iterations of sequential pattern mining algorithms. Indeed, the memory limitation can be overcome thanks to a depth-first search strategy and numerous improvements were proposed to obtain faster algorithms. But in the worst case, the candidate generation step will still suffer from producing a set with an exponential size. As a scan of the data has to be done to check every length of candidate set, such generation is expensive in time complexity. The pattern growth paradigm removes the need for the candidate generation and for the pruning steps that are used in the Apriori type algorithms. This is done by compressing the database representing the frequent sequences into a frequent pattern tree. This tree is finally used to generate the subdatabases containing subsets of the frequent sequence set and those databases can be mined separately.

The FreeSpan [Han et al., 2000a] algorithm is the first pattern growth algorithm. To generate the projected subdatabases, FreeSpan starts like other algorithms with the enumeration of all frequent 1-sequences. The set of frequent 1-sequences (items) is then ordered by support in descending order. The given example of such list is $f_{list} = \langle a:4, b:4, c:4, d:3, e:3, f:3 \rangle$.

Each 1-sequences will then be used as a pivot to generate a database containing this item and the previous item in the list. For example, the database based on a will contain a. The database based on b will contain a and b. To avoid a redundant extraction of a same pattern in each base, the pivot 1-sequence has to be a subsequence of the extracted patterns. For example, all the patterns extracted from the base with b as pivot will contain b. Thereby, there is a balance between (i) the databases with few items from which the most frequent patterns will be extracted and (ii) the databases with a lot of items but from which only patterns based on the less frequent items will be extracted.

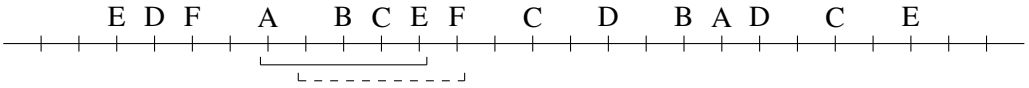


Figure 2.3: trace representation and two windows example as illustrated in [Mannila et al., 1997]. The window size is set to 5 in this example.

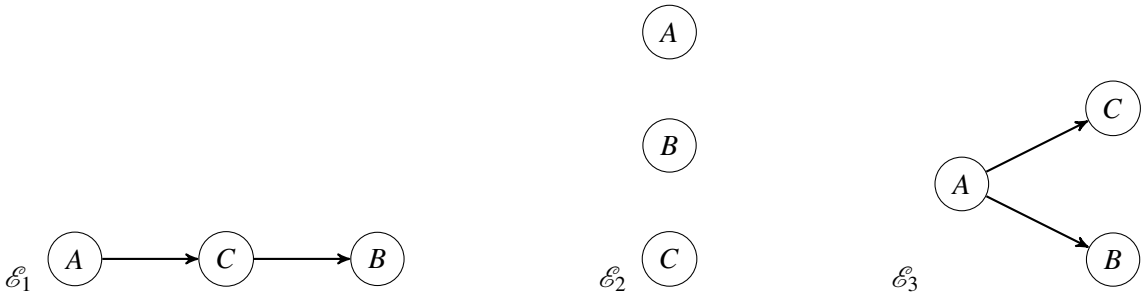


Figure 2.4: Three examples of episodes. Episode $\mathscr{E}_1$ is a serial episode. Episode $\mathscr{E}_2$ is a parallel episode.

The generation of the projected databases is then improvement with the PrefixSpan [Han et al., 2001] algorithm.

Through this overview of the state of the art, it can be noticed that standard sequential patterns were much studied. Optimized algorithms are efficiently implemented for this type of pattern and can be used to extract patterns from big set of data like in industrial environment. As an example, the FP-growth [Han et al., 2000b] and PrefixSpan [Han et al., 2001] algorithms were both implemented for Spark [Zaharia et al., 2010] in the MapReduce [Dean and Ghemawat, 2008] framework. The adaptation of FP-growth is described in the literature [Li et al., 2008].

The extraction of knowledge from datasets using such algorithms is, however, limited by the standard sequential pattern model. The total order between the whole set of events and the reduction of the temporal domain as sequentiality do not allow to extract some interesting behaviors. More expressive patterns were so needed to extract knowledge from temporal data where the temporal information is strongly linked with the interesting behaviors.

2.1.2 Partially ordered patterns

The total sequential order defined on the itemset of a sequential pattern can be too constrained to extract some recurrences in a sequence set. Some patterns like *episodes* [Mannila et al., 1997], when extracted from traces, or *partially ordered patterns* [Fabrègue et al., 2013], when extracted from sequence set, were proposed to relax this constraint. In [Achar et al., 2012], a framework is proposed to generalize and unify the different algorithm dedicated to frequent episode mining based on the Apriori framework.

Such type of patterns was initially introduced to mine traces [Mannila et al., 1997]. The difference between trace and sequence set is that for traces the goal is to mine patterns occurring in one single long sequence. Support computation is different for those two types of data. For example, in the method proposed for the introduction of episodes is the number of windows in which the episode occurs. An example of such stream and window is by Figure 2.3. Two windows can be overlapping as illustrated by the figure 2.3 but they have a fixed size defined by a user parameter.

An episode ϕ is defined by a graph $\phi = (V, \leq, g)$ such that V is a set of nodes, $\leq$ is a partial order on V and $g : V \rightarrow \mathbb{E}$ is a mapping function associating each node with an event type in $\mathbb{E}$.

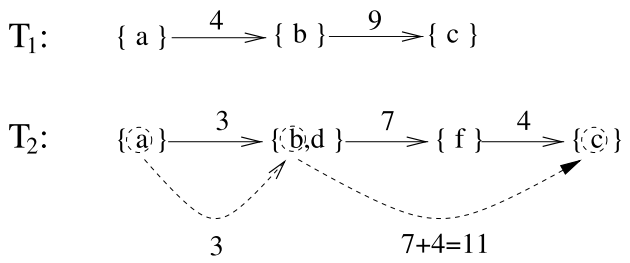


Figure 2.5: Two examples of temporal-annoted sequences as they are illustrated in their original paper [Giannotti et al., 2006]. The temporal constraints of those TAS are integers.

Figure 2.4 illustrated episodes mapping the events A, B and C. Two specific types of episodes are considered: the serial and the parallel episodes. An episode is a serial episode if and only if the partial order relation $\leq$ is a total order. Serial episodes are equivalent to the sequential patterns presented in the previous section. In the opposite way, an episode is a parallel episode if and only if the partial order $\leq$ is trivial *i.e.* there are no constraints on the relative order of events.

The extraction of the whole set of frequent episodes or partially ordered patterns is generally more complex than the whole set of frequent standard sequential patterns. Thereby, it can be useful to extract a subset of the frequent episodes. The definition of parallel and serial episodes are then often used to extract such subset. The patterns extracted are so more expressive than the standard sequential patterns without being too complex to be extracted.

Nonetheless, such type of patterns lacks of a numerical representation of the time or at least a representation more expressive than the sequentiality.

2.2 Mining of patterns including numerical temporal information

Even if sequential pattern mining is the most studied subdomain of temporal pattern mining, some approaches tried to include a numerical information in the patterns. Standard sequential patterns were extended is temporal-annoted sequences (TAS) in [Giannotti et al., 2006]. The difference between standard sequential pattern and TAS is that the gap duration between two events is constrained by a temporal constraint. This constraint can be set up by a number to fix the gap or by intervals. TAS use numbers for temporal constraints. It is illustrated by Figure 2.5. These numbers are positive what makes TAS be standard sequential patterns constrained by temporal constraints between the events.

Other approaches propose to extract patterns by constraining the absolute occurrences of events. In [Guyet and Quiniou, 2011], a temporal pattern corresponds to a set of events constrained by absolute temporal intervals. As the events are allowed to have a starting and an ending timestamps, the matching of a pattern is relaxed as a similarity with the occurrences of the event set. Similar patterns are extracted in [Moskovitch and Shahar, 2015] by constraining relatively the events. Those patterns include, however, not temporal information as they use the Allen's temporal logic [Allen, 1984].

Finally, the chronicle model, firstly introduced in [Dousson and Duong, 1999] seemed to be the more expressive and so the more adapted to extract interesting behaviors from temporal databases. This model will be detailed in the rest of this section and studied in this thesis.

This section presents so the chronicle model and the different chronicle mining approaches preceding this work. The chronicles are expressive temporal patterns that constrain numerically the temporal dimension and without being limited to sequential constraints. These patterns can be drawn as graphs.

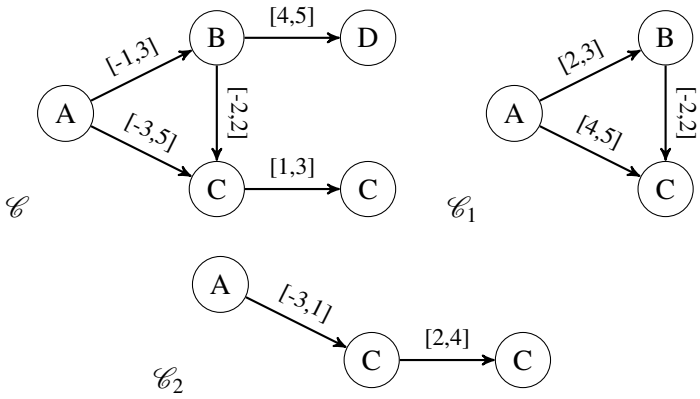


Figure 2.6: Example of three chronicles occurring in Table 2.1 (*cf.* Examples 1 and 4). No edge between two events is equivalent to the temporal constraint $]-\infty,\infty[$.

2.2.1 Chronicle definition

A **temporal constraint** is a tuple (e_1,e_2,t^-,t^+), also noted $e_1[t^-,t^+]e_2$, where $e_1,e_2 \in \mathbb{E}$, $e_1 \leq_{\mathbb{E}} e_2$ and $t^-,t^+ \in \mathbb{T} \cup \{-\infty,+\infty\}$, $t^- \leq t^+$. A temporal constraint $e_1[t^-,t^+]e_2$ is said satisfied by a couple of events $((e,t),(e',t'))$ iff $e = e_1$, $e' = e_2$ and $t'-t \in [t^-,t^+]$.

A **chronicle** is a couple $(\mathscr{E},\mathscr{T})$:

- The set $\mathscr{E} = \{\!\{e_1...e_n\}\!\}$, $e_i \in \mathbb{E}$ and $\forall i,j, 1 \leq i < j \leq n$, where $e_i \leq_{\mathbb{E}} e_j$ is a **multiset**, *i.e.* $\mathscr{E}$ can contain several occurrences of a same event type.
- The set $\mathscr{T}$ is a temporal constraint set: $\mathscr{T} = \{e[a,b]e' \mid e,e' \in \mathscr{E},\ e \leq_{\mathbb{E}} e'\}$. As the constraint $e[a,b]e'$ is equivalent to $e'[-b,-a]e$, we impose the order on items, $\leq_{\mathbb{E}}$, to decide which one is represented in the chronicle. Moreover, while $e = e'$, we can obtain two different chronicles which are equivalent: one based on $e[a,b]e'$ where $a \geq 0$ and one other based on $e'[-b,-a]e$. In such case we choose the positive one because it seems to be a more natural representation. Intuitively, two similar events can always be ordered: one is the earliest of their type in the sequence and the other is the latest. The positive temporal constraint represents this order.

Example 1. Figure 2.6 illustrates three chronicles represented by graphs. Chronicle $\mathscr{C} = (\mathscr{E},\mathscr{T})$ where $\mathscr{E} = \{\!\{e_1 = A, e_2 = B, e_3 = C, e_4 = C, e_5 = D\}\!\}$ and $\mathscr{T} = \{e_1[-1,3]e_2, e_1[-3,5]e_3, e_2[-2,2]e_3, e_2[4,5]e_5, e_3[1,3]e_4\}$ is illustrated at the top left. This graph is not complete. No edge between two events is equivalent to the temporal constraint $]-\infty,\infty[$, *i.e.* there is no constraint.

It can be noticed that chronicles are temporal constraint networks [Dechter et al., 1991]. Such temporal constraints are interdependent in such a way that we have to take care of, on the one side, their inconsistency and, on the other side, of their redundancy. Dechter et *al.* proposed an algorithm to reduce temporal constraints of a chronicle to an equivalent minimal chronicle, *i.e.* a chronicle with minimal interval lengths.

Example 2. Figure 2.7 illustrates reduced chronicles equivalent to $\mathscr{C}_1$ and $\mathscr{C}_2$ from Figure 2.6. We see that the difference between $\mathscr{C}_1$ and $\mathscr{C}_1'$ is the edge $B[-2,2]C$ for $\mathscr{C}_1$ and the edge $B[1,2]C$ for $\mathscr{C}_1'$. Indeed, as we have $A[4,5]C$ and $A[2,3]B$, it is not possible to observe C 2 time units before B. If B occurs 2 time units after A, C must occur 2 or 3 time units after B to satisfy $A[4,5]C$. If B occurs 3 time units after A, C must occur 1 or 2 time units after B to satisfy $A[4,5]C$. As 3 was not in the original temporal interval $[-2,2]$, we obtained with this reasoning the new temporal constraint $B[1,2]C$.

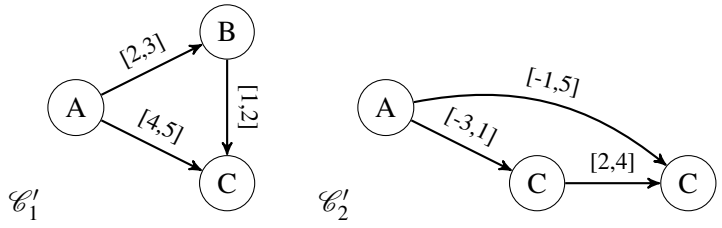


Figure 2.7: The chronicles $\mathscr{C}'_1$ and $\mathscr{C}'_2$ are respectively the reduced chronicles of $\mathscr{C}_1$ and $\mathscr{C}_2$ of Figure 2.6.

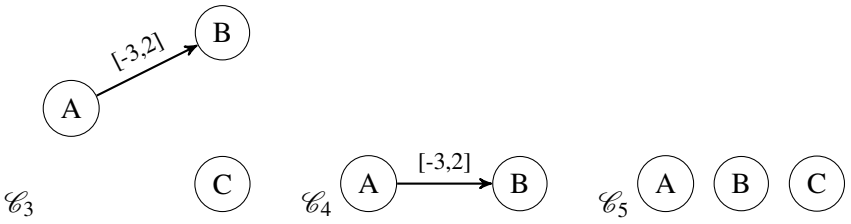


Figure 2.8: Example of a chronicle specialization/generalization. $\mathscr{C}_4$ and $\mathscr{C}_5$ are generalizations of $\mathscr{C}_3$. $\mathscr{C}_4$ has a more general multiset and $\mathscr{C}_5$ has more general temporal constraints.

The reasoning is the same for $\mathscr{C}_2$ except that the interval between A and the second C is initially $]-\infty,+\infty[$. By adding the lower bounds of the temporal constraints between A and the first C (-3) and between the two C (2) we obtain the lower bound between A and the second C (-1) and by adding the upper bounds (1 and 4) we obtain the upper bound (5).

Given two chronicles $\mathscr{C}_1 = (\mathscr{E}_1, \mathscr{T}_1)$ and $\mathscr{C}_2 = (\mathscr{E}_2, \mathscr{T}_2)$, we define the partial order $\preceq$ where $\mathscr{C}_1 \preceq \mathscr{C}_2$ if $\mathscr{E}_2 \subseteq \mathscr{E}_1$ and there is a strictly increasing function f where $\forall i, j,\ 1 \leq i < j \leq |\mathscr{E}_2|$, $1 \leq f(i) < f(j) \leq |\mathscr{E}_1|$, $e_i, e_j \in \mathscr{E}_2$, $e_{f(i)}, e_{f(j)} \in \mathscr{E}_1$, $e_{f(i)}[a,b]e_{f(j)} \in \mathscr{T}_1$, $e_i[a',b']e_j \in \mathscr{T}_2$, $e_{f(i)}[a,b]e_{f(j)} \subseteq e_i[a',b']e_j$. If $\mathscr{C}_1 \preceq \mathscr{C}_2$ and $\mathscr{C}_1 \neq \mathscr{C}_2$, we say that $\mathscr{C}_1$ is **more specific** than $\mathscr{C}_2$. On the contrary, $\mathscr{C}_2$ is **more general** than $\mathscr{C}_1$.

Example 3. Figure 2.8 illustrates a chronicle $\mathscr{C}_3 = (\mathscr{E}_3, \mathscr{T}_3)$ and two more general chronicles $\mathscr{C}_4 = (\mathscr{E}_4, \mathscr{T}_4)$ and $\mathscr{C}_5 = (\mathscr{E}_5, \mathscr{T}_5)$. We have $\mathscr{C}_3 \preceq \mathscr{C}_4$ because $\mathscr{E}_4 \subset \mathscr{E}_3$ and the only constraint in $\mathscr{T}_3$ is shared by $\mathscr{T}_4$. On the other hand we have $\mathscr{C}_3 \preceq \mathscr{C}_5$ because $\mathscr{E}_3 = \mathscr{E}_5$, but $\mathscr{T}_3$ has a more specific constraint than $\mathscr{T}_5$. In fact, $\mathscr{T}_5$ having no constraint, any additional constraint is more specific.

2.2.1.1 Chronicle support

Let $s = \langle (e_1,t_1), ..., (e_n,t_n) \rangle$ be a sequence and $\mathscr{C} = (\mathscr{E} = \{\!\{e'_1, ..., e'_m\}\!\}, \mathscr{T})$ be a chronicle. An **occurrence** of $\mathscr{C}$ in s is a subsequence of s, $\tilde{s} = \langle (e_{f(1)},t_{f(1)}),\ ...,\ (e_{f(m)},t_{f(m)}) \rangle$ such that 1) $f : [1,m] \mapsto [1,n]$ is an injective function, 2) $\forall i,\ e'_i = e_{f(i)}$ and 3) $\forall i, j,\ t_{f(j)} - t_{f(i)} \in [a,b]$ where $e'_i[a,b]e'_j \in \mathscr{T}$. It is worth noting that f is not necessarily increasing. In fact, there is a difference between (i) the order of the chronicle multiset defined on items, $\leq_{\mathbb{E}}$, and (ii) the order of events in sequences, $\leq_{\mathbb{T}}$, defined on the temporal domain. The chronicle $\mathscr{C}$ **occurs** in s, denoted $\mathscr{C} \in s$, iff there is at least one occurrence of $\mathscr{C}$ in s. The **support** of a chronicle $\mathscr{C}$ in a sequence set $\mathscr{S}$ is the number of sequences in which $\mathscr{C}$ occurs: $supp(\mathscr{C}, \mathscr{S}) = |\{s \in \mathscr{S} \mid \mathscr{C} \in s\}|$. Given a minimal support threshold σ_{min}, a chronicle is **frequent** iff $supp(\mathscr{C}, \mathscr{S}) \geq \sigma_{min}$.

Example 4. Chronicle $\mathscr{C}$ (see Figure 2.6 at the top left), occurs in sequences 1, 3 and 6 of Table 2.1. We notice there are two occurrences of $\mathscr{C}$ in sequence 1. Nonetheless, its support is $supp(\mathscr{C}, \mathscr{S}) = 3$. This chronicle is frequent in $\mathscr{S}$ for any minimal support threshold σ_{min} lower

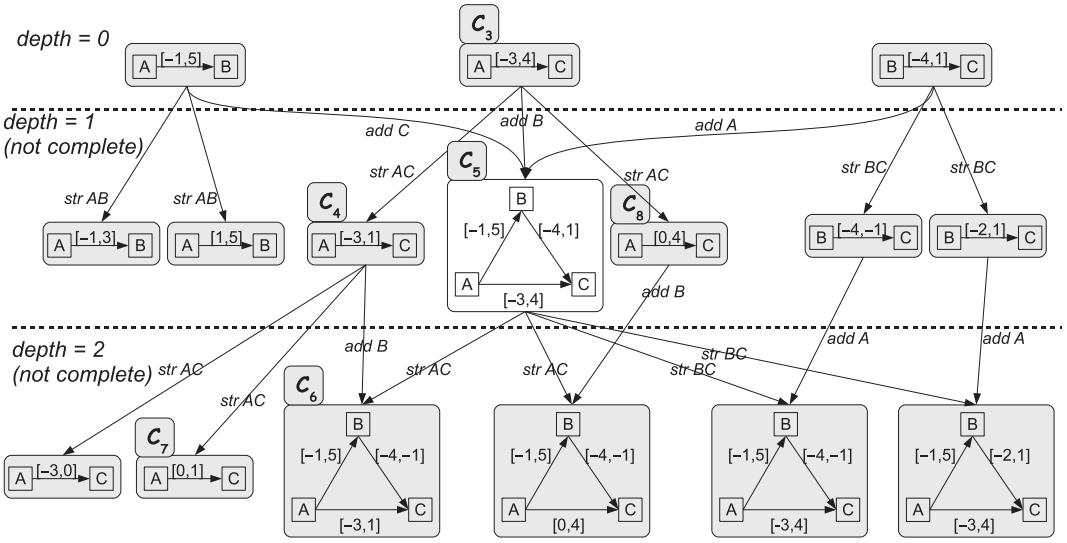


Figure 2.9: The chronicle pattern-growth step as illustrated in [Cram et al., 2012].

or equal to 3. The two other chronicles, denoted $\mathscr{C}_1$ and $\mathscr{C}_2$, occur respectively in sequences 1 and 3; and in sequence 6. Their supports are $supp(\mathscr{C}_1, \mathscr{S}) = 2$ and $supp(\mathscr{C}_2, \mathscr{S}) = 1$.

The support of chronicles is anti-monotone using the previously defined partial order $\preceq$. Indeed for two chronicles $\mathscr{C}_1$ and $\mathscr{C}_2$ and a sequence set $\mathscr{S}$, $\mathscr{C}_1 \preceq \mathscr{C}_2$ implies $supp(\mathscr{C}_1, \mathscr{S}) \leq supp(\mathscr{C}_2, \mathscr{S})$. This property derives from the fact that the most specific chronicle $\mathscr{C}_1$ cannot occur in more sequences than $\mathscr{C}_2$ because it is more constrained. This property has been formally proved and used in previous works on chronicles [Dousson and Duong, 1999, Cram et al., 2012, Huang et al., 2012, Alvarez et al., 2013].

2.2.2 Chronicle mining approaches

The paper introducing frequent chronicle mining [Dousson and Duong, 1999] works on alarm logs. The algorithm used to mine frequent chronicles is based on the generate and test framework (*i.e.* Apriori based). The algorithm firstly selects the frequent temporal constraint to be used for the generation. In the paper, it is called the discovering of chronicles of size 2. At most one temporal constraint is selected for a pair of events. Then, it combines these pairs of events to generate larger chronicles to evaluate. This algorithm is so not complete but it was an early version of chronicle mining algorithm.

HCDA [Cram et al., 2012] is a generalization of the Dousson's algorithm [Dousson and Duong, 1999] to extract the complete set of chronicles. The main difference between the two approaches is the selection of frequent temporal constraints. In HCDA, all possible frequent temporal constraints are used to enumerate the frequent chronicles. Figure 2.9 illustrates the enumeration process. It is based on a breadth-first search strategy. At first step the whole set of frequent temporal constraint is extracted. This set corresponds to the set of chronicles with two events. The candidate chronicles are then generated by adding new temporal constraints to a frequent chronicle before its frequency test. Figure 2.9 illustrates how the complete set of chronicle is extracted from the smallest chronicles.

Frequent chronicle mining is used in [Huang et al., 2012] to extract frequent behaviors in clinical pathways. The main difference between the two previous approaches is that the chronicles are extracted from a sequence set but not a trace. The issue implied by the difference between occurrence number and sequence number is tackled by the constraint that each event may occur at

most one time per sequence. Similarly to Cram et al., the complete set of frequent chronicles is extracted.

ASTPminer [Alvarez et al., 2013] is an apriori based algorithm that extracts the complete set of frequent chronicles from a sequence set. The datasets on which this algorithm is used are obtained from polysomnography tests in patients with sleep apnea–hypopnea syndromes. Contrary to the Huang et al. approach, a same event can occur several times in a sequence. This implies a big time complexity. To tackle this, a window size constraint of 80 seconds, *i.e.* the constraint that all events of a chronicle occur in a window of 80 seconds, is used to reduce the size of the extracted pattern set.

Several other works dedicated to the fault diagnosis domain are based on chronicles matching and learning [Quiniou et al., 2001, Morin and Debar, 2003, Cordier et al., 2007, Vásquez et al., 2017]. For example, HCDA is extended in HCDAM [Subias et al., 2014] to learn chronicles from multiple sequences. This work is more similar to the extraction of chronicles from a sequence set but still define the support as the total number of occurrences of a chronicle and not the number of sequences in which the chronicle occurs.

Efficient algorithms have already been proposed to extract frequent chronicles. [Cram et al., 2012] have defined the frequent chronicle lattice and proposed a complete algorithms to extract them. Finally, the chronicle model was used in medical contexts. The chronicle model is expressive and represents interesting behaviors for clinicians.

But the frequent chronicle mining task seems to be not sufficient to be used for pharmacoepidemiology. The set of frequent chronicles is large even with reasonable minimal support. Furthermore, the sequence label information is very important for our application and is not used by such approaches. We then have to fill this gap by proposing the discriminant chronicle mining.

Chapter 3

Supervised pattern mining

The extraction of all behaviors contained in a dataset is often too expensive in terms of time and memory. For example, in a dataset containing n different event types, 2^n itemsets can be extracted from it. The minimal support constraint help to extract a pattern subset but this set can be still huge and contains some redundancy. Unfortunately, it is so difficult for a user to look at the whole set of extracted patterns and so difficult to learn knowledge from it.

Algorithms were designed to extract a smaller pattern set than the frequent pattern set or at least more interesting for the user. This interestingness can be defined by the user with constraints.

In the following part, we are reviewing approaches that considers interesting the discriminant patterns *i.e.*, patterns that occur more in a subset of sequence than in the others. Those approaches are based on a label information on each sequence. This label information is represented on the dataset of Table 2.1 by an attribute apart from the sequence. The user has to define to which class correspond each sequence to extract the behaviors that are discriminant for each class. For example, in a pharmacoepidemiology context, the dataset could represent a specific population where each person will be treated for a disease. The class label could be the result of the treatment. The dataset of Table 3.1 illustrates such dataset.

3.1 Discriminant rule mining

The discriminant rule mining is the domain of discriminant pattern mining techniques specialized to extract discriminant behaviors from relational data. Relational data and sequential data are different because the length of each example is stated where the length of sequences can be variable.

SID	Healthcare pathway	Treatment result
1	$(Drug\ A,1)$, $(Drug\ B,3)$, $(Drug\ A,4)$, $(Drug\ C,5)$, $(Drug\ C,6)$	*Good*
2	$(Drug\ B,2)$, $(Drug\ D,4)$, $(Drug\ A,5)$	*Good*
3	$(Drug\ A,1)$, $(Drug\ B,4)$, $(Drug\ C,5)$, $(Drug\ B,6)$, $(Drug\ C,8)$	*Good*
4	$(Drug\ B,4)$, $(Drug\ A,6)$, $(Drug\ E,8)$	*Bad*
5	$(Drug\ B,1)$, $(Drug\ A,3)$, $(Drug\ C,4)$	*Bad*
6	$(Drug\ C,4)$, $(Drug\ B,5)$, $(Drug\ A,6)$, $(Drug\ C,7)$, $(Drug\ D,10)$	*Bad*

Table 3.1: Dataset example for the pharmaco-epidemiology domain. Each sequence represents the healthcare pathway of a patient. The label information is the treatment result for a disease. An application of discriminant pattern mining on such dataset could be to extract behaviors mostly occurring for good treatment result or on the contrary mostly occurring for bad treatment.

ID	Radius	Texture	Perimeter	Area	Smoothness	Compactness	Diagnostic
842302	17.99	10.38	122.8	1001	0.1184	0.2776	M
842517	20.57	17.77	132.9	1326	0.08474	0.07864	M
84300903	19.69	21.25	130	1203	0.1096	0.1599	M
84348301	11.42	20.38	77.58	386.1	0.1425	0.2839	M
84358402	20.29	14.34	135.1	1297	0.1003	0.1328	M
843786	12.45	15.7	82.57	477.1	0.1278	0.17	M
844359	18.25	19.98	119.6	1040	0.09463	0.109	M
84458202	13.71	20.83	90.2	577.9	0.1189	0.1645	M
844981	13	21.82	87.5	519.8	0.1273	0.1932	M
84501001	12.46	24.04	83.97	475.9	0.1186	0.2396	M
845636	16.02	23.24	102.7	797.8	0.08206	0.06669	M
84610002	15.78	17.89	103.6	781	0.0971	0.1292	M
8510426	13.54	14.36	87.46	566.3	0.09779	0.08129	B
8510653	13.08	15.71	85.63	520	0.1075	0.127	B
8510824	9.504	12.44	60.34	273.9	0.1024	0.06492	B

Table 3.2: A subset of the relational dataset: Wisconsin Diagnostic Breast Cancer (WDBC). Each line corresponds to a woman, identified by the ID attribute, concerned by breast cancer. The numerical attributes are measures obtained from a digitize image of a fine needle aspirate of a breast mass. The last attribute is the diagnostic of the cancer: benign (B) or malignant (M).

Relational data are so different from sequential data but algorithms dedicates to extract discriminant patterns from relational data can be used for extracting subsets of discriminant patterns or to learn a behavior from a pattern set as it is shown in the following parts. The dataset of Table 3.2 is an example of such type of dataset.

One of the standard approach to handle relational data is to learn decision trees. Standard algorithms dedicated to extract decision trees are *ID*3 [Quinlan, 1986], *C*4.5 [Quinlan, 1993] or *CART* [Breiman et al., 1984]. In this case, only one decision tree is learned to classify the whole dataset. An example of decision tree dedicated to the classification of Table 3.2 is given by Figure 3.1. In this example, the values are quantitative. The decision tree can so be built using the order defined on the numbers. But such tree can also build for qualitative values. For example, an attribute could be the *color* and its value domain $\{blue, red, green\}$. A node of the decision tree based on the color could then separate the examples for which $color = blue$ from the examples for which $color \neq blue$. Several implementations of decision trees do not make a difference between qualitative and quantitative values. It is the case of the implementation made in scikit-learn [Pedregosa et al., 2011] where all values are numbers. Quantitative values have to be mapped with integers to be used by such implementation. For example, $\{blue, red, green\}$ could be mapped with $\{0, 1, 2\}$. Thereby, $color \neq blue$ could be described by $color > 0$. Random decision forests [Ho, 1995, Ho, 1998] are designed to handle more complex data using several decision trees.

Another approach to handle such data is to extract ordered rules. When extracted from a dataset, an ordered rule set as to be read sequentially. Indeed, the second rule of the set covers only examples that are not covered by the first rule. An example of such rule set for dataset of Table 3.2 is:

1. $texture > 15.71 \rightarrow Diagnostic = M$
2. $smoothness \leq 0.1075$ *AND* $compactness \leq 0.127 \rightarrow Diagnostic = B$

In this example, the second rule is not true for the example with $ID = 844359$ but as this example covers was already covered by the first rule, the rule set perfectly classify the dataset. Such rule

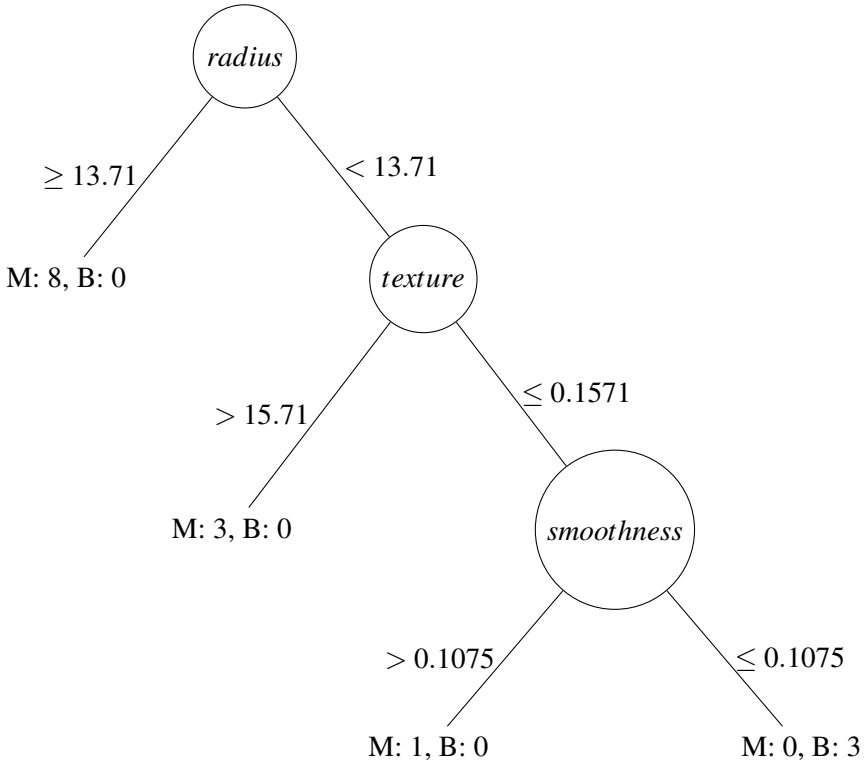


Figure 3.1: A decision tree classifying the examples of dataset in Table 3.2. Each node of the tree is an attribute used to split the dataset. The label on each edge define the examples concerned by each subtree. The number of each example concerned by the split is given in the leaves.

set can be seen as a compressed and simpler decision tree [Quinlan, 1987], however, rule learning systems initially outperformed decision tree learners [Pagallo and Haussler, 1990, Weiss and Indurkhya, 1991]. One of the best algorithms of the state of the art to extract such numerical rules is $Ripper_k$ [Cohen, 1995]. The $Ripper_k$ algorithm was designed to be competitive with $C4.5rules$, a rule learning algorithm derived from the decision tree learning algorithm $C4.5$ [Quinlan, 1993]. The two algorithms have similar results in terms of error rates but $Ripper_k$ is much faster the $C4.5rules$. The $Ripper_k$ algorithm is also designed to efficiently process noisy datasets.

The previous approaches handled mainly numerical data but some other are specific to symbolic data. It is the case for emerging patterns, subgroup discovery and contrast set. A review of those three domains and the unifying of the definitions of those domains is done in [Novak et al., 2009].

3.2 Discriminant temporal pattern mining

For large datasets, the number of temporal patterns may be huge and not equally interesting. Extracting less but more significant patterns becomes the goal in this case. The first research directions specified additional constraints on the expected sequential patterns [Pei et al., 2002] including constraints on patterns, on their occurrences or on the pattern set. More recent approaches try to use soft constraints, such as preferences, instead of hard constraints. For example, in [Lakshmanan et al., 2013], the temporal patterns are ranked according to their correlation with a particular patient outcome. It enables the expert to identify the potentially most interesting patterns easily. Nonetheless, their number is not reduced.

Mining discriminant sequential pattern can be seen as an additional hard constraint on the expected patterns. Several definitions related to the notion of discrimination have been proposed. Novak et *al.* [Novak et al., 2009] reviewed these approaches, including emerging patterns [Dong and Li, 1999], contrast patterns [Bay and Pazzani, 2001], subgroup discovery [Herrera et al., 2011] and more recently Exceptional Model Mining [Duivesteijn et al., 2016]. All these approaches have been explored for itemset mining and can be easily extended to sequential pattern mining. But extracting quantitative temporal constraints, such in chronicle patterns, becomes too complex when applying the same complete strategy, and target algorithms have to be proposed. In our framework, the discrimination constraint is based on the growth rate of pattern frequency, which is similar to the emerging pattern approach.

Compared to all studies on frequent temporal data mining, there are only few proposals to mine discriminant temporal patterns. One of the first approaches dedicated to extract discriminant temporal patterns used inductive logic to extract patterns with quantified inter-event duration [Quiniou et al., 2001]. Discriminant sequential patterns were also extracted in [Salle et al., 2009]. The extracted patterns using this approach are chronicles but an expert knowledge is needed to bound the search space.

3.3 Pattern-based classification

An alternative approach for learning how to label sequences is the pattern-based classification [Bringmann et al., 2011]. Pattern-based classification is a subdomain of sequence classification. A survey of this domain is done in [Xing et al., 2010]. The general purpose of sequence classification is to build efficient classifiers to label sequences. In pattern-based classification, patterns are extracted and then are used as rules or as features in a (standard) classifier, such as a SVM classifier. Figure 3.2 illustrates classification based on itemsets. Even if the original dataset and pattern type are different, the illustrated process is the same for temporal patterns. Before the classification

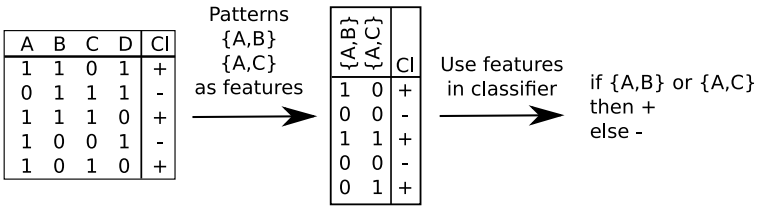


Figure 3.2: Illustration of the pattern-based classification process extracted from [Bringmann et al., 2011]. In this example, the patterns are itemsets but the process is the same for temporal patterns.

task, it can be useful to prune the extracted pattern set to remove those useless or keep those more useful for classification. In some cases, useful patterns (*e.g.* the frequent ones) can be directly selected and avoid an inefficient post-processing step. Batal et *al.* [Batal et al., 2013] proposed to use discriminant temporal pattern as potentially interesting subset of frequent patterns. Their temporal patterns, called minimal predictive temporal patterns, are couples of an itemset and Allen's temporal relations constraining the itemset occurrences [Allen, 1984]. The principle of minimal predictive temporal patterns selection is to compare a pattern $\mathscr{P}$ with those include in it to prune the search tree if $\mathscr{P}$ is enough predictive. The predictive measure is the confidence. In [Fradkin and Mörchen, 2015], timestamped event sequences are described by binary features encoding the presence/absence of a patterns. Feature vectors are then classified by a *SVM* classifier. Their most accurate approach extracts all frequent sequential patterns and use them all as features. The large number of extracted patterns and their redundancy is a problem for the efficiency and the accuracy of the *SVM* classifier. Fradkin et *al.* proposed several strategies to reduce the number of pattern to use in *SVM*. Different approaches (*BIDE-D*, *BIDE-DC*, *SMBT* and *SMBT-FS*) dedicated to discriminant pattern mining are developed and tested to reduce this number while keeping as high as possible the accuracy.

The main drawback of these approaches is their interpretability. The strength of pattern mining approaches is the interpretability of their outputs. This is an essential property we would like in our application field. But, extracted patterns are used as features for the classifier which is a black box for the expert. In such cases, the expert cannot relate a pattern to a behavior. Our approach consists in extracting patterns that hold the discrimination information such that they can be individually interpreted.

Discriminant temporal patterns are finally used to present a few number of patterns to the experts [Fabrègue et al., 2014]. In this paper, discriminant patterns are extracted from hydro-ecological data and the goal of the proposed approach is to support analysts to extract knowledge from those data. This approach does not extract discriminant patterns, but it selects those that satisfy a discrimination constraint among the frequent patterns. This post-processing approach misses potentially discriminant patterns that may be useful for experts.

For the great expressiveness of chronicles and their algorithmic properties, we are interested in frequent and discriminant chronicles mining. Our work continues previous work about chronicle mining and discriminant temporal pattern mining. As in [Fabrègue et al., 2014], our goal is to support clinicians to extract knowledge from data then our patterns must be interpretable rules but we could not rely on a black-box classifier to provide a good accuracy. Mining discriminant chronicles is a compromise between interpretability and accuracy.

A simple way to do pattern based classification is by using association rules [Ma and Liu, 1998]. The classification is then done like for $C4.5Rules$ or $Ripper_k$ but the associated items are, in this case, patterns.

In [De Smedt et al., 2017], sequential patterns are extracted in a declarative way to classify

SID	Sequence	Label
1	$(A,1)$, $(A,2)$, $(B,3)$, $(C,5)$, $(D,6)$	+
2	$(A,2)$, $(D,4)$, $(C,5)$, $(B,7)$	+
3	$(A,1)$, $(B,4)$, $(C,4)$	+
4	$(B,4)$, $(A,6)$, $(E,8)$, $(C,400)$	-
5	$(B,1)$, $(A,3)$, $(C,400)$	-
6	$(A,1)$, $(B,7)$, $(D,10)$, $(C,600)$	-

Table 3.3: Dataset example that highlights the importance of numerical temporal information in discriminant patterns. In this dataset, the sequential pattern $\langle A,B\rangle$ is not discriminant but a pattern constraining the duration between A and C to be lower than 400 is.

sequences. The patterns are directly extracted to classify the sequences. The results are hugely better than in [Fradkin and Mörchen, 2015].

Finally, an accurate method to do temporal pattern-based classification is proposed in [Bornemann et al., 2016]. In this work temporal patterns are itemsets for which each item that has starting and ending timestamps are constrained to match as most as possible a defined interval. The classification is then done using the similarity of the real occurrences of each itemset and the associated interval constraints.

3.4 Positioning

We choose to focus this thesis work on the discriminant chronicle mining task because of a lack of discriminant expressive pattern mining approaches.

The chronicle model was initially chosen to be the most expressive temporal pattern model while being sufficiently interpretable and graphically representable. Chronicle was preferred over the sequential patterns because of the numerical description of the time. Discriminant sequential pattern mining approaches were already studied but the lack of numerical information concerning the duration could lead to avoid interesting behaviors. Table 3.3 illustrates a dataset for which numerical temporal information is essential to describe the discriminant behavior based on A and C. Sequential patterns could only describe that A is always followed by C or C followed by A but those behaviors are not discriminant and not frequent respectively. Chronicles could easily describe this behavior by constraining C to occur at most, for example, 50 days after A.

On the other hand, we could not choose an expressive temporal pattern model without the discriminant dimension. Thereby, previous works on frequent chronicle mining were not usable in our context. Extracting sets of frequent chronicles from pharmaco-epidemiology data leads to a lot of uninteresting patterns. For example, the majority of the frequent chronicles extracted from the GENEPI dataset contained an event linked to a paracetamol delivery.

The discriminant chronicle model and the discriminant chronicle mining task were so needed.

Part II

Contributions

Chapter 4

Discriminant chronicle mining

Discriminant pattern mining is dedicated to mine patterns that are discriminant for some dataset in comparison to others. It aims at identifying patterns in structured and labeled examples that can be used to classify new examples. In this work, we aim at mining labeled temporal sequences, *i.e.* sequences of timestamped events. Our objective is not to predict future events but to assign a label to a sequence.

Sequence classification can be encountered in numerous application fields: comparing groups of customers (*e.g.* large family *vs* single child family), analyzing supermarket purchase data, identifying behavior of customers (who churns *vs* loyal customers), etc. Our application field is the analysis of care pathways, *i.e.* sequences of medical events of a patient (drugs deliveries, hospital stays, etc.): we would like to characterize care pathways relatively to some patient diseases. The objective is to identify therapeutic sequences that may favor diseases. In all these contexts, the temporal dimension may be discriminant. For instance, a short delay between the delivery of two adverse drugs may help discriminate patients, sick from healthy patients. This is especially important for clinicians. By taking quantitative temporal constraints into account, we aim at improving classification accuracy in order to provide better insights to data scientists.

Discovering such patterns raises important challenges both in the fields of pattern mining and machine learning. On the one hand, pattern mining approaches are based on a (exhaustive) search strategy in a structured version space. Our objective is to discover *quantitative* temporal information. This generates too large a search space for applying classical approaches. On the other hand, machine learning algorithms have difficulties to handle structured data such as sequences. As a consequence, we propose a specific pattern model and a new approach that combines solutions from these two research fields.

In this chapter, we explore temporal patterns called *chronicles* [Dousson and Duong, 1999]. A chronicle is a set of events linked by quantitative temporal constraints. In constraint satisfaction domain, chronicle can be seen as temporal constraint network [Dechter et al., 1991]. These complex but highly expressive patterns enable to take into account the quantitative temporal dimension of the data contrary to classical sequential patterns that capture only the sequentiality between events.

The remaining of the article is organized as follows. Next section presents related works. The section 4.1 introduces the discriminant chronicle mining task, while section 4.2 presents the DCM algorithm, our solution for this task. Section 4.4 presents how to take decision from a set of chronicles. Section 4.5 evaluates the DCM algorithm. After a first round of experiments on synthetic data, we compare the results of discriminant chronicles in classification context with those of discriminant sequential patterns [Fradkin and Mörchen, 2015] on *UCI* datasets.

SID	Sequence	Label
1	$(A,1)$, $(B,3)$, $(A,4)$, $(C,5)$, $(C,6)$, $(D,7)$	$+$
2	$(B,2)$, $(D,4)$, $(A,5)$, $(C,7)$	$+$
3	$(A,1)$, $(B,4)$, $(C,5)$, $(B,6)$, $(C,8)$,$(D,9)$	$+$
4	$(B,4)$, $(A,6)$, $(E,8)$	$-$
5	$(B,1)$, $(A,3)$, $(C,4)$	$-$
6	$(C,4)$, $(B,5)$, $(A,6)$, $(C,7)$, $(D,10)$	$-$

Table 4.1: Set of six sequences labeled with two classes $\{+,-\}$. This table is the same as the table 2.1 in the section 2.2.1.

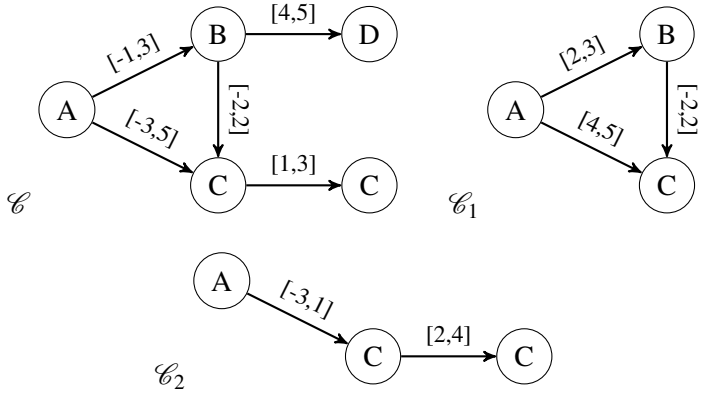


Figure 4.1: Examples of three chronicles occurring in Table 4.1 (*cf.* Examples 1 and 4). No edge between two events is equivalent to the temporal constraint $[-\infty,\infty]$. This figure is the same as the figure 2.6 in the section 2.2.1.

4.1 Discriminant chronicles

In this chapter we address the task of mining sequence sets with timestamped events. More specifically, we have a supervised mining task. Each sequence of the dataset is labeled. From another point of view, we can say that sequences are categorized in several classes. We are interested in discovering patterns that are more frequent in one class than in the others. To take time into account, we search for patterns called chronicles [Dousson and Duong, 1999]. This section introduces more formally the basic definitions related to the discriminant chronicle mining task.

Let $\mathscr{S}^+$ and $\mathscr{S}^-$ be two sets of sequences, and $\sigma_{min} \in \mathbb{N}^+$, $g_{min} \in]1,\infty[$ be two user-defined parameters. A chronicle $\mathscr{C}$ is **discriminant** for $\mathscr{S}^+$ iff $supp(\mathscr{C},\mathscr{S}^+) \geq \sigma_{min}$ and $supp(\mathscr{C},\mathscr{S}^+) \geq g_{min} \times supp(\mathscr{C},\mathscr{S}^-)$. In other words, a chronicle is discriminant iff its **growth rate** $g(\mathscr{C},\mathscr{S})$ is above the given threshold g_{min}. To take into account the special case of a null support in negative sequence set, the growth rate is defined as follows:

$$g(\mathscr{C},\mathscr{S}) = \begin{cases} \frac{supp(\mathscr{C},\mathscr{S}^+)}{supp(\mathscr{C},\mathscr{S}^-)} & \text{if } supp(\mathscr{C},\mathscr{S}^-) > 0 \\ +\infty & \text{otherwise} \end{cases}.$$

Contrary to the minimal support constraint which is anti-monotone, the minimal growth constraint is not. This is due to the independence of $supp(\mathscr{C},\mathscr{S}^+)$ and $supp(\mathscr{C},\mathscr{S}^-)$. For example, a chronicle $\mathscr{C}$ is not discriminant for $g_{min} = 2$ with $supp(\mathscr{C},\mathscr{S}^+) = 10$ and $supp(\mathscr{C},\mathscr{S}^-) = 10$ but $\mathscr{C}_1$ which is more specific than $\mathscr{C}$ is discriminant with $supp(\mathscr{C},\mathscr{S}^+) = 10$ and $supp(\mathscr{C},\mathscr{S}^-) = 5$ and $\mathscr{C}_2$ which is more specific than $\mathscr{C}_1$ is not discriminant with $supp(\mathscr{C},\mathscr{S}^+) = 2$ and $supp(\mathscr{C},\mathscr{S}^-) =$

Algorithme 2 Algorithm DCM for discriminant chronicle mining

Require: $\mathscr{S}^+$, $\mathscr{S}^-$: sequence sets, σ_{min} : minimal support threshold, g_{min} : minimal growth threshold
1: $\mathbb{M} \leftarrow$ EXTRACTMULTISETS($\mathscr{S}^+$, σ_{min}) ▷ $\mathbb{M}$ is the frequent multiset set
2: $\mathbb{C} \leftarrow \emptyset$ ▷ $\mathbb{C}$ is the discriminant chronicle set
3: **for all** $ms \in \mathbb{M}$ **do**
4: **if** $supp\left(\mathscr{S}^+, (ms, \mathscr{T}_\infty)\right) \geq g_{min} \times supp\left(\mathscr{S}^-, (ms, \mathscr{T}_\infty)\right)$ **then**
5: $\mathbb{C} \leftarrow \mathbb{C} \cup \{(ms, \mathscr{T}_\infty)\}$ ▷ Discriminant chronicle without temporal constraints
6: **else**
7: **for all** $\mathscr{T} \in$ EXTRACTDISCRCONSTRAINTS($\mathscr{S}^+$, $\mathscr{S}^-$, ms, g_{min}, σ_{min}) **do**
8: $\mathbb{C} \leftarrow \mathbb{C} \cup \{(ms, \mathscr{T})\}$ ▷ Add a new discriminant chronicle
9: **return** $\mathbb{C}$

5 and so forth. This property makes the mining of discriminant chronicles more difficult than standard chronicle mining because it is not possible to conclude that all the more specific chronicles of a not discriminant chronicle are not discriminant either.

Example 5. With chronicle $\mathscr{C}$ of Figure 4.1, $supp(\mathscr{C}, \mathscr{S}^+) = 2$, $supp(\mathscr{C}, \mathscr{S}^-) = 1$, where $\mathscr{S}^+$ (resp. $\mathscr{S}^-$) is the sequence set of Table 4.1 labeled with $+$ (resp. $-$). Considering that $g(\mathscr{C}, \mathscr{S}) = 2$, $\mathscr{C}$ is discriminant if $g_{min} \leq 2$. For chronicles $\mathscr{C}_1$ and $\mathscr{C}_2$, $supp(\mathscr{C}_1, \mathscr{S}^+) = 2$ and $supp(\mathscr{C}_1, \mathscr{S}^-) = 0$ so $g(\mathscr{C}_1, \mathscr{S}) = +\infty$ and $supp(\mathscr{C}_2, \mathscr{S}^+) = 0$ and $supp(\mathscr{C}_2, \mathscr{S}^-) = 1$ so $g(\mathscr{C}_2, \mathscr{S}) = 0$. $\mathscr{C}_2$ is not discriminant, but $\mathscr{C}_1$ is for any g_{min} value.

The support constraint, using σ_{min}, prunes the infrequent, and so insignificant, chronicles. For example, a chronicle like $\mathscr{C}_1$ such that $g(\mathscr{C}_1, \mathscr{S}) = +\infty$ but $supp(\mathscr{C}_1, \mathscr{S}^+) = 2$ is discriminant but would not be interesting for a bigger sequence set because it generalizes only two sequences. Pruning can be done efficiently thanks to the anti-monotonicity of support. More specifically, if a chronicle[1] $(\mathscr{E}, \mathscr{T}_\infty)$ is not frequent, then no chronicle of the form $(\mathscr{E}, \mathscr{T})$ will be frequent. This means that temporal constraints may be extracted only for frequent multisets.

Extracting the complete set of discriminant chronicles is not interesting because it may contain discriminant chronicles with same multiset and similar temporal constraints. Such chronicles could match an almost similar set of sequences and could be considered as redundant for the analyst. It is preferable to extract chronicles whose temporal constraints are the most generalized. The approach proposed in the next section efficiently extracts a set of meaningful discriminant chronicles above given support and growth rate thresholds.

4.2 DCM algorithm

The DCM algorithm is detailed in Algorithm 2. The algorithm extracts discriminant chronicles in two steps: First, it extracts frequent multisets, which are chronicles without temporal constraints, and then it mines discriminant temporal constraints from these multisets.

At first, line 1 (EXTRACTMULTISETS) extracts $\mathbb{M}$, the frequent multiset set in $\mathscr{S}^+$. This task is described in Section 4.2.1. In a second step, lines 3 to 8 extract the discriminant temporal constraints of each multiset. The naive approach would be to extract discriminant temporal constraints for all frequent multisets. A multiset $\mathscr{E}$ (*i.e.* a chronicle $(\mathscr{E}, \mathscr{T}_\infty)$), which is discriminant, may yield numerous similar discriminant chronicles with most specific temporal constraints. We consider them as useless and, as a consequence, line 4 tests whether the multiset *ms* is discriminant. If so, $(ms, \mathscr{T}_\infty)$ is added to the discriminant patterns set without more specification of the temporal constraints. Otherwise, lines 7-8 generate chronicles from discriminant temporal constraints identified by the routine EXTRACTDISCRCONSTRAINTS. This routine is detailed in Section 4.2.2.

[1] $\mathscr{T}_\infty$ is the set of temporal constraints $\{e] - \infty, +\infty[e' \mid e, e' \in \mathscr{E}\}$.

4.2.1 Multiset mining

This section elaborates on the frequent multiset mining. Compared to frequent itemset mining, it takes care of the cardinality of items *i.e.* its number of repetitions.

This task can be easily solved by applying a regular frequent itemset mining algorithm on a transaction set encoding multiple occurrences of a same item as several items. Such dataset used for multiset mining contains a transaction for each positive sequence in the initial temporal sequence set. For each transaction, an item $a \in \mathbb{E}$ occurring n times in a sequence is encoded by n items: $I_1^a, \ldots, I_n^a$.

SID	Items					
1	I_1^A	I_2^A	I_1^B	I_1^C	I_2^C	I_1^D
2		I_1^A	I_1^B	I_1^C	I_1^D	
3	I_1^A	I_1^B	I_2^B	I_1^C	I_2^C	I_1^D

Table 4.2: The transaction set obtained from the positive sequences of the dataset of the table 4.1 to extract multisets.

Example 6. Table 4.2 shows the transaction set obtained for the set of temporal sequences of Table 2.1 (page 29). The transactions correspond only to the positive sequences. Each transaction contains the same number of items as the number of events in the corresponding sequence. In first transaction, the two occurrences of the events of type A became two distinct items I_1^A and I_2^A.

A frequent itemset of size m, $(I_{i_k}^{e_k})_{1 \leq k \leq m}$, extracted from this dataset is transformed into the multiset containing, i_k occurrences of the event e_k. Frequent itemsets with two items $I_{i_k}^{e_k}$, $I_{i_l}^{e_l}$ such that $e_k = e_l$ and $i_k \neq i_l$ are redundant and, thus, these itemsets are ignored to avoid generating several times the same multiset. This could be tackled by adding pattern constraints into the itemset mining algorithm. Practically, we implemented a post-processing step.

Example 7. For $\sigma_{min} = 2$, the itemset $\{I_1^A, I_1^B, I_1^C\}$ is frequent in the dataset of the table 4.2 since it occurs in each transaction. Then, the extracted itemset is translated as the multiset $\{\!\{A, B, C\}\!\}$ because I_1^A, I_1^B and I_1^C correspond to one occurrence of A, one occurrence of B and one occurrence of C respectively. Another frequent itemset is $\{I_1^A, I_1^B, I_2^C\}$. This itemset is frequent for $\sigma_{min} = 2$ because it occurs in the two transactions 1 and 3. It can be translated as the multiset $\{\!\{A, B, C, C\}\!\}$ because I_2^C corresponds to two occurrences of C. To avoid redundancy, itemsets containing two items referring to the same event will be ignored. An example of such itemsets is $\{I_1^A, I_1^B, I_1^C, I_2^C\}$. This itemset is ignored because it has the same support as $\{I_1^A, I_1^B, I_2^C\}$ and contains both I_1^C and I_2^C that are equivalent to I_2^C alone.

4.2.2 Temporal constraint mining

The general idea of EXTRACTDISCCONSTRAINTS is to extract discriminant temporal constraints using a classical numerical rule learning task [Cohen, 1995].

Let $\mathscr{E} = \{\!\{e_1..e_n\}\!\}$ be a frequent multiset. A relational[2] dataset, denoted $\mathscr{D}$, is generated with all occurrences of $\mathscr{E}$ in $\mathscr{S}$. As $\mathscr{C} = (\mathscr{E}, \mathscr{T}) \preceq (\mathscr{E}, \mathscr{T}_\infty)$, $\mathscr{D}$ is sufficient to describe the entire occurrences set of $\mathscr{C}$. The numerical attributes of $\mathscr{D}$ are inter-event duration between each pair

[2] In some context, relational dataset designates a dataset whose organization is based on a relational data model. In our context, the relational model designed an attribute-value dataset, *i.e.* a single relation.

SID	$\mathscr{A}_{A\to B}$	$\mathscr{A}_{B\to C}$	$\mathscr{A}_{A\to C}$	Label
1	2	2	4	+
1	2	3	5	+
1	−1	2	1	+
1	−1	3	2	+
2	−3	5	2	+
3	3	1	4	+
3	3	4	7	+
3	5	2	7	+
3	5	−1	4	+
5	−2	3	1	−
6	−1	−1	−2	−
6	−1	2	1	−

Table 4.3: Tabular dataset for the multiset $\{\!\{A,B,C\}\!\}$ corresponding to its occurrences in table 4.1.

(e_i,e_j) where $e_i,e_j \in \mathscr{E}$ and $e_i \leq_{\mathbb{E}} e_j$, denoted by $\mathscr{A}_{e_i\to e_j}$. An example is labeled by its sequence label ($L \in \mathbb{L}$). If a sequence has several occurrences of $\mathscr{E}$, then each occurrence yields one example.

A rule learning algorithm induces numerical rules from $\mathscr{D}$. A rule has for conclusion a label and its premise is a conjunction of conditions on attribute values. Conditions are inequalities in the form: $\mathscr{A}_{e_i\to e_j} \geq x \wedge \mathscr{A}_{e_i\to e_j} \leq y$, where $(x,y) \in \overline{\mathbb{R}}^2$. Such a rule is translated as a temporal constraint set, $\mathscr{T} = \{e_i[x,y]e_j\}$. Infinite temporal constraint bounds are used to represent the lack of conditions. For example, the condition $\mathscr{A}_{e_i\to e_j} \geq x$ is translated as $e_i[x,\infty[e_j$. The couple $(\mathscr{E},\mathscr{T})$ is then a potential discriminant chronicle. At this stage, we are not sure that the chronicle is discriminant for the sequences. In fact, the rule learning algorithm extracts discriminant temporal constraints based on the dataset $\mathscr{D}$. The multiple instances of the multiset have to be carefully managed. We will discuss about this limitation in Section 4.3.

Example 8. Table 4.3 is the relational dataset obtained from the occurrences of $\{\!\{A,B,C\}\!\}$ containing by the dataset of Table 4.1. The attribute $\mathscr{A}_{A\to B}$ denotes the duration between A and B. We can see on Table 4.3 that an example contains the *SID* of the sequence, the duration for each pair of events and the label of the sequence. It is worth noticing that several examples may come from the same sequence. The rule $\mathscr{A}_{A\to C} \geq 2 \implies +$ characterizes almost all the examples labeled by $+$ in Table 4.3 and characterizes all the different *SID* labeled by $+$. It is translated into the discriminant temporal constraints $\{A[2,\infty[C\}$ which gives the discriminant chronicle $\mathscr{C} = (\{\!\{e_1 = A, e_2 = B, e_3 = C\}\!\}, \{e_1[2,\infty[e_3\})$.

It is interesting to notice that the discriminant temporal constraints $\{A[-1,\infty[B,\ B[3,\infty[C\}$ is less discriminant from the point of view of the dataset of Table 4.3, but it also discriminates perfectly the sequences, as it occurs in all positive sequences (at least for one multiset instance) and no negative sequence.

4.2.2.1 Rule learning for discriminant temporal constraint extraction

We want to obtain rules from numerical data to translate them into temporal constraint sets. Those rules have to be unordered to be translated independently from the others.

The rule learning task is done in practice by the $Ripper_k$ algorithm [Cohen, 1995], a relational rule learning algorithm. This algorithm was selected because of its high accuracy and because it allows the extraction of unordered rules. An unordered rule is independent from the others. Thereby, each rule of an unordered rule set is still true if read alone. It is not true for ordered rules. The problem with ordered rules learners (*e.g.* CN2, C4.5) comes with the validity of a rule

SID	$\mathscr{A}_{A\to B}$	$\mathscr{A}_{B\to C}$	$\mathscr{A}_{A\to C}$	Label
1	2	0	2	+
1	2	2	4	+
1	2	1	3	+
2	0	1	1	+
3	0	2	2	−
4	2	1	3	−

Table 4.4: Relational dataset for the multiset $\{\!\{A,B,C\}\!\}$ (different of 4.3 and no correlated with Table 4.1) to show the limitation of $Ripper_k$ in multiple instance cases.

at position n. This rule is valid only if the previous $n-1$ rules are not valid. Thereby if a chronicle set is based on ordered rules, some chronicles may not be discriminant if read alone.

$Ripper_k$ splits the dataset in two parts: *Grow* and *Prune*. *Grow* is used to construct the conjunction of conditions discriminating examples of one class with respect to the others. Growing stops when it is no longer possible to add a condition to the conjunction that improves accuracy. *Prune* is used to prune the constructed rule. If the conjunction is more discriminating without its last condition, this rule is removed from the conjunction and the last new condition is tested until the accuracy can no longer be improved. This approach allows to manage the overfitting made on *Grow*. If the accuracy is not satisfied for this conjunction of conditions, the search stops for that label. Otherwise the conjunction is returned as a rule, the examples associated with it are removed from *Grow* and the search starts again. The aforementioned steps are repeated for each label of the dataset.

However, the rules generated by $Ripper_k$ may not be discriminant or not frequent. Such case occurs because the support of rules generated by $Ripper_k$ is based on the number of examples and not the number of sequences. Since several examples can be generated by a single sequence, those two numbers can be different. Thereby, the definition of support for a rule extracted by $Ripper_k$ is different from our definition of support. To restore the correctness *i.e.* to only return discriminant patterns, the support and growth rate of each rule are recomputed and the rules detected as not frequent or not discriminant are pruned. This solution does not ensure that discriminant chronicles are extracted, even if one exists, *i.e.* all extracted chronicles could be pruned. However, such solution does not add computational heaviness and is sufficient to guarantee the correctness of the approach. The main idea behind the possibility that $Ripper_k$ can extract no discriminant rules, even if one exists, is the selection of the best rule covering a set of examples made by $Ripper_k$.

Example 9. Since each perfect discriminant rules that could be extract from Table 4.4 covers only one example and then will be overfitting, the first candidate rule for an algorithm like $Ripper_k$ should be $\mathscr{R}_1 : \mathscr{A}_{A\to B} \leq 2 \wedge \mathscr{A}_{A\to B} \geq 2 \implies +$. This rule will be validate on the *Prune* set and the generation will stop because there will be no more enough positive examples in the *Grow* set.

$\mathscr{R}_1$ will be finally pruned because its growth rate is equal to 1 and the extraction will not keep any chronicle despite the rule $\mathscr{R}_2 : \mathscr{A}_{B\to C} \leq 1 \wedge \mathscr{A}_{B\to C} \geq 1 \implies +$ that could be discriminant for a minimal growth rate of 2.

The choice of using a rule learning algorithm based on an incomplete heuristic, in this case a heuristic based on the MDL principle (*Minimum Description Length*), is required for computational reasons. However, $Ripper_k$ combines (1) a reasonable algorithmic complexity – the computation times remain reasonable, (2) good classification performance – the extracted chronicles are therefore well representative of the dataset – and (3) reduced rule sets – retrieved chains remain easily interpretable [Lattner et al., 2003].

4.3 Dealing with the multiple instance issues

The problem of learning rules on objects that could produce several examples in the learning sets is a known problem and is studied by the *multiple instance learning* domain. Thereby, solutions have been developed to tackle our issue where $Ripper_k$ generates not frequent or not discriminant rules.

4.3.1 Multiple instance problem

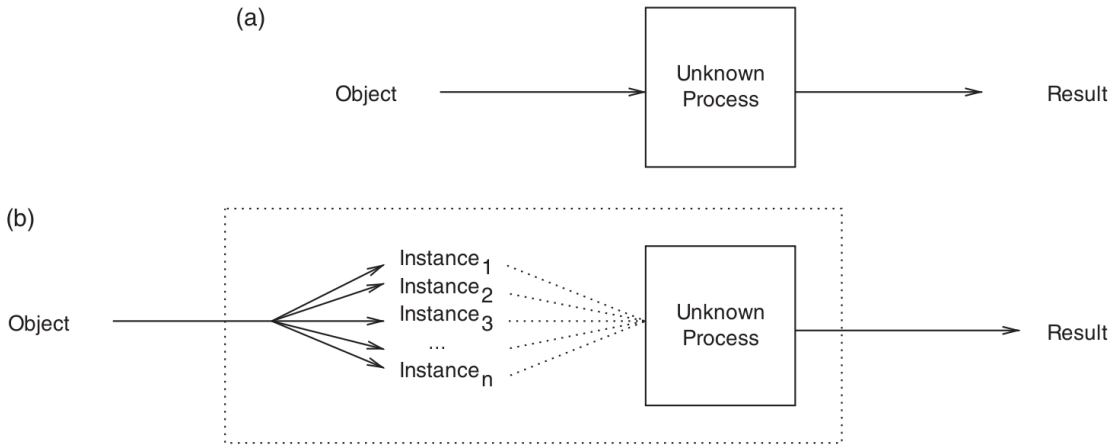


Figure 4.2: (a) The traditional supervised machine-learning scenario. (b) Multi instance learning. Figure from [Foulds and Frank, 2010] based on a similar diagram in [Dietterich et al., 1997].

The multiple instance learning [Foulds and Frank, 2010] refers to the counting problems which happens when a pattern appears multiple times in an object. Figure 4.2 shows the general differences between (a) the traditional supervised machine-learning scenario and (b) the multiple instance learning. Both scenarios have for goal to apply processes on some objects to produce a result. In the more specific context of rule learning on tabular data, the goal is to learn rules describing objects. The difference between the two scenarios is that, in traditional supervised machine-learning scenario, we can represent an object as a single feature vector but, in multiple instance learning, an object is described by a set of instances.

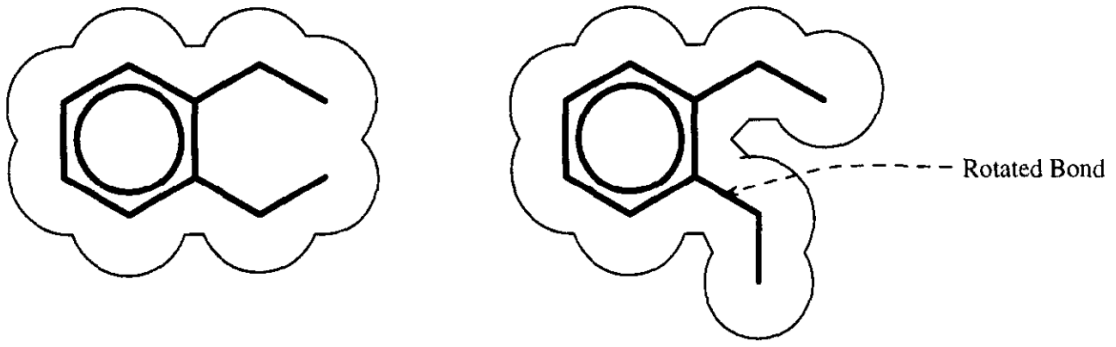


Figure 4.3: The original figure from [Dietterich et al., 1997] showing two possible shapes of a same molecule.

Initially, multiple instance learning was used for drug activity prediction [Dietterich et al.,

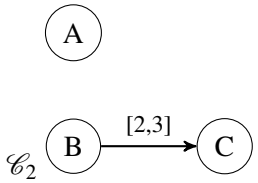


Figure 4.4: Example of not discriminant chronicle that could be extracted by $Ripper_k$. This chronicle is not discriminant because it occurs both in one positive and one negative sequences in the Table 4.1 but could be extracted because it covers two different occurrences in positive sequences in the Table 4.3.

SID	$\mathscr{A}_{A\to B}$	$\mathscr{A}_{B\to C}$	$\mathscr{A}_{A\to C}$	Label
1	2	1	3	+
2	0	1	1	+
3	0	2	2	−
4	2	1	3	−

Table 4.5: A mono-instance dataset for the multiset $\{\!\{A,B,C\}\!\}$ containing the witnesses of the table 4.4.

1997]. In drug activity prediction, the goal is to construct a model predicting which kind of drug will bind to a target molecule. Since drug is generally itself a small molecule, the multiple instance scenario occurs when a drug can have several shapes. Figure 4.3 is an example of such molecule for which several shapes are possible. In such context, the shape of the molecule can have an impact on the drug activity. The rule learning dataset becomes so a set of examples corresponding to all the shapes of the studied drugs and their known activity on the target molecule. Since the support of a rule must be the number of drugs covered by this rule and not the number of drug shapes, it becomes necessary to use multiple instance learning algorithms on those data.

In our case, an object is a sequence and the multiple instances are the multiple occurrences of a multiset in this sequence. This situation is encountered when building the dataset (*cf.* Table 4.3) and distorts the count of objects validating a standard conjunction of rules. For example, when rows 1 and 2 in the Table 4.3 are covered by a rule, they must only count for an object (*i.e.* a *SID*). The rule learning algorithm used to extract the discriminant temporal constraints, in our case $Ripper_k$, can then extract rules that are not discriminant in a multiple instance context.

Example 10. Figure 4.4 is an example of a chronicle that could be generated by the rule $\mathscr{A}_{B\to C} \leq 3 \wedge \mathscr{A}_{B\to C} \geq 2 \implies +$. This rule is discriminant for a standard rule learning algorithm as it covers more positive than negative examples in the table 4.3 but, since the first two examples correspond to the same sequence, this rule is not discriminant according to our definition of support.

4.3.2 Witness selection for $Ripper_k$

An alternative solution to the post-processing of $Ripper_k$ results would be to use methods adapting multiple instance datasets for standard machine learning algorithms. For example, the *miSVM* approach [Doran and Ray, 2014] identify for each object which instance has to be used by the classifier. This instance is seen as a witness of the object class. One possible approach for our algorithm would be to pre-process the dataset using the *miSVM* algorithm in order to identify the witness instances. In such a case, $Ripper_k$ would be applied on a mono-instance dataset.

Example 11. The table 4.5 is an example of witnesses dataset that the *miSVM* algorithm could compute from Table 4.4. Since there is no more duplicate occurrences, $Ripper_k$ will extract the

rule $\mathscr{R}_2 : \mathscr{A}_{B\to C} \leq 1 \wedge \mathscr{A}_{B\to C} \geq 1 \implies +$ that is really discriminant and no more $\mathscr{R}_1 : \mathscr{A}_{A\to B} \leq 2 \wedge \mathscr{A}_{A\to B} \geq 2 \implies +$.

This solution seems to be the most reliable. However, it adds an important complexity to the process of extracting temporal constraints. Furthermore, too many patterns could be removed by such approach. This high removal of patterns can be explained by the choice of only one witness per sequence. The hypothesis made for witness selection, *i.e.* the hypothesis that there is only one example describing the specificity of an object, is a very strong hypothesis. In our case, the hypothesis made is that discriminant chronicles can be extracted considering only one occurrence of each multiset in a sequence. This hypothesis overrules the potential co-occurrences of discriminant patterns.

4.3.3 A multiple instance rule learning algorithm: MITI

Several rule learning algorithms for multiple instance rule learning have been proposed in the literature [Chevaleyre and Zucker, 2001, Dietterich et al., 1997, Gärtner et al., 2002]. Replace the use of $Ripper_k$ by one of those algorithms could be a good solution for this problem. Since none of them extract unordered rules, we can use a multiple instance decision tree learning algorithm like MITI [Blockeel et al., 2005]. Using such algorithm, a discriminant temporal constraint will be a branch of the decision tree leading to a discriminant and frequent enough conclusion.

Taking into account multiple instances directly in the rule learning step would therefore necessarily be done with adding computational complexity. Since multiple instances are rarely encountered in our applications, we have focused on computational efficiency by correcting the results with the post-processing strategy. Furthermore, the $Ripper_k$ accuracy is in general cases equivalent or better than this of multiple instance algorithms.

4.4 Decision based on discriminant pattern sets

The decision based on the extracted discriminant chronicles is not the main idea of this chapter, however, we need to describe the general principles since we use it as evaluation. A discriminant chronicle extracted for a class can be seen as a rule for which an occurrence of this chronicle implies the associated label. We can extract rules of the form $\mu(\mathscr{C}, s) \implies l_i$ for each class where $\mathscr{C}$ is a chronicle, s a sequence, μ the mapping function determining if $\mathscr{C}$ occurs in s and l_i the label of the i^{th} class. The discriminant chronicle mining task extracts patterns that also occur in the classes different from that initially discriminated. Moreover, two extracted discriminant chronicles could have been extracted for different growth rates and supports. Therefore it is possible to have to predict the label of a sequence s, the two rules $\mu(\mathscr{C}, s) \implies l_i$ and $\mu(\mathscr{C}', s) \implies l_j$ where $l_i \neq l_j$ and $\mu(\mathscr{C}, s) = \mu(\mathscr{C}', s) = 1$. In such case we have to apply an order on the extracted rule set to make a decision.

To order the chronicles, we choose to use a measure based on the same growth rate definition that we use to extract them. Discriminant chronicles can have different support and in the case of two discriminant chronicles $\mathscr{C}_\infty$ and $\mathscr{C}_\in$ extracted for l_1 and l_2 respectively, $supp(\mathscr{C}_\infty, l_1) = 10$, $supp(\mathscr{C}_\infty, \overline{l_1}) = 0$, $supp(\mathscr{C}_\in, l_2) = 100$ and $supp(\mathscr{C}_\in, \overline{l_2}) = 1$, we could prefer to use $\mathscr{C}_\in$ before $\mathscr{C}_\infty$ even if $\mathscr{C}_\infty$ has a better growth rate. The reason for this preference is due to a wider covering of $\mathscr{C}_\in$ and, thereby, a lower sensibility to the false positive rate. We choose to recompute the growth rate of each chronicle with adding a virtual sequence to the negative support. This measure is currently used in the subgroup discovery domain and is presented in [Herrera et al., 2011] as the precision measure $Q_g(R) = \dfrac{TP}{FP+g}$ where, in our case, $TP = supp(\mathscr{C}, l_i)$, $FP = supp(\mathscr{C}, \overline{l_i})$ and g is set up to 1.

4.5 Benchmark

This section is dedicated to evaluate the efficiency of our algorithm on synthetic and real data. The real datasets have been used for other pattern-based classification approaches [Fradkin and Mörchen, 2015, Bornemann et al., 2016].The DCM implementation is written in C++ and relies on pre-existing implementations of *LCM* [Uno et al., 2004] and $Ripper_k$ [Cohen, 1995].

4.5.1 Synthetic data

In this section, we show the results of the evaluation of DCM on synthetic data.

4.5.1.1 Dataset generation

The general principle of the simulator is to generate sequences based on two different chronicles $\mathscr{C}^+$ and $\mathscr{C}^-$. In a first step, one of these chronicles is assigned to every sequence of the two datasets $\mathscr{S}^+$ and $\mathscr{S}^-$ such that $\mathscr{C}^+$ (respectively $\mathscr{C}^-$) is introduced mainly in the dataset $\mathscr{S}^+$ (respectively $\mathscr{S}^-$) (see Figure 4.5). In a second step, the chronicle assigned to the sequence is used as a "template": the sequence holds the items of the chronicle with timestamps randomly drawn but satisfying the temporal constraints, and additional random items. For each sequence, the chronicle to used to define its "template" and then random items are generated to introduce noise.



Figure 4.5: Distribution of chronicles $\mathscr{C}^+$ and $\mathscr{C}^-$ in sequence datasets $\mathscr{S}^+$ and $\mathscr{S}^-$.

The dataset $\mathscr{S} = \mathscr{S}^+ \cup \mathscr{S}^-$ contains labeled sequences ($\mathbb{L} = \{+,-\}$). Sequences labeled with $+$ (respectively $-$) are then characterized by $\mathscr{C}^+$ (respectively $\mathscr{C}^-$). Two parameters manage the introduction of these chronicles: $\widehat{f}_{min}$ and $\widehat{g}_{min}$.

- The parameter $\widehat{f}_{min}$ is the percentage of sequences in the majority class in which a chronicle appears.
- The parameter $\widehat{g}_{min}$ is the ratio between the sequence numbers of the two classes in which this chronicle appears.

The objective of the experiment is to evaluate the ability of DCM to extract correct discriminant temporal constraints. To this end, we use chronicles for $\mathscr{C}^+$ and $\mathscr{C}^-$ that differ only on one temporal constraints. The more similar the chronicles, the harder the mining task.

More precisely, the $Base_N$ datasets have been created from chronicles based on the same multiset $\{\!\{A,B\}\!\}$ of size 2. Therefore each chronicle has a unique time interval. For each dataset, the intervals of $\mathscr{C}^-$ and $\mathscr{C}^+$ overlap more or less. For all these datasets, we set $\mathscr{C}^+ = (\{\!\{A,B\}\!\}, \{A[3,10]B\})$; we set $\mathscr{C}^- = (\{\!\{A,B\}\!\}, \{A[7,8]B\})$ to generate $Base_1$. $Base_2$, $Base_3$, $Base_4$ and $Base_5$ are generated using similar constraints for chronicles $\mathscr{C}^-$: respectively, $A[6,8]B$, $A[4,8]B$, $A[4,9]B$ and $A[4,10]B$. These chronicles are illustrated in the Table 4.6.

It is important to notice that the discriminant mining task does not aim at extracting $\mathscr{C}^+$ and $\mathscr{C}^-$ from $\mathscr{S}$. Its goal is to extract patterns describing the maximum number of occurrences of $\mathscr{C}^+$

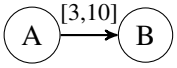
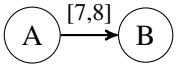
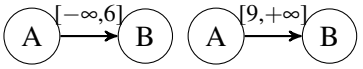
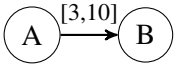
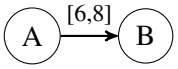
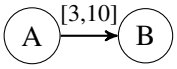
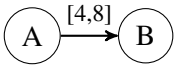
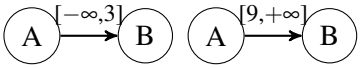
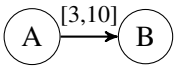
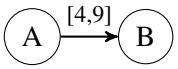
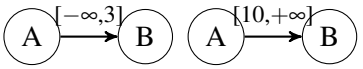
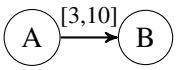
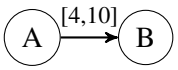
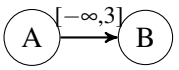
Dataset	$\mathscr{C}^+$	$\mathscr{C}^-$	Expected discriminant chronicles
$Base_1$			
$Base_2$			
$Base_3$			
$Base_4$			
$Base_5$			

Table 4.6: Chronicles introduced in each dataset type and expected discriminant chronicles for $\mathscr{S}^+$ in those datasets for each pair of introduced chronicles. As discriminant chronicles of $Base_1$ are more general than those of $Base_5$, $Base_1$ sequences are easier to discriminate than those of $Base_5$.

and the minimum number of occurrences of $\mathscr{C}^-$. The third column of Table 4.6 gives the expected discriminant chronicles to extract.

For each dataset type $Base_N$, the results are averaged over 20 datasets generated independently. The simulator is set to generate by default datasets containing 800 sequences. The average length of sequences is 10. The size of the vocabulary is set at 50 items and $\widehat{f}_{min}$ and $\widehat{g}_{min}$ are respectively set to 80% and 10%.

The ability of DCM to extract the correct patterns is evaluated by comparing the sets of extracted patterns with the expected patterns. Two measures are used: $\Delta c_{(m,\widehat{m})}$ the coverage ratio and $\Delta g_{(m,\widehat{m})}$ the growth ratio. The variable m and $\widehat{m}$ denote extracted and expected patterns respectively.

- $\Delta c_{(m,\widehat{m})} = \dfrac{|\{s \in \mathscr{S}^+ \mid m \in s\} \cap \{s \in \mathscr{S}^+ \mid \widehat{m} \in s\}|}{|\{s \in \mathscr{S}^+ \mid \widehat{m} \in s\}|}$ is a **coverage ratio** similar to the unilateral Jaccard similarity [Santisteban and Tejada-Cárcamo, 2015] coefficient and represents the ratio between the number of sequences of the $\mathscr{S}^+$ class containing one occurrence shared by m and $\widehat{m}$ and the number of sequences in $\mathscr{S}^+$ containing $\widehat{m}$. This measure determines how much similar are the sets of sequences labeled as positive by the two patterns. It is a recall measure.
- $\Delta g_{(m,\widehat{m})} = \begin{cases} min\left(\dfrac{g_{(m,\mathscr{S})}}{g_{(\widehat{m},\mathscr{S})}}, 1\right) & \text{, if } g_{(\widehat{m},\mathscr{S})} < +\infty \\ 1 & \text{, if } g_{(m,\mathscr{S})} = g_{(\widehat{m},\mathscr{S})} = +\infty \\ 0 & \text{otherwise} \end{cases}$
  is the **growth ratio** and represents the ratio of the growth rates of m to $\widehat{m}$.

For each expected patterns $\widehat{m}$, a single pair $(\Delta c_{(m,\widehat{m})}, \Delta g_{(m,\widehat{m})})$ is retained such that $\Delta c_{(m,\widehat{m})}$ is the highest for any extracted pattern m and that $\Delta g_{(m,\widehat{m})}$ is the highest if there are several $\Delta c_{(m,\widehat{m})}$ maximal. To simplify the notations, $(\Delta c, \Delta g)$ is used to denote this pair. In addition to the previous measure used as recall, we use this measure as accuracy.

4.5.1.2 Results

Figure 4.6 shows the extraction results of DCM on the 20 datasets generated for each $Base_N$. The coverage ratio is given in abscissa and the growth ratio in ordinates. Given a coverage ratio Δg

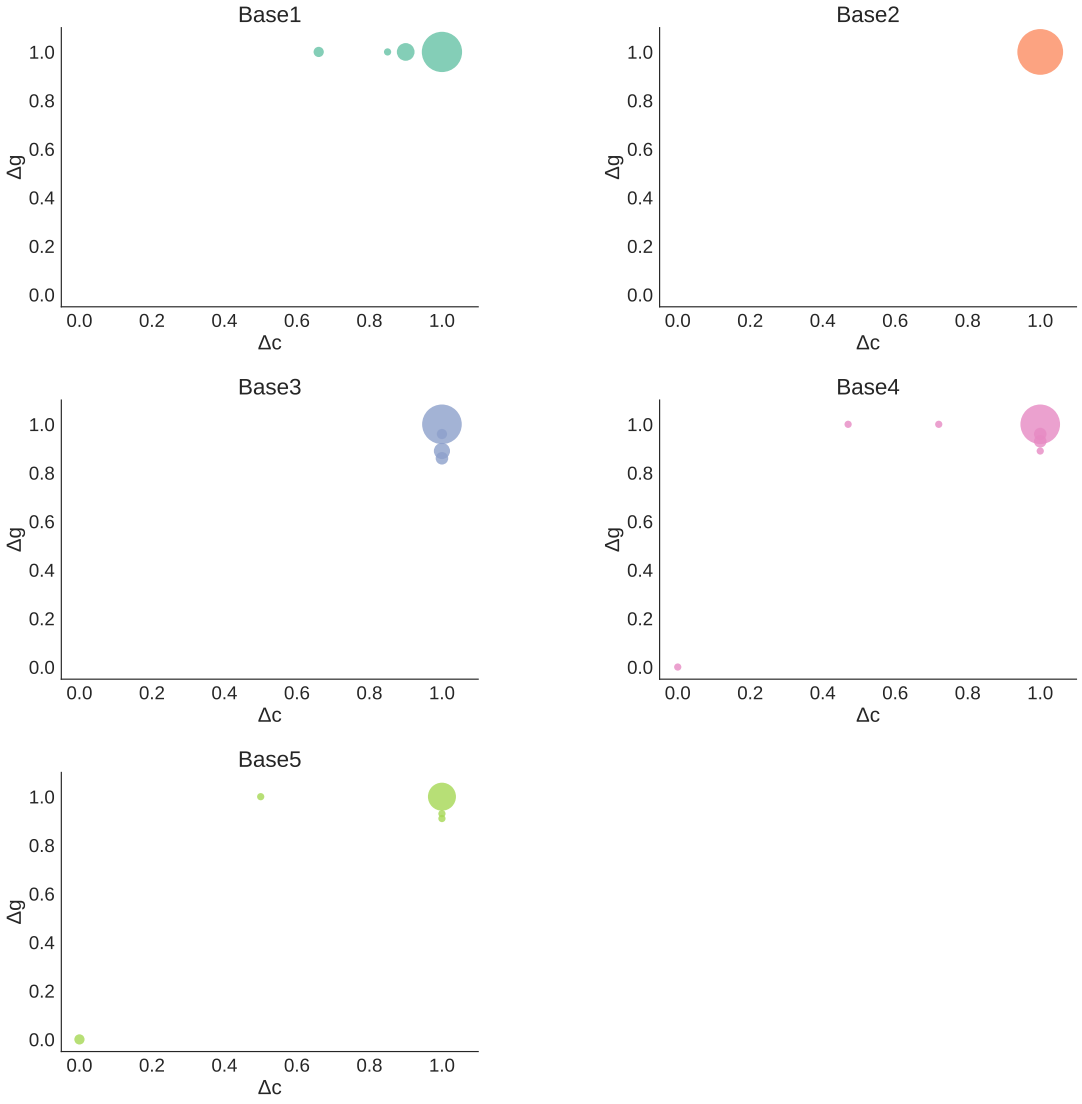


Figure 4.6: Scatter plots of the extracted patterns represented by $(\Delta c, \Delta g)$ for each dataset. For each expected discriminant pattern $\widehat{m}$ only the best extracted pattern m according to $(\Delta c_{(m,\widehat{m})}, \Delta g_{(m,\widehat{m})})$ is used in these plots. The dot size represents the number of extracted patterns.

and a growth ratio Δg, the dot size gives the number of chronicles similar to one of the expected chronicles (see Table 4.6). This number is the sum of patterns for the 20 datasets, meaning that the maximal dot size is 40 for database 1 to 4 (20 datasets with 2 expected patterns) and 20 for database 5.

The biggest dots per database, from 1 to 5, represent respectively 31, 40, 30, 30 and 15 extracted chronicles that was expected.

We notice on Figure 4.6 that DCM extracts at least 75% of the expected discriminant patterns in each dataset a discriminant pattern whose coverage and growth ratios are perfect ($\Delta c = 1$ and $\Delta g = 1$). In addition, for the $Base_4$ and $Base_5$ datasets, DCM does not extract any patterns for certain introduced discriminant patterns, but only 1 out of 40 for $Base_4$ and 2 on 20 for $Base_5$.

To explain these errors, we extract chronicles from the same dataset with a minimum growth rate of 1.1. With this setting, all expected $Base_4$ patterns match with at least extracted pattern (with at least $\Delta c = 1$ and $\Delta g = 0.83$). It is worth noting that the extracted pattern corresponding to

$\Delta c = 0.47$ for the first extraction of the *Base*4 patterns reaches the same Δc and Δg ratios ($\Delta c = 1$ and $\Delta g = 0.83$).

These results on simple synthetic data show that DCM actually extracts the discriminating patterns with robustness. For datasets that can only be discriminate through the temporal dimension, DCM extracts the useful discriminant temporal constraints representing the expected discriminant patterns.

4.5.2 *BIDE-D* comparison

We compare DCM to the *BIDE-D* algorithms on datasets from [Fradkin and Mörchen, 2015]. These datasets come from a variety of applications. In order to compare DCM with *BIDE-D* because these datasets are neither as simple as *blocks* nor as *Auslan*2, we chose to focus on *asl-bu*, *asl-gt* and *context*.

- **asl-bu**: The intervals are transcriptions from videos of American Sign Language expressions provided by Boston University [Papapetrou et al., 2005]. It consists of observation interval sequences with labels such as *head mvmt: nod rapid* or *shoulders forward* that belong to one of 7 classes like *yes-no question* or *rhetorical question*.

- **asl-gt**: The intervals are derived from 16-dimensional numerical time series with features derived from videos of American Sign Language expressions [Starner et al., 1998]. The numerical time series are discretized into $2-4$ states. Each sequence represents one of the 40 words such as *brown* or *fish*.

- **context**: The intervals are derived from categoric and numeric data describing the context of a mobile device carried by humans in different situations [Mäntyjärvi et al., 2004]. Numeric sensors are discretized using $2-3$ bins chosen manually based on exploratory data analysis. Each sequence represents one of five scenarios such as *street* or *meeting*.

These datasets are databases of intervals interpreted as sequential databases by considering beginning and ending boundaries of an interval as two distinct events.

The couple $\langle \mathscr{C}, L \rangle$ of discriminant chronicles $\mathscr{C}$ and label $L \in \mathbb{L}$, is used to predict the label of a sequence. A label L will be predicted for a sequence in case this sequence contains the chronicle $\mathscr{C}$. In the case of several chronicles appearing in the sequence, we label the sequence with the predicted label by the chronicle with the highest growth rate. This simple classification method is more accurate than the use of a standard machine learning classifier to evaluate the extracted chronicles. On the other hand, discriminant chronicles accuracy would not be overrated compared to *BIDE-D* approaches. Indeed, the *BIDE-D* approaches use a *SVM* classifier and *SVM* is generally more accurate than a simple rule-based classifier.

The results discussed below are presented in Table 4.7. We set a size limit of 6 items per chronicle in order to limit the number of patterns. The results presented in [Fradkin and Mörchen, 2015] are given in Table 4.8.

On *asl-gt* *BIDE-D* outperforms DCM. Where accuracy ranges from 27.31% for $\sigma_{min} = 0.6$ to 82.94% for $\sigma_{min} = 0.2$ for *BIDE-D*, it hardly exceeds 30% for discriminant chronicles. The standard deviation of *BIDE-D* and DCM are similar. Many patterns are extracted but are poorly distributed among the dataset labels. While discriminant chronicles can represent discriminant sequential patterns, these bad results on *asl-gt* are due to a lack of discriminant patterns. This lack is due to the large number of classes that are 40 for *asl-gt*. As a result, a single pattern is difficulty discriminant. The results obtained for *BIDE-D* are due to the selection of the best discriminant patterns and not to a minimal growth rate threshold. Thereby, even if there is no discriminant sequential patterns in *asl-gt*, the *BIDE-D* approaches extract some patterns to use in the classifier.

dataset	σ_{min}	g_{min} = 2	3	4	5
asl-bu	**0.4**	52.64 (±2.21)	49.20 (±0.51)	52.41 (±3.51)	48.28 (±1.99)
	0.5	51.26 (±1.31)	51.72 (±1.99)	49.66 (±6.82)	48.51 (±5.60)
	0.6	51.72 (±3.35)	50.34 (±3.58)	44.14 (±3.31)	39.08 (±3.72)
asl-gt	**0.2**	31.55 (±0.91)	31.61 (±0.92)	30.20 (±1.82)	30.15 (±0.92)
	0.3	31.17 (±0.44)	29.18 (±1.53)	27.75 (±1.58)	26.96 (±1.89)
	0.4	27.34 (±2.10)	25.82 (±0.42)	25.91 (±0.12)	25.32 (±0.19)
	0.5	25.44 (±0.34)	25.20 (±0.13)	24.68 (±0.50)	24.12 (±0.41)
	0.6	24.30 (±0.42)	23.92 (±0.53)	23.89 (±0.52)	23.13 (±0.44)
context	**0.2**	64.78 (±2.83)	57.39 (±4.76)	46.09 (±3.89)	53.48 (±6.07)
	0.3	56.09 (±5.83)	42.61 (±7.62)	52.61 (±3.22)	36.96 (±7.53)
	0.4	47.83 (±4.07)	39.57 (±3.57)	50.43 (±5.41)	47.39 (±4.96)
	0.5	53.91 (±4.46)	38.70 (±0.97)	30.43 (±5.10)	47.83 (±7.37)
	0.6	50.87 (±2.48)	34.78 (±4.35)	30.87 (±4.18)	28.70 (±4.46)

Table 4.7: Mean accuracy (in %) with respect to σ_{min} (minimal support threshold), g_{min} (minimal growth rate threshold) for different datasets.

On *asl-bu* DCM performs slightly worse than *BIDE-D*. The accuracy using $g_{min} = 2$ is equivalent from $\sigma_{min} = 0.4$ to $\sigma_{min} = 0.6$, whereas the number of extracted patterns is reduced from more than $30,000$ to $1,600$. DCM's accuracy exhibits better performance than *BIDE-D* in terms of standard deviation. We can notice that *asl-bu* is a difficult type of dataset for chronicle mining due to multiple occurrences of the same event types. As a result we do not obtain a result for $\sigma_{min} = 0.2$ or $\sigma_{min} = 0.3$. DCM takes too long time and does not end for these parameters within the 12 hour timeout.

Finally, on *context*, DCM outperforms *BIDE-D* in accuracy. The accuracy standard deviation is higher than for the two other datasets but is, however, similar to the one of *BIDE-D*. We note that the low results of *BIDE-D* approaches may be due to the use of a different minimum support threshold strategy for this dataset. The difference in the number of patterns extracted by the $\sigma_{min} = 0.2$ or $\sigma_{min} = 0.6$ parameters is much smaller than for *asl-bu*. The count goes from 360 patterns for $\sigma_{min} = 0.2$ and $g_{min} = 2$ to 145 for $\sigma_{min} = 0.6$ and $g_{min} = 4$.

We conclude from these results that discriminant chronicles as rules can produce similar and sometimes better accuracy than sequential patterns as features. Furthermore, our results show that it seems easier to obtain a good accuracy for datasets containing few classes (*e.g. context*: 5 classes) than datasets for which the sequences are distributed in a large set of classes (*e.g. asl-gt*: 40 classes). Indeed, a large class number implies a generalization problem. A pattern discriminating a class against 39 other classes has to be more specific than a pattern discriminating a class against 4 other classes. This generalization problem comes from the unbalanced distribution between positive and negative sequences that requires to be very specific to obtain discriminant patterns. This problem is less important for the *BIDE-D* approaches because the *SVM* classifier learns combinations between discriminant patterns extracted for a subset of classes to generalize it for all classes. Thereby, the *SVM* classifier can predict a class without using discriminant pattern for this class but it is not possible with discriminant chronicles used as rules. The definition of an extracted discriminant patterns is also different for DCM and *BIDE-D* and affects this generaliza-

dataset	σ_{min}	BIDE-C	BIDE	BIDE-D (k=90)	BIDE-DC (k=90)
asl-bu	**0.2**	60.01 (± 5.47)	59.71 (± 4.87)	52.91 (± 4.15)	56.01 (± 4.62)
	0.3	58.05 (± 4.90)	58.50 (± 4.40)	52.83 (± 4.03)	57.89 (± 5.00)
	0.4	57.74 (± 4.03)	58.42 (± 4.15)	53.06 (± 3.91)	59.03 (± 5.09)
	0.5	57.74 (± 4.03)	58.42 (± 4.15)	53.06 (± 3.91)	59.03 (± 5.09)
	0.6	55.25 (± 5.58)	58.80 (± 4.37)	53.44 (± 4.18)	57.21 (± 5.40)
asl-gt	**0.2**		82.78 (± 1.31)	82.94 (± 1.07)	82.30 (± 1.52)
	0.3		73.16 (± 1.33)	73.14 (± 1.43)	73.17 (± 1.32)
	0.4		53.38 (± 1.11)	53.40 (± 1.11)	53.36 (± 1.05)
	0.5		33.67 (± 1.43)	33.70 (± 1.41)	33.68 (± 1.43)
	0.6		27.31 (± 0.72)	27.31 (± 0.72)	27.31 (± 0.72)
context	**0.7**		53.47 (± 6.80)	37.36 (± 3.73)	51.53 (± 6.93)
	0.8		48.75 (± 7.29)	48.89 (± 6.34)	47.92 (± 5.94)
	0.9		26.25 (± 4.08)	27.08 (± 3.78)	26.53 (± 4.27)

Table 4.8: Mean accuracy (in %) with respect to σ_{min} (minimal support threshold) and the different methods proposed in [Fradkin and Mörchen, 2015] for different datasets. The methods *BIDE-D* and *BIDE-DC* extract discriminant sequential patterns. *BIDE* and *BIDE-C* extract frequent sequential patterns. Support thresholds used to obtain results from context are much higher than for the other datasets.

tion problem. A pattern is extracted by DCM if its growth rate is greater than a threshold. In this case, it is possible to extract no patterns for a threshold. A pattern is extracted by *BIDE-D* if its growth rate is one of the highest. Thereby, it is ensured that some patterns will be extracted and used as features by the classifier. Finally, DCM is easy to set up while a grid search is needed to obtain the parameters of the *SVM* that use the sequential patterns as features (the standard *SVM* parameter *C* is a power of 10 and is tested from 10^{-3} to 10^{3}).

4.6 Conclusion

Chronicles mining was already known but in a task of frequent chronicle mining. The originality of this work is the mining of discriminant chronicles. To the best of our knowledge, no previous work on discriminant chronicles mining, neither on any discriminant pattern mining studies with quantitative temporal information. The DCM algorithm has been proposed to solve this problem. It allows, on the one hand, to take into account a rich temporal information in comparison with algorithms dedicated to discriminant sequential patterns mining and, on the other hand, to constrain the extracted chronicles with a minimal growth rate constraint. This algorithm relies on rule learning to discover the discriminant rules efficiently from quantitative attributes. Furthermore, DCM is a modular algorithm because different rule learning algorithms can be used to extract the discriminant temporal constraints.

The experiments using the $Ripper_k$ algorithm showed that DCM was able to extract discriminant chronicles efficiently. Comparisons in terms of accuracy between DCM and the *BIDE-D* [Fradkin and Mörchen, 2015] algorithms showed that discriminant chronicles extracted with DCM are able to compete with state of the art approaches without having to involve a classifier like *SVM*.

The patterns extracted by DCM can be interpreted by experts.

The "global model" dedicated to classification is not the main goal of discriminant chronicles mining. Furthermore, we did not design discriminant chronicles mining as local pattern discovery to construct a global model like for the *LeGo* [Knobbe et al., 2008] framework. Discriminant chronicles mining has been thought in the context of hypothesis generation:

- The association of a chronicle to a class label is a possibly interesting hypothesis to further investigate.
- Each extracted chronicle has to be interpreted independently.

Classification is used to evaluate the capacity of DCM to generalize sequences. Such interest of the chronicle model has to be more deeply studied in the following parts.

Chapter 5

Interpretable temporal pattern-based classification

We are interested in extracting patterns that will be presented to the experts. This requires to have easily interpretable patterns. We mainly work on temporal data, specifically temporal sequence sets. Temporal patterns are dedicated to catch the complexity of temporal sequences. Thereby, we only interest in this type of pattern. In temporal pattern model set, chronicle seems to be a good model to represent complex candidate behaviors with good interpretability.

The number of extracted frequent patterns can be huge for large datasets then the whole pattern set can be difficult to manage. It could not be presented as it to the experts:

1. They will be overwhelmed by the mass of pattern data.

2. Navigate in those data will consume a lot of time that they may not take.

3. The computing time to extract all those patterns could not be reasonable.

Furthermore, the interest of a frequent pattern mining approach is limited. It generates more results than in the original analyzed dataset. In this case, read the original dataset could be easier than read the extracted pattern set.

Methods dedicated to extract less but more interesting patterns can be useful to represent a concise set of new knowledge. When the data are labeled, one interesting constraint to extract less but more interesting patterns is the minimal growth rate constraint. This constraint prunes the frequent pattern set to keep only the pattern occurring more in a class than the others. A pattern satisfying this constraint for a label is said discriminant for it. But in practice, such constraint is not good enough to really prune the frequent pattern set. The use of incomplete heuristics like the use of $Ripper_k$ for extracting discriminant chronicles is then needed to reduce the size of the pattern set.

It is not easy to evaluate the ability of a pattern or a pattern set to represent knowledge. Some papers use the argument of authority that experts found the extracted patterns are interesting or useful or that the results match the expert knowledge to validate their approaches [Fabrègue et al., 2014, Fabrègue et al., 2013, Huang et al., 2012]. This type of conclusion is necessary when the goal of the approach is to propose a better solution than the existing one to a practical problem. But it is limited to generalize the interest of the approach to other problems. Indeed, this way of evaluation is difficulty reproducible and does not seem appropriate to evaluate the efficiency of the pattern model in general. Some works try to provide objective measures through pattern-based classification [De Smedt et al., 2017, Fradkin and Mörchen, 2015].

Classification has the advantage to sum up an approach to a bunch of interpretable measures like accuracy, recall, f-measure, etc. The hypotheses done here are: (i) if the model can discriminate the dataset classes and (ii) if the algorithm dedicated to extract this pattern type from a dataset covers sufficiently well the whole dataset, then the model accuracy in terms of classification will be good. Through these hypotheses we could then compare the algorithm and/or pattern model to determine which one is the best to extract/represent the knowledge of a dataset.

An issue occurs when we try to experiment pattern-based classification: How do we use patterns in classification? Most of the works on temporal pattern-based classification based classification on a black box classifier [Batal et al., 2013, Fradkin and Mörchen, 2015]. Each sequence is transformed into a feature vector such that each feature f_i of a feature vector v_j is a boolean b_i^j such that $b_i^j = 1$ if the pattern P_i occurs in the sequence s_j and otherwise $b_i^j = 0$. Then, these feature vectors are classified using accurate classifiers such that SVM. This approach generates generally accurate models but does not give direct information about the classification power of the pattern set or its interestingness. This lack of information is due to the use of such classifiers. It is not a problem when the goal is to build an accurate classifier. But it is difficult to correlate the classification results with the classifier model or with the pattern set. Thereby, a black box classifier could not be used to evaluate the interestingness or the classification power of a pattern set through objective measure like accuracy.

In [Fradkin and Mörchen, 2015], we can notice that the best accurate approach is the one using *BIDE i.e.* using the whole set of frequent sequential patterns to be used as features. The classification power of a pattern set used as features is related to the capacity of the classifier to use the good patterns. The best quality of a pattern set to improve the accuracy in such context is to cover as large as possible the whole dataset. That is why the whole frequent pattern set provides the best results.

With such conclusion, studies like [Batal et al., 2013, Fradkin and Mörchen, 2015] try to reduce the number of patterns used as features while reducing at least the accuracy. We can suppose that a smaller pattern set with same accuracy is more interesting than the initial one but the accuracy power comes still from the classifier. The accuracy obtained by the classifier using all frequent patterns is then a baseline for the other classifiers. From this accuracy, we can then evaluate the trade-off between the loss of accuracy and the gain in interpretability.

The use of a black box classifier only provides the information that the pattern set contains enough knowledge or not to classify a dataset. On the other hand, if the classifier is difficultly interpretable, it is more difficult to know if an expert could apprehend this knowledge. It is contrary to the goal of the pattern mining task that is to extract interpretable new knowledge. We are then also interested in an interpretable classification method dedicated to evaluate a pattern set. As for the pattern set, the best accuracy obtained by the best black-box classifier could be a baseline to evaluate the loss of accuracy using interpretable classifiers.

We are then interested by two levels of interpretability in pattern-based classification:

1. the interpretability of the pattern type

2. the interpretability of the classification process

In this chapter we will begin by defining generalized chronicles and classification using this type of pattern. This definition will then be used to constrain this work on a subset of temporal patterns. Later, we will discuss the generalized chronicle interpretability and the trade-off between the accuracy and the interpretability. Finally, we will use several types of generalized chronicles to classify synthetic and real datasets to experimentally test this trade-off.

5.1 Generalized chronicles

In this section, we define generalized chronicles to our case study. To illustrate our definitions, several examples of temporal pattern models are presented and compared. For those models, we choose an occurrence mapping function using the numerical temporal information of the multiset occurrences.

From chronicle definition in section 2.2.1, a **temporal pattern** $\mathscr{P} = (\mathscr{E}, \mu_{\mathscr{T}})$ is a generalization of a chronicle $\mathscr{C} = (\mathscr{E}, \mathscr{T})$ where $\mathscr{E}$ is a multiset and $\mathscr{T}$ is generalized as an **occurrence mapping function** $\mu_{\mathscr{T}}$. For a pattern $\mathscr{P} = (\mathscr{E}, \mu_{\mathscr{T}})$, we can define $\mu_{\mathscr{T}}$ as $\mu_{\mathscr{T}} : \mathbb{N}^{|\mathscr{E}|} \to [0,1]$ where the domain $\mathbb{N}^{|\mathscr{E}|}$ contains all the potential occurrences of the multiset $\mathscr{E}$ where an occurrence is represented by the timestamps of each event in $\mathscr{E}$. Defining the codomain of $\mu_{\mathscr{T}}$ as $[0,1]$ corresponds to define $\mu_{\mathscr{T}}(o)$ as the similarity between o and the pattern $\mathscr{P}$. In practice, the standard temporal and sequential pattern such that chronicles, sequential patterns or episodes are defined with $\{0,1\}$ as codomain of $\mu_{\mathscr{T}}$. In this case, an occurrence o of a multiset $\mathscr{E}$ can only be exactly an occurrence of the pattern $\mathscr{P} = (\mathscr{E}, \mu_{\mathscr{T}})$ with $\mu_{\mathscr{T}}(o) = 1$ or exactly not an occurrence of $\mathscr{P}$ with $\mu_{\mathscr{T}}(o) = 0$.

Example 12. For chronicles the occurrence mapping function $\mu_{\mathscr{T}}$ corresponds to the mapping of the temporal constraints $\mathscr{T}$ in the occurrences of the multiset. From the definition of a chronicle occurrence in Section 2.2.1.1, we can formally define $\mu_{\mathscr{T}}$ for chronicle as:

$$\mu_{\mathscr{T}}(o) = \begin{cases} 1 & if \quad \forall e_i[a,b]e_j \in \mathscr{T}, a \leq o_j - o_i \leq b \\ 0 & otherwise \end{cases}$$

Example 13. For sequential patterns, the occurrence mapping function corresponds to the order between the events of the multiset. For example, the sequential pattern $\langle a,b,c \rangle$ could be represented as the temporal pattern $\mathscr{P} = (\mathscr{E}, \mu_{\mathscr{T}})$ where $\mathscr{E} = \{\!\{a,b,c\}\!\}$ and $\mu_{\mathscr{T}}$ corresponds to the matching of the order of the three events. This type of pattern can be defined as chronicle with its own $\mu_{\mathscr{T}}$. To do this, $\mathscr{T}$ as to be defined for sequential patterns. Let $\mathbb{O} = \{<, >, =\}$ be the set of operators used to define sequential patterns. The temporal constraint set can be now defined as $\mathscr{T} = \{(i, j, \lessgtr) \mid e_i, e_j \in \mathscr{E}, e_i \leq_{\mathscr{E}} e_j, \lessgtr \in \mathbb{O}\}$. A sequential pattern $\mathscr{P}$ can then be defined for such $\mathscr{T}$ as $\mathscr{P} = (\mathscr{E}, \mu_{\mathscr{T}})$ for $\mu_{\mathscr{T}}$ defined by:

$$\mu_{\mathscr{T}}(o) = \begin{cases} 1 & if \quad \forall (i, j, \lessgtr) \in \mathscr{T}, o_i \lessgtr o_j \\ 0 & otherwise \end{cases}$$

Example 14. The example above describe one way to define a new $\mu_{\mathscr{T}}$ for sequential patterns. This is just one way to do it because, for example, we could use the same $\mu_{\mathscr{T}}$ as for chronicle in example 12 and simply constraining the temporal constraint set $\mathscr{T}$. The temporal constraint set $\mathscr{T}$ will be the same except that the domain of the temporal constraint will be constrained like that $\mathscr{T} = \{e_i[a,b]e_j \mid e_i, e_j \in \mathscr{E}, e_i \leq_{\mathscr{E}} e_j, [a,b] \in \{[0,0], [1,+\infty[,]-\infty,-1]\}\}$. We consider $[1,+\infty[$ and $]-\infty,-1]$ to use inclusive bounds like in our chronicle implementation but those constraints are equivalent to $]0,+\infty[$ and $]-\infty,0[$ if the temporal domain $\mathbb{T}$ is included in $\mathbb{R}$.

Example 15. As the last example, we can relax the definition of sequential patterns to obtain partially ordered patterns[1] [Fabrègue et al., 2013] (or episodes). To do this we can include a new operator $\approx$ in the set $\mathbb{O}$ from the example 13 that defines no order relation *i.e.* $\forall o_i, o_j, o_i \approx o_j \leftrightarrow \top$. We can also model this type of pattern from the example 13 by relaxing the definition of $\mathscr{T}$ itself. As the difference between sequential and partially ordered patterns is that there is no order

[1] Partially ordered patterns are patterns for which the events are partially ordered and not a set of patterns that are partially ordered.

relation between some events for the second model of patterns, we can just define $\mathscr{T}$ as a set of relations on some pair of events. Formally, for partially ordered patterns, $\mathscr{T}$ can be defined as $\mathscr{T} = \{(i, j, \lessgtr) \mid e_i, e_j \in \mathscr{E}', e_i \leq_{\mathscr{E}} e_j, \mathscr{E}' \subseteq \mathscr{E}, \lessgtr \in \mathbb{O}\}$. Finally, we can also define partially ordered patterns from the definition of the example 14 by adding $[-\infty, +\infty]$ to the possible interval for a temporal constraint. $\mathscr{T}$ will then be defined as $\mathscr{T} = \{e_i[a,b]e_j \mid e_i, e_j \in \mathscr{E}, e_i \leq_{\mathscr{E}} e_j, [a,b] \in \{[0,0], [1,+\infty[,]-\infty,-1],]-\infty,+\infty[\}\}$.

All pattern models of the previous examples can be represented with a $\mu_{\mathscr{T}}$ function with the codomain $\{0,1\}$. Those pattern models could be represented by a generic $\mu_{\mathscr{T}}$ function validating or not the constraints of $\mathscr{T}$ or not. This definition of temporal pattern can also be extended to pattern models that could not be represented by the chronicle model. For example $\mu_{\mathscr{T}}$ could be a *SVM* classifier. Like the temporal constraints of chronicles, the *SVM* classifier of this type of pattern will be used to determine whether an occurrence of a multiset $\mathscr{E}$ of a pattern $\mathscr{P} = (\mathscr{E}, \mu_{\mathscr{T}})$ is an occurrence of the pattern $\mathscr{P}$ itself or not.

5.1.1 Discriminant generalized chronicles

The temporal pattern definition presented above can be extended for discriminant temporal patterns. The goal of a discriminant pattern is to represent a behavior of a certain class in the data. We can then define a discriminant temporal patterns as a temporal pattern for which the $\mu_{\mathscr{T}}$ function will give information on the label corresponding to the occurrence of the multiset and not only if the occurrence corresponds to the pattern. More formally for a label set $\mathbb{L}$, $\mu_{\mathscr{T}}$ can be extended from $\mu_{\mathscr{T}} : \mathbb{N}^{|\mathscr{E}|} \to [0,1]$ to $\mu_{\mathscr{T}} : \mathbb{N}^{|\mathscr{E}|} \to [p_1, p_2, ..., p_{|\mathbb{L}|+1}]$ where $0 \leq p_i \leq 1$ and $p_{|\mathbb{L}|+1}$ correspond to the dissimilarity with all the known labels. This is an extension of the previous definition of $\mu_{\mathscr{T}}$ because each p_i corresponds to the use of $\mu_{\mathscr{T}} : \mathbb{N}^{|\mathscr{E}|} \to [0,1]$ on the data of the label $l_i \in \mathbb{L}$. One more p_i is used as $p_{|\mathbb{L}|+1}$ to represent the case when an occurrence seems not linked to any label from $\mathbb{L}$. It is useful to define $[p_1, p_2, ..., p_{|\mathbb{L}|+1}]$ with only the constraint $0 \leq p_i \leq 1$ for cases where an occurrence could be linked to several labels. If each sequence is labeled with one and only one label, it can be interesting to define the vector $[p_1, p_2, ..., p_{|\mathbb{L}|+1}]$ as a probability vector. This is done by constraining $[p_1, p_2, ..., p_{|\mathbb{L}|+1}]$ with the constraint $\sum_{i=1}^{|\mathbb{L}|+1} p_i = 1$.

Example 16. For example, we can extend the definition of chronicle proposed in the example 12 to the definition of discriminant chronicle. A discriminant chronicle is a chronicle extracted for a certain label. For a discriminant chronicle $\mathscr{C}$ extracted for a label l_i, we can then define that $\mu_{\mathscr{T}}(o)$ equals a vector such that $p_i = 1$ and $\forall j, j \neq i, p_j = 0$ if o is an occurrence of $\mathscr{C}$ or a vector such that $p_{|\mathbb{L}|+1} = 1$ and $\forall j, j \neq |\mathbb{L}| + 1, p_j = 0$ if o is not an occurrence of $\mathscr{C}$. Formally $\mu_{\mathscr{T}}$ can be defined as:

$$\mu_{\mathscr{T}}(o) = \begin{cases} p_i = 1, \forall j, j \neq i, p_j = 0 & if \quad \forall e_i[a,b]e_j \in \mathscr{T}, a \leq o_j - o_i \leq b \\ p_{|\mathbb{L}|+1} = 1, \forall j, j \neq |\mathbb{L}| + 1, p_j = 0 & otherwise \end{cases}$$

Example 17. The example 16 can also be adapted to obtain the same pattern model as the one explained in Section 4.4 to use chronicles as rules. To determine which chronicle should be used for labeling, a biased growth rate was computed for each chronicle and the chronicle occurring in a sequence with the highest growth rate was used for the labeling. Such type of pattern can easily be handled by a $\mu_{\mathscr{T}}$ function returning a probability vector for an occurrence. The biased growth rate $Q_g = \dfrac{supp(\mathscr{C}, l_i)}{supp(\mathscr{C}, l_i) + supp(\mathscr{C}, \overline{l_i}) + g}$ of a chronicle $\mathscr{C}$ extracted for the label l_i can be integrated in $\mu_{\mathscr{T}}(o)$ with $p_i = Q_g$ and $p_{|\mathbb{L}|+1} = 1 - Q_g$ if o is an occurrence of $\mathscr{C}$. Formally we obtain the definition:

SID	$\mathscr{A}_{A\to B}$	$\mathscr{A}_{B\to C}$	$\mathscr{A}_{A\to C}$	Label
1	2	2	4	+
1	−1	2	1	+
2	5	−2	3	+
3	3	0	3	+
5	−1	3	1	−
6	6	−1	5	−

SID	$\mathscr{A}_{A\to B}$	$\mathscr{A}_{B\to C}$	$\mathscr{A}_{A\to C}$	Label
1	1	1	1	+
1	−1	1	1	+
2	1	−1	1	+
3	1	0	1	+
5	−1	1	1	−
6	1	−1	1	−

Table 5.1: On the right, the tabular dataset for the multiset $\{\!\{A,B,C\}\!\}$ corresponding to its occurrences in Table 4.1 and encoding for discriminant partially ordered mining. On the left, the original occurrence dataset.

$\mu_{\mathscr{T}}(o) =$

$$\begin{cases} p_i = Q_g, p_{|\mathbb{L}|+1} = 1-Q_g, \forall j, j\neq i, j\neq |\mathbb{L}|+1, p_j = 0 & if \quad \forall e_i[a,b]e_j \in \mathscr{T}, a \leq o_j - o_i \leq b \\ p_{|\mathbb{L}|+1} = 1, \forall j, j\neq |\mathbb{L}|+1, p_j = 0 & otherwise \end{cases}$$

An issue with this definition is that the definition of $\mu_{\mathscr{T}}(o)$ depends on Q_g and the definition of Q_g depends on the support of the chronicle that depends on $\mu_{\mathscr{T}}(o)$. In practice Q_g have to be computed in the first step using the classical definition of chronicle and then this type of pattern can be extracted in a second step.

5.1.1.1 Discriminant partially ordered patterns

This section describes the discriminant partially ordered patterns as they are implemented and extracted in the experiment section 5.2. Partially ordered patterns [Fabrègue et al., 2014, Fabrègue et al., 2013] is the name of patterns equivalent to episodes [Mannila et al., 1997] but extracted from sequence sets. In example 15 of Section 5.1 dedicated to the generalization of the chronicle model, we formalized the partially ordered pattern model by redefining the temporal constraint set $\mathscr{T}$ in several ways. The one that is interesting for this section is the one defining $\mathscr{T}$ as a set of temporal constraints $\mathscr{T} = \{e_i[a,b]e_j \mid e_i, e_j \in \mathscr{E}, e_i \leq_{\mathscr{E}} e_j, [a,b] \in \{[0,0],[1,+\infty[,]-\infty,-1],]-\infty,+\infty[\}\}$.

The goal here is to extract different types of patterns than chronicle while reusing the DCM principle. It is possible to constrain our algorithm of discriminant chronicles mining to extract pattern with a similar definition of $\mathscr{T}$. Indeed, at the step where we transmit the relational dataset encoding the occurrences of a multiset to $Ripper_k$, we can modify this dataset to avoid numerical rule extraction. This can be done by replacing all strictly positive integers by 1 and all strictly negative integers by -1. One problem could be the artificial satisfaction of the constraints $[-1,-1]$ and $[1,1]$ due to the replacement of all the integer lower than -1 and all the integer greater than 1 respectively. For example, the table 4.3 in Section 4.2.2 will be replaced by the table 5.1. However, this problem is solved by $Ripper_k$ that prunes unnecessary constraints and so that prefers the rules translated by $]-\infty,-1]$ to $[-1,-1]$ if the dataset did not contain integers lower than -1 and prefers $[1,+\infty[$ to $[1,1]$ if the dataset did not contain integers greater than 1. Extracted patterns after this modification can be defined by the temporal constraint set $\mathscr{T} = \{e_i[a,b]e_j \mid e_i, e_j \in \mathscr{E}, e_i \leq_{\mathscr{E}} e_j, [a,b] \in \{[0,0],[0,+\infty[,[1,+\infty[,]-\infty,-1],]-\infty,0],]-\infty,+\infty[\}\}$. Our version of partially ordered patterns is then a more generalized definition than the standard definition [Fabrègue et al., 2014, Fabrègue et al., 2013] due to the possible constraint $[0,+\infty[$ and $]-\infty,0]$ that can be interpreted as "at same time or after" and "before or at same time" respectively. For example, the pattern $\mathscr{P} = (\{\!\{A,B,C\}\!\}, \{B[0,+\infty[C\})$ could be extracted from Table 5.1 with $supp(\mathscr{P},\mathscr{S}^+) = 2$ and $supp(\mathscr{P},\mathscr{S}^-) = 1$.

5.1.2 Classification based on generalized chronicles

As described in section 3.3, the classification of sequences based on pattern sets is a well-studied way to classify sequences. Pattern-based classification may have two goals:

- Be a standard classification model. As in [De Smedt et al., 2017], the pattern-based classification can be done to obtain a good classifier. The patterns are then extracted in this way and are not initially dedicated to be studied apart from the classifier.

- Be a way to characterize a pattern set. As in [Fradkin and Mörchen, 2015], the pattern-based classification is done directly to obtain a good classifier but to obtain objectives measure for a pattern set. In this way we may prefer to extract a pattern set for reasons that make it interesting without optimizing the accuracy that it can produce. For example in [Fradkin and Mörchen, 2015], they compare small sets of discriminant patterns and the set of all frequent closed patterns in terms of accuracy. The set of all frequent closed patterns frequently obtain better results but they consider the other pattern sets to be more interesting. The important measure is then not the accuracy itself but the difference of accuracy produce by the two pattern sets. This measure is used to determine if the selection of a more interesting pattern set does not hide too hardly the knowledge useful to obtain a good accuracy.

This is this second way that we consider in this section. The classification of sequences based on generalized chronicles is useful for us as an objective way of comparison of several models of generalized chronicles.

Figure 5.1 represents the standard scheme of classification based on generalized chronicles. The classification process is split in two parts dedicated to the main steps: (i) the pattern mining and (ii) the decision process. In the first step patterns are extracted from a train set according to the definition of their pattern model. In the second step the patterns are used by a classifier to label the sequences of a test set. We then have to define how to construct a classifier using generalized chronicles.

In recent papers [Fradkin and Mörchen, 2015, De Smedt et al., 2017], sequential patterns are used as features in classifiers. In this case, a feature is represented by a boolean corresponding to the occurrence of a pattern in a sequence. We can so transform a sequence into a feature vector *i.e.* an occurrence vector. A dataset is then transformed in a feature vector set *i.e.* a feature matrix. This process does not rely on a specificity of the sequential patterns and can be done with chronicles. The example 18 shows how it can be done on the example of the previous chapter 4.1. Standard classifiers like decision trees or SVM classifiers are able to treat such data. The classification is then done by such classifier on datasets transformed into feature matrices. The validation set illustrated on Figure 5.1 is different from the standard definition of validation set. In this scheme the validation set is useful to fit a classifier on pattern occurrences and then can be seen as the train set for this classifier. The difference made between the train and validation sets is done to limit the overfit that could be done during the pattern mining step. If a pattern is very frequent in the train set but not in the other sets, the classifier should not give it a strong importance using the validation set.

Example 18. For this example, we apply the feature transformation process on the dataset of Table 4.1in section 4.1. The same dataset is included on the left of Table 5.2. To produce the occurrence matrix we consider the chronicles of Figure 5.2. The dataset on the right of table 5.2 represents this matrix. We can observe that the matrix is composed of the occurrences of each chronicle in each sequence. As $\mathscr{C}$ occurs in the first sequence, the corresponding value in the matrix is 1. On the other hand, as $\mathscr{C}_2$ does not occur in the first sequence, the corresponding value in the matrix is 0. The goal for a classifier is then to associate each feature vector to a label. For example, we

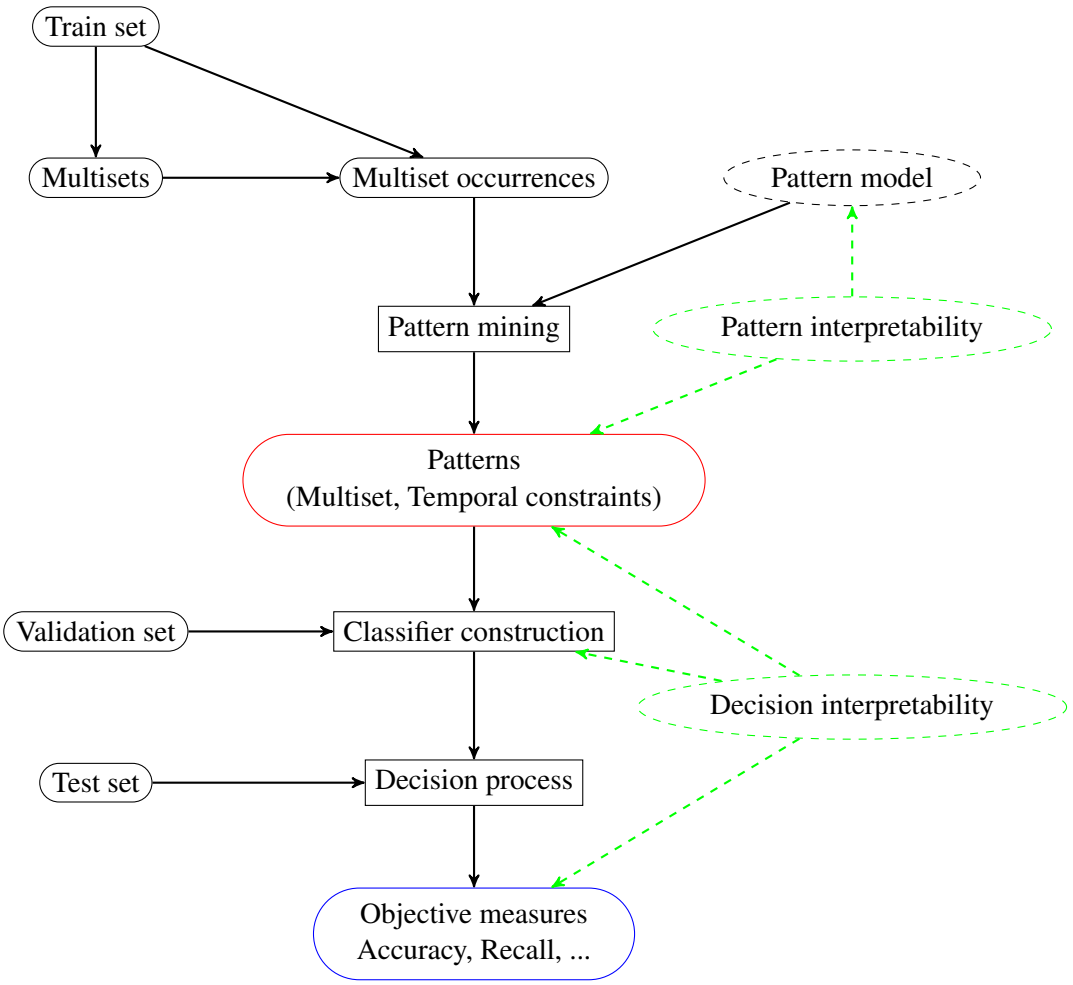


Figure 5.1: Temporal pattern-based classification scheme where the temporal pattern mining step goes from the train set to the patterns (the red node) and the classification goes from the patterns to the obtained objective measures (the blue node).

can associate the vector $\langle 1,1,0\rangle$ to the label $+$. Feature selection approaches can also be applied on this dataset. Such approach could detect that $\mathscr{C}$ is useless as feature because occurrences of $\mathscr{C}_1$ are more correlated to the label than $\mathscr{C}$ and cover the sequences that could be classified using $\mathscr{C}$. Finally, this example cannot produce a perfect classifier because there is no difference between the feature vectors of sequences 2, 4 and 5 while they are labeled differently. The vector $\langle 0,0,0\rangle$ will then potentially be associated with the label $-$ but with an error for sequence 2.

Using the previous definitions, this case corresponds to a matrix of occurrence mapping function results for occurrence mapping functions defined as $\mu_{\mathscr{T}} : \mathbb{N}^{|\mathscr{E}|} \rightarrow [0,1]$. We want so to extend the notion of feature vectors to vectors of occurrence mapping function results for the other type of occurrence mapping function. Formally, for a set of patterns $\mathbb{P}$ and a sequence s the feature vector of s obtained from $\mathbb{P}$ is defined as $\langle \mu_{\mathscr{T}_i}(o) | \mathscr{P}_i = (\mathscr{E}_i, \mu_{\mathscr{T}_i}) \in \mathbb{P}$ and o is an occurrence of $\mathscr{E}_i$ in $s\rangle$. In our cases we do not extract at most one occurrence per sequence but a bag of occurrences per sequence. The occurrence mapping functions have then to handle occurrence bags or have to merge the different results to produce a feature. This merge is needed to avoid the generation of feature vectors with different lengths. The case of multiple instance management for linear *SVM* patterns is described below in paragraph 5.1.2.1.

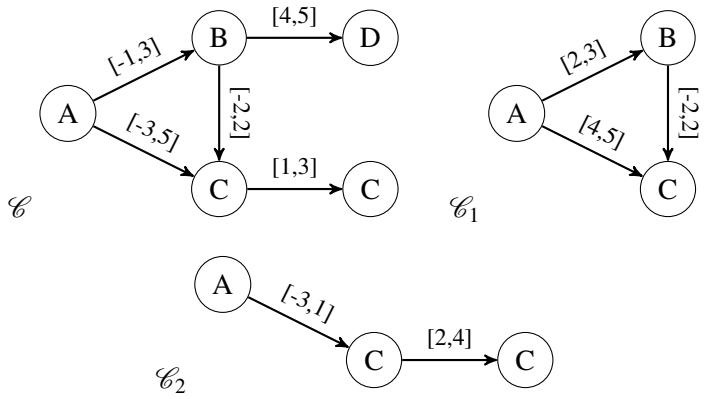


Figure 5.2: Example of three chronicles. This figure is the same as the figure 2.6 in the section 2.2.1.

SID	Sequence	Label
1	$(A,1)$, $(B,3)$, $(A,4)$, $(C,5)$, $(C,6)$, $(D,7)$	$+$
2	$(B,2)$, $(D,4)$, $(A,5)$, $(C,7)$	$+$
3	$(A,1)$, $(B,4)$, $(C,5)$, $(B,6)$, $(C,8)$,$(D,9)$	$+$
4	$(B,4)$, $(A,6)$, $(E,8)$, $(C,9)$	$-$
5	$(B,1)$, $(A,3)$, $(C,4)$	$-$
6	$(C,4)$, $(B,5)$, $(A,6)$, $(C,7)$, $(D,10)$	$-$

SID	$\mathscr{C}$	$\mathscr{C}_1$	$\mathscr{C}_2$	Label
1	1	1	0	$+$
2	0	0	0	$+$
3	1	1	0	$+$
4	0	0	0	$-$
5	0	0	0	$-$
6	1	0	1	$-$

Table 5.2: On the left, the set of six sequences labeled with two classes $\{+,-\}$ previously used as an example in the section 2.2.1. On the right, the corresponding feature matrix corresponding to the occurrences of the chronicles of Figure 5.2.

5.1.2.1 Patterns based on linear *SVM*

This section describes the implementation of the classification based on linear *SVM* patterns. In this section and the following, the patterns using a linear *SVM* as occurrence mapping function will be called linear *SVM* patterns. We will now describe the implementation of the classification process using linear *SVM* pattern as features. The difference with the discriminant chronicle mining presented in Section 4.2 is that the kernel is trained in one single step contrary to the several "one against all" step for the discriminant chronicle. The relational dataset used to construct such classifier contains so potentially more than two labels. The occurrence mapping function of such pattern will then choose to which class belongs an occurrence of a multiset. For the discriminant chronicles, it was only an answer if this occurrence relies to the class for which this pattern was extracted. The difference corresponds then to the difference between the definition of the occurrence mapping function for the generalized chronicles and the occurrence mapping function for the discriminant generalized chronicles. This difference is mainly favorable to the linear *SVM* pattern in classification. Intuitively, this improvement of the classification accuracy can be related to the final classifier that has to manage fewer features and more qualitative features.

Each frequent multiset will then generate exactly one pattern. To reduce the number of extracted patterns and so the computation time, standard constraints on multisets can be used like minimal/maximal size, top-k frequent selection for each class or closure. A pattern could also be pruned from the pattern set if its training score *i.e.* accuracy on the train set is lower than a minimal threshold.

To merge the decision of the different patterns, a linear *SVM* is then trained on a validation set using the decision of all the patterns as features. The classification step is finally done on each

sequence of the test set using this decision tree on the vector of the pattern decisions. We try to use decision tree or naive bayes classifiers instead of the global linear *SVM* but they produced lower results.

Multiple instance management Similar to discriminant chronicle mining task, all classifiers lead to the same problem of multiple instances when a multiset occurs several times in a sequence. We have to manage this problem firstly in the training step and secondly in the labeling step.

For the training step, we consider all the occurrences of a multiset in a sequence as examples this multiset for the sequence label. This involves the strong assumption that all occurrences of a multiset influence on the sequence label. Some other assumption like the selection of a witness occurrence [Andrews et al., 2003] or one of those listed in [Foulds and Frank, 2010] could have been used. This assumption is potentially wrong in many cases but there is not better concrete alternative. It is a strong difference with the use of discriminant chronicles because we can learn wrong patterns. The validation set will, however, allow to prune those patterns if they are not relevant for classification. Even if this assumption could involve wrong decision for a label, the set of decision made from the whole pattern set can soften this effect.

For the labeling step, we want to train a final classifier using the evaluation of each pattern. We tried then 3 formats of prediction as features for the patterns: (i) a label format, (ii) a cardinality format and (iii) a normalized cardinality format.

Let $\mathscr{S}$ be a set of sequences, $\mathscr{L}$ be the set of labels in $\mathscr{S}$ and $\mathscr{P}$ a pattern set. For the label prediction, we want to predict the label of s for each $s \in \mathscr{S}$. Each pattern will then vote for the label that best fits the sequence. To do this, each sequence s is encoded as a feature vector $< l_1, ..., l_i, ..., l_n >$ where $l_i \in \mathscr{L}$ and l_i is produced by the decision of $p_i = (\mathscr{E}_i, \mathscr{T}_i) \in \mathscr{P}$ on the occurrence set of $\mathscr{E}_i$ in s. The label l_i decided by the pattern p_i is in this case the most common label predicted by p_i on each occurrence of $\mathscr{E}_i$ in s. For the cardinality format, the pattern produce a weighted vote for each label. As this pattern chooses a label for a sequence among several occurrences, the cardinality format allows to obtain the number of occurrences leading to the vote of a label. With this format, the multiple-instance assumption is not included in the occurrence mapping function of the pattern but transferred to the global classifier. The feature vector becomes of the form $< c_{1,1}, ..., c_{i,j}, ..., c_{n,m} >$ where $c_{i,j} \in \mathbb{N}^+$, $l_i \in \mathscr{L}$ and $p_j \in \mathscr{P}$ and $c_{i,j}$ is the number of occurrences of $\mathscr{E}_j$ in s for which p_j predicted l_i. Finally, the normalized cardinality prediction has the same form as the cardinality prediction but is a feature vector of $\widehat{c}_{i,j} \in \mathbb{R}^+$ where $\widehat{c}_{i,j} = \dfrac{c_{i,j}}{n_j}$ and where n_j is the number of occurrences of $\mathscr{E}_j$ in s. This format seems fairer if some sequences generate too different numbers of occurrences for a multiset than others.

5.1.3 Temporal pattern interpretability

To explain what is called interpretability in this chapter, we based our definitions on some of the definitions of the state of the art listed in [Lipton, 2016]. The interpretability of a pattern or a pattern set can have many different definitions. Lipton [Lipton, 2016] regroups then the different uses of the term "interpretability" in their different meaning. Thereby, we cannot use the interpretability as an axiom. In this section, we define what is the interpretability in our case and what interests us in interpretability.

5.1.3.1 Pattern set interpretability

We will first define what an interpretable pattern set is and why we want this pattern set to be interpretable. The definition that interests us concerning the pattern set is the **transparency** of an interpretable model. Transparency is introduced by Lipton [Lipton, 2016] as the opposite of

opacity or blackbox-ness. Transparency describes the ability of understanding the mechanism by which the model works. For a pattern sets, we will define transparency as the ability of understanding what knowledge is caught by it. Transparency is detailed in three subdefinitions: (i) **simulatability**, (ii) **decomposability** and (iii) **algorithmic transparency**.

Simulatability The simulatability of a model is the ability of this model to be contemplated at once. For pattern set, it is the ability to contemplate all the patterns at once. For pattern-based classification it includes the classifier parameters. The simulability of a pattern set relies mainly on the cardinality of the pattern set. The simulability of a pattern set also relies on the complexity of the pattern model. If the pattern model and the patterns are simple, the pattern set can be big and still simulatable. The evaluation of the simulatability of a model is subjective as it involves the cognition capacity of the human who has to understand the pattern set and so different humans can understand different types of models. Nonetheless, Lipton [Lipton, 2016] defines that this subjectivity can only span several orders of magnitude due to the limited capacity of human cognition.

Decomposability The decomposability of a model is that each part of the model admits an intuitive explanation. Lipton [Lipton, 2016] linked it to the intelligibility definition in [Lou et al., 2012]. A standard pattern set is naturally decomposable because each pattern is independent from the other. For pattern-based classification, it can be difficult to give an intuitive explanation to a parameter. The decomposability of a pattern set relies so more on the interpretability of the pattern model itself than on the ability to read a pattern without the rest of the pattern set. To improve the interpretability of each independent pattern in this decomposition, it can be useful to represent the pattern set using visual or text artifact. For example, we can cite the graphical representation of a chronicle as a graph.

Algorithmic transparency The last notion of transparency is the algorithmic transparency. Two examples of algorithmic transparency are given in [Lipton, 2016]: linear models and deep neural networks. The first is considered transparent because we can prove its convergence and understand the shape of the error surface. The second lacks of algorithmic transparency because we cannot guarantee that they will work on new problems. For pattern-based classification it will mainly depend on the algorithm used to construct the classifier.

5.1.3.2 Pattern model interpretability

To obtain an interpretable pattern set, each pattern has to be interpretable. To obtain such patterns, we have to choose an interpretable pattern model and so define what is an interpretable pattern model. In our case, we use the pattern mining task as an automatic hypothesis generation. As we want those hypotheses to be meaningful for the user, the patterns have to be interpretable.

Among the Lipton characteristics, I choose two characteristics that I want to find in an interpretable pattern and so in an interpretable pattern model: (i) **informativeness** and (ii) **causality**.

Informativeness The informativeness is closely linked to the unsupervised learning task. The unsupervised learning task relies on the exploration of the data, an algorithm explores data and produces a model containing information. Even if the task of discriminant temporal pattern mining is at first sight a subdomain of supervised learning tasks, we still need informativeness to use temporal patterns in human decision. The informativeness of a model is needed in a weak supervision context. In this case, the supervision is used as context to bound the exploration of the data but does not define precisely what has to be learned. The supervised learning task is then

used in context closer to the unsupervised learning where the goal is to explore the specificity of a dataset that does not occur in a more general dataset. An informative pattern is then potentially an interpretable hypothesis.

Causality The causality is the purpose of an interpretable pattern seen as a hypothesis. It is not a property we want to observe in a pattern model that we define interpretable. But this is a property that we want to test for a result pattern. The standard learning task and so the standard pattern mining task make associations and learn the correlations occurring in the data. To effectively use the knowledge represented by a pattern in a human decision, the causality of the pattern as to be tested to validate the hypothesis relying in it.

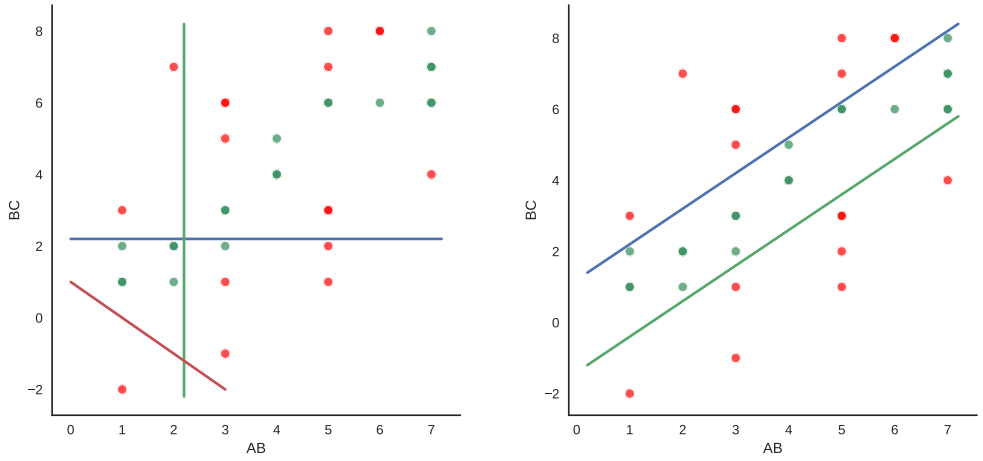


Figure 5.3: Graphical representation of the classification of several occurrences of a multiset $\{\!\{A,B,C\}\!\}$. AB corresponds to the temporal duration between the event A and B and BC is the same for B and C. The graph on the left represents a classification made using temporal constraints like those used in chronicles. The blue one corresponds to $BC \leq 2$, the green one to $AB \leq 2$ and the red one to $AC \geq 1$. The graph on the right represents a classification made using linear constraints like for a linear classifier. The blue one corresponds to $BC \leq AB + 1$ and the green one to $BC \geq AB - 1$.

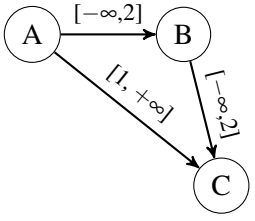


Figure 5.4: Graphical representation of the chronicle corresponding to the temporal constraints used on the left plot of the figure 5.3.

5.1.3.3 Trade-off between interpretability and accuracy

Constraining the pattern models and the extracted pattern sets to be interpretable as a cost. Indeed, some models are designed to catch complex knowledge but are difficultly interpretable. Not using those models can imply to miss those knowledge and so produce a lass accurate classifier. In

this section, we will use the example of the chronicles. The temporal constraints of chronicles are formally constrained to be conjunctions of inequations between a single event interval and an integer. Some complex knowledge like proportionality between pairs of interval cannot be caught by the chronicle temporal constraints but could be caught by more complex model like pairs of multiset and linear constraint set.

For example, we have the representation of two discriminant temporal patterns on the figure 5.3. Several occurrences of the multiset $\{\!\{A,B,C\}\!\}$ are distributed between two classes, the positive one (green points) and the negative one (red points). The figure on the left corresponds to the chronicle represented by the figure 5.4. We can see that this chronicle discriminates perfectly a subset of the green occurrences of $\{\!\{A,B,C\}\!\}$ but the knowledge contained in this chronicle is more specific than the knowledge represented by the second pattern. Because of this lack of generalization, the chronicle set have to represent each subgroup of positive occurrences to approach the general knowledge. Here, using chronicle as an interpretable pattern model implies to obtain a bigger pattern set and so a less simulatable pattern set due to a less accurate model.

On the other hand, we have on the figure on the right a temporal pattern using more complex linear constraints. The pattern defined by those constraints is: $(\mathscr{E} = \{\!\{A,B,C\}\!\},\ \mathscr{T} = \{BC \geq AB-1, BC \leq AB+1\})$. This pattern is much more interesting because it groups all the positive occurrences and none of the negative. This pattern represents the knowledge useful to discriminate the occurrences of the two classes but it is more difficult to represent and, if it was extracted with a black box algorithm, more difficult to read.

There is a trade-off between, the interpretability of the pattern model, the interpretability of the pattern set and the accuracy of the patterns. In this example we have a trade-off in accuracy between the interpretability of the pattern model and the accuracy. But the loss of accuracy implied by using the chronicle model in this example can be compensated if we extract a chronicle set covering the same occurrence set than the pattern with linear constraints. This implies another trade-off between the accuracy and the interpretability of the pattern set. In this case we can improve the accuracy by reducing the simulatability of the pattern set. Or improve the accuracy by using a less decomposable model.

5.2 Experiments

In this part, we compare different results obtained in pattern-based classification using discriminant partially ordered patterns, discriminant chronicles or using patterns with linear *SVM* as occurrence mapping function. The choice of a linear *SVM* mainly relies on the computational time required to build such classifier. Indeed a more complex classifier based on kernels like rbf or *polynomial* kernels was prohibitive for the experiments. This comparison is done to highlight the different notions of the interpretability and the trade-off between those notions and the accuracy that could be implied by a pattern model used in temporal pattern-based classification.

5.2.1 Experimental setup

The goal of those experiments is to evaluate the trade-off between the discriminant chronicle model and a model using more complex temporal constraints in terms of accuracy. To do this, we have to compare the different ways to use linear *SVM* patterns in pattern-based classification to obtain the best results in terms of accuracy. Then we could compare those results with the previous results obtained for classification based on chronicles. We classify two types of datasets: (i) two synthetic datasets designed to linear constraints and (ii) the datasets for which we obtained previous results for the discriminant chronicle model (section 4.5).

For the two types of experiments, several parameters were used to set up the classification based on linear *SVM* patterns:

- Minimal support: Like for the discriminant chronicle mining step, the linear *SVM* patterns have to be extracted using a minimal support parameter. This parameter will be used to prevent from learning linear *SVM* for infrequent multiset *i.e.* on too few examples. By default, the same minimal support as for chronicles will be used.

- Minimal and maximal size of multiset: Those parameters are used to constrain the set of linear *SVM* patterns to be small enough. The size of the multiset $\mathscr{E}$ is defined as its cardinality $|\mathscr{E}|$. By default, the minimal size is set to 2. The goal of this setting is to ignore multisets containing only one event and so multisets that could not be temporally constrained except on the absolute timestamp of this single event. The maximal size parameter is more efficient to limit the computational complexity of the approach. With a maximal size parameter set up to 4 or 5, we can expect to catch sufficiently complex patterns to represent all the behaviors occurring in the datasets while avoiding training linear *SVM* on too big relational datasets.

- The *top-k* frequent parameter: Another option based on the multiset set to limit the size of the pattern set is to prune the multiset set with a *top-k* constraint. Only the k most frequent multisets are used to generate linear *SVM* patterns. When this constraint is used, the default k parameter is set to 90 as for *BIDE-D* approaches [Fradkin and Mörchen, 2015].

- Condensed representation of the multiset set: To avoid redundancy in the pattern set, the multiset set can be extracted in a condensed representation. To do this, we use the parameters provided by the Eclat implementation in PyFIM [Borgelt, 2003]: (i) all the patterns, (ii) the set of closed patterns [Pasquier et al., 1999], (iii) the set of free patterns and (iv) the set of maximal patterns [Pasquier et al., 1999]. The most used condensed multiset sets are the closed and the free multisets. The intuition behind using the closed patterns is that it is useless to extract patterns based on two similar multisets a and b if $a \subseteq b$ and $supp(a) = supp(b)$ because the whole information of a is in b. The intuition behind using the free patterns is the same as for the closed but the conclusion is the opposite. If $supp(a) = supp(b)$ and $a \subseteq b$, we prefer to keep a because it contains the core knowledge and potentially not the frequent noise.

- C parameter for each linear *SVM* in a pattern: The C parameter is a standard parameter for a *SVM* classifier. This parameter is generally set up with a power of 10 and the default setup is 1. The lower the parameter is, the more the classifier will penalize errors in the model. One drawback of a low value for C is to overfit the training data and learn model too different from the reality. On the other side, a bigger C will produce a less accurate classifier but with a lower risk of overfit. In general, too low or high C parameters result in a huge computational time.

- global *SVM* kernel: Contrary to the *SVM* classifier of each pattern, the use of another kernel for the global *SVM* does not increase the computational time. Thereby, it may be interesting to evaluate kernels like rbf or *polynomial* instead of simply using a linear *SVM*.

- C parameter for the global *SVM*: Like for the *SVM* classifier in each pattern, the global *SVM* can be set up using different C.

- Features format: As detailed in Section 5.1.2.1, there is not one objective features format produced by each patterns due to the multiple instance problem. We tested then the three

Algorithme 3 Interval sequence generation

Require: x_b, x_e: bounds of the random interval for x, t_b, t_e: bounds of the random interval for t_{B_b}, t_{A_b}: beginning timestamp of A, k: proportionality parameter, n: number of sequences to generate

1: $\mathscr{S} \leftarrow \emptyset$ ▷ $\mathscr{S}$ is the sequence set
2: **for** $i = 1$ to n **do**
3: x a random number such that $x \in [x_b, x_e]$
4: t_{B_b} a random number such that $t_{B_b} \in [t_b, t_e]$
5: $\mathscr{S} \leftarrow \mathscr{S} \cup \{\{(A_b, t_{A_b}), (A_e, t_{A_b} + x), (B_b, t_{B_b}), (B_e, \lfloor t_{B_b} + kx \rfloor)\}\}$ ▷ Generate a new interval sequence as a set of events.
6: **return** $\mathscr{S}$

formats presented in Section 5.1.2.1 as parameters of the classification using linear *SVM* patterns.

5.2.1.1 Synthetic data

The goal of experiments on synthetic data is to prove that it exists dataset, at least synthetic, on which classifiers based on linear *SVM* patterns are more accurate than classifiers based on discriminant chronicles. In fact, the goal is to experimentally test the example described by Figure 5.3. As described in Section 5.1.3.3, chronicle fail to find the underlying behavior where linear *SVM* can.

The template used to generate datasets containing proportional behaviors is as follows:

- Each sequence contains two intervals: A and B.
- The two intervals are represented in the data by their beginning and ending events: A_b, A_e, B_b and B_e.
- Two sequence classes are distinguished by the way interval are randomly generated: more precisely we control the proportionality between the duration of A and the duration and B *i.e.* $t_{A_e} - t_{A_b} = k_C(t_{B_e} - t_{B_b})$ where k_C depends on class C.

In practice, to generate different datasets we have to determine some parameters. For the first dataset, we choose to generate two classes as it is illustrated in figure 5.5. For this dataset, we have generated 100 sequences for each class. The sequence set of the first class $\mathscr{S}^+$ was generated using algorithm 3 using the parameter set: $x_b = 1$, $x_e = 9$, $t_b = 1$, $t_e = 29$, $t_{A_b} = 15$, $k = 2$ and $n = 100$. The sequence set of the second class $\mathscr{S}^-$ was generated using the parameter set: $x_b = 1$, $x_e = 9$, $t_b = 1$, $t_e = 29$, $t_{A_b} = 15$, $k = 1$ and $n = 100$. The length of interval A noted x is an integer randomly chosen in the interval $[1,9]$. The beginning event A_b occurs in each sequence at the timestamp 15. The beginning event B_b timestamp is an integer randomly chosen in the interval $[1,29]$. This noise generation is done to reduce the discriminant power of behaviors that are not the proportionality. In the positive dataset, the length of B is equal to the length of A. In the negative dataset, the length of B is twice the length of A. The behaviors discriminating the whole positive set and the whole negative set are then $2(t_{A_e} - t_{A_b}) = t_{B_e} - t_{B_b}$ and $t_{A_e} - t_{A_b} = t_{B_e} - t_{B_b}$ respectively.

For the second dataset, we choose to generate three classes as it is illustrated in figure 5.6. We still have the positive and negative classes but we added to this dataset the *other* class. For this dataset, we generated 1000 sequences for each class. The parameters for the random intervals are unchanged and A_b still occurs at the timestamp 15. To generate this dataset, k was set to $\frac{1}{2}$, 1 and $\frac{3}{2}$ in *other*, negative and positive classes respectively.

The difference of complexity between those two datasets will allow us to evaluate the robustness of discriminant chronicles for such dataset type.

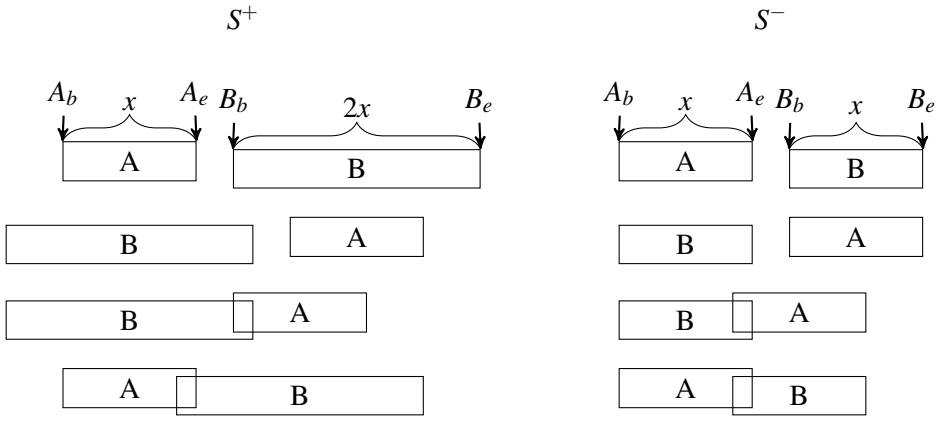


Figure 5.5: Interval representation of the first synthetic dataset. On the left, the positive dataset contains sequences for which interval B is 2 times bigger as interval A. On the right, the negative dataset contains sequences for which interval B length is equal to interval A length.

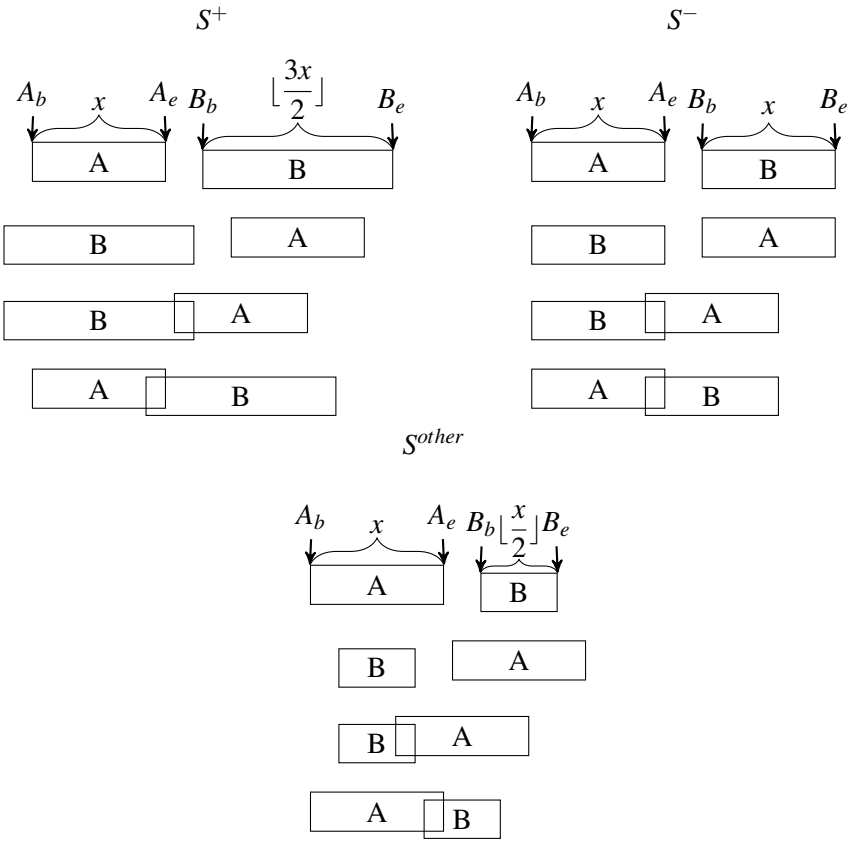


Figure 5.6: Interval representation of the second synthetic dataset. On the top left, the positive dataset contains sequences for which interval B is 1.5 times bigger as interval A. On the top right, the negative dataset contains sequences for which interval B length is equal to interval A length. On the bottom, the dataset of the class *other* contains sequences for which interval A is 2 times bigger than interval B.

5.2.2 Results

This section presents and discusses the results obtain by the experiments previously detailed.

5.2.2.1 Synthetic data

In this section, we will present the classification accuracy measures for three pattern models on the synthetic datasets. Those three pattern models are discriminant partially ordered patterns, discriminant chronicles and linear *SVM* patterns. We present the accuracy results of those three pattern models in ascending order. Those accuracy results confirm the intuition that a sequential pattern model like partially ordered patterns produce the worst accuracy compared to two temporal pattern model. They also confirm that the linear *SVM* model produces the best accuracy that is almost perfect. Finally, those results show the trade-off between the interpretability of the model, the interpretability of the pattern set and the accuracy. The limit of this trade-off is also shown for the too simple model that is the partially ordered patterns on this type of dataset.

Partially ordered patterns The best accuracy for discriminant partially ordered patterns on the first synthetic dataset was 0.6606 with a standard deviation of 0.0744 for the parameters minimal support $\sigma_{min} = 0.2$, minimal growth rate $g_{min} = 2$ and a linear *SVM* to use the partially ordered patterns as features. Those accuracy results are low. However, those results were expected for such type of patterns on a dataset dedicated to linear temporal constraints. As a partially ordered pattern contains only sequential information, it is not able to describe different sizes of interval. Even if those partially ordered patterns are not able to catch the real behavior introduced in the dataset, the extracted partially ordered patterns have caught behaviors indirectly introduced by the random generation parameters. The extracted partially ordered pattern $\mathscr{P}_1 = (\{\!\{A_b, A_e, B_b, B_e\}\!\}, \{t_{B_b} < t_{A_b}, t_{A_b} < t_{B_e}\})$ is graphically represented on the top left of Figure 5.7. This partially ordered pattern represent the higher probability that B begins before A but ends after the begins of A in the positive sequences than in the negative sequences. This is due to the setup of the generation that makes B twice longer in the positive sequences than in the negative sequences. This partially ordered pattern is interesting because it sums up three different sequential patterns. They are graphically illustrated on Figure 5.7 by the subfigures (i), (ii) and (iii). Those patterns describe 3 interval relations between A and B:

- B overlaps A: This relation is represented by figure 5.7(i). This case exists only in the positive sequences because B has to be longer than A. To obtain this behavior in a sequence, t_{B_b} has to be lower than $t_{A_b} = 15$. As t_{B_b} is randomly chosen in the interval $[1,29]$, $P(t_{B_b} < t_{A_b}) = \frac{14}{29}$. Then, $t_{B_b} + 2x$ has to be bigger than $t_{A_b} + x$ *i.e.* $t_{A_b} - t_{B_b} < x$. As x is randomly chosen in the interval $[1,9]$, x must be greater than 1 for $t_{B_b} = 14$, greater than 2 for $t_{B_b} = 13$, etc. We have then 36 pairs of randomly chosen x and t_{B_b} that generates this behavior. As we have 29 possible values for t_{B_b} and 9 for x, the probability of B overlapping A is $\frac{36}{261} \simeq 13.79\%$. The pattern representing B overlapping A is a prefect discriminant pattern, but its support is normally lower than the minimal support of 20% used as parameters and so it could not be extracted as it. The pattern $\mathscr{P}_1$ is then a less discriminant generalization of B overlap A that is sufficiently frequent to be extracted.

- B begins before A and they end at same time: This relation is represented by figure 5.7(ii). Similar to the previous case, B has to be longer than A and so this case exists only in sequences of class $+$. In this case we still have the constraint that $t_{B_b} < t_{A_b}$. In this case we do not have the constraint that $t_{A_b} - t_{B_b}$ is lower than x but that $t_{A_b} - t_{B_b}$ is equal to x.

Thereby only 9 combinations are possible to obtain $t_{B_b} + x = t_{A_b}$ based on the 9 possible values for x. The probability that "B begins before A and that they end at same time" is then $\frac{9}{261} \simeq 3.45\%$ far below the minimal support. The disjunction between cases (i) and (ii) and so another prefect discriminant pattern but occurring only in $\frac{45}{261} \simeq 17,24\%$ of the positive dataset. This pattern can be represented by our model of partially ordered patterns with adding the constraint $t_{B_e} \geq t_{A_e}$ to the temporal constraint set of $\mathscr{P}_1$.

- B begins before A and ends during A: This relation is represented by the subfigure (iii) of 5.7. This case is definitely less discriminant than the two others because it could occurs in positive and negative sequences. In this case we still have the constraint that t_{B_b} is lower than t_{A_b}. We will now compute the probability of this behavior in the positive sequences. To have B ending after that A begins, we need that $t_{A_b} - t_{B_b}$ be lower than $2x$. To have B ending before that A ends, we need that $t_{A_b} - t_{B_b}$ be greater than x. We have then the constraint $x < t_{A_b} - t_{B_b} < 2x$ and so $x + t_{B_b} < t_{A_b}$ and $2x + t_{B_b} < t_{A_b}$. By enumeration, we obtain 32 combinations of x and t_{B_b} satisfying those constraints and then a probability of $\frac{32}{261} \simeq 12.26\%$ to generate this behavior in a positive sequence. We will now compute the probability of this behavior in the negative sequences. In addition to the constraint $t_{B_b} < t_{A_b}$, we need that $t_{A_b} - t_{B_b} < x$. This was previously computed for the case (i) and so, the probability to generate this behavior in a negative sequence is $\frac{36}{261} \simeq 13.79\%$. This last case is then mostly not discriminant and as more chance to occur in negative than in positive sequences but its frequency allows $\mathscr{P}_1$ to be sufficiently frequent with a probability of occurring in a positive sequence equals to $\frac{77}{261} \simeq 29.50\%$ and a probability of occurring in a negative sequence equals to $\frac{36}{261} \simeq 13.79\%$. The pattern $\mathscr{P}_1$ is not perfectly discriminant but it is sufficiently frequently generated according to a minimal support of 20% and sufficiently discriminant for a minimal growth rate of 2.

This is an example of the trade-off between the interpretability of the model, the interpretability of the pattern set and its accuracy. In this case we choose to discriminate such dataset with patterns that are easily readable and understandable and can be considered interpretable at a first glance. The pattern set of this pattern model seems to be interpretable because it is easily decomposable. This pattern model is not able to catch the real discriminant behavior introduced in the dataset and several induced behavior as to be extracted to obtain a reasonable accuracy if we use them for classification. If we extract patterns with a minimal support lower than 20% we could generate the pattern catching the subbehaviors (i) and (ii) of $\mathscr{P}_1$ and so maybe improve the accuracy thanks to more discriminant patterns. On the other hand, if we use a lower minimal support, we will extract more patterns and so generate a bigger pattern set. If the pattern set grows too much, the simulatability of the pattern set will significantly decrease and so the pattern set will be less interpretable. Choosing those parameters and so patterns like $\mathscr{P}_1$ allows then to produce few patterns easily interpretable but that roughly discriminate the dataset classes.

Chronicles The best accuracy for discriminant chronicles on the first synthetic dataset was 0.8981 with a standard deviation of 0.0432 for the parameters minimal support $\sigma_{min} = 0.2$, minimal growth rate $g_{min} = 2$ and a linear SVM to use the chronicles as features. The accuracy for the same parameters but using chronicles as rules is 0.8538 with a standard deviation of 0.0669 using the precision measure Q_g with a bias parameter $g = 1$ to sort them. With $g = 0$ the accuracy is 0.8481 with a standard deviation of 0.0608. This difference seems to be not significant. Without

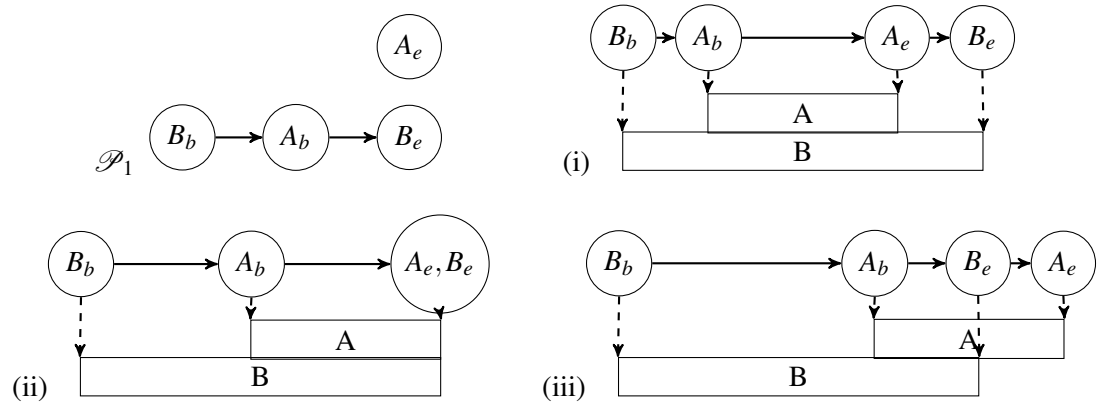


Figure 5.7: The discriminant partially ordered pattern $\mathscr{P}_1$ extracted for a classification run on the first synthetic dataset and the sequential cases (i), (ii) and (iii) that it represents. This partially ordered pattern was extracted for a support of 24 in the positive sequences and 7 in the negative sequences. This represents 30% of positive sequences and 8.75% of negative sequences of the train set.

the strategy to predict one of the most represented classes, here the positive one, this accuracy fell to 0.7306 with a standard deviation of 0.0865. The proportion of misclassified sequences is then $1-0.7306=0.2694$ and so the accuracy without the strategy plus half of the misclassified sequences 0.8653 corresponds to the accuracy with the strategy. The strategy for predicting the label of a sequence not covered by the chronicles could then to randomly choose the sequence label for the sequences not covered by discriminant chronicles. Using or not a minimal size constraint set up to 4 for the extraction of the discriminant chronicles did not impact significantly the accuracy results, the following examples of extracted patterns will so contain the 4 events A_b, A_e, B_b and B_e.

As expected results in terms of accuracy are much better for the chronicle model than for the partially ordered pattern model. It can be simply explained through the dataset generation using numerical parameter for the temporal information. Those parameters can be more precisely caught by a model like the chronicle model than the partially ordered pattern model because the second one does not represent the temporal information with numerical values.

Chronicles extracted for one of the classification runs are illustrated by Figure 5.8. The chronicles $\mathscr{C}_1$ and $\mathscr{C}_4$ were extracted for all runs. The chronicle $\mathscr{C}_2$ was extracted for all but one run. The chronicle $\mathscr{C}_3$ was extracted for several runs or was replaced by similar chronicles. For two runs, only the chronicles $\mathscr{C}_1$, $\mathscr{C}_2$ and $\mathscr{C}_4$ were extracted. Each of those four chronicles describes a part of the proportionality of the generation for specific subset of the search space.

- The most general chronicle is $\mathscr{C}_1$. The constraint "$B_b \rightarrow B_e$" ≥ 10 is based on the fact that x is an integer randomly chosen in the interval $[1,10[$. This chronicle is more general than the other because it will always be discriminant for datasets for which the length A is equal to the length B in the negative sequences and B is bigger than A in the positive sequences. The constraint "$B_b \rightarrow B_e$" ≥ 10 could be generalized as "$B_b \rightarrow B_e$" $\geq b$ for x randomly chosen in the interval $[a,b[$. However, even if this chronicle is discriminant for those cases, the high frequency of the chronicle $\mathscr{C}_1$ is implied by the fact that B is twice longer than A.

- The chronicles $\mathscr{C}_2$ and $\mathscr{C}_4$ discriminate both subsets of the search space where "$B_b \rightarrow B_e$" $<$ 10 *i.e.* subsets not covered by $\mathscr{C}_1$. The chronicle $\mathscr{C}_2$ is built like the chronicle $\mathscr{C}_1$ but specify the maximal length of A *i.e.* the maximal x that it covers. The first constraint "$A_b \rightarrow A_e$" ≤ 4 defines the cases that are not covered by $\mathscr{C}_1$ because for "$A_b \rightarrow A_e$" > 4 *i.e.* $x > 4$, "$B_b \rightarrow B_e$" ≥ 10. The second constraint "$B_b \rightarrow B_e$" ≥ 6 works like "$B_b \rightarrow B_e$" ≤ 10. For all the

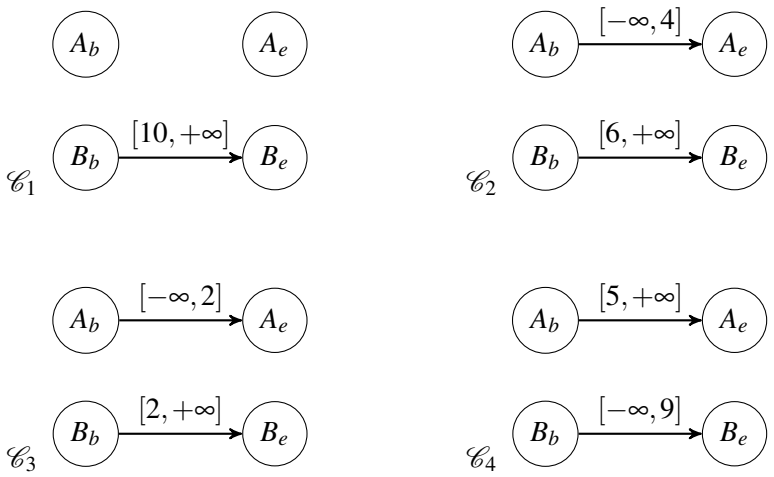


Figure 5.8: The discriminant chronicles extracted for a classification run on the first synthetic dataset. The chronicles $\mathscr{C}_1$, $\mathscr{C}_2$ and $\mathscr{C}_3$ discriminant the positive sequences and the $\mathscr{C}_4$ discriminate the negative sequences. The chronicles $\mathscr{C}_1$, $\mathscr{C}_2$ and $\mathscr{C}_4$ are perfectly discriminant and were extracted for a support of 40, 20 and 45 respectively. The chronicle $\mathscr{C}_3$ was extracted for a support of 20 in positive sequences and 9 in negative sequences.

negative examples for which $x \leq 4$, we are sure that the length of the interval B is lower than 6 because of the length of B equals x too. The chronicle $\mathscr{C}_4$ is constructed like $\mathscr{C}_2$ as it covers cases not covered by $\mathscr{C}_1$ but is the opposite of $\mathscr{C}_2$ because it discriminates negative sequences and not positive sequences. As the length of B is $2x$ in positive, the chronicle $\mathscr{C}_4$ describe that if $x \geq 5$ and the length of B is lower than 10 then it is a negative sequence.

- The chronicle $\mathscr{C}_3$ tries to cover a subset of the sequences not covered by $\mathscr{C}_1$, $\mathscr{C}_2$ and $\mathscr{C}_4$ *i.e.* those where $x \leq 4$ and the length of B is lower than 6. Like for the discriminant partially ordered pattern $\mathscr{P}_1$, $\mathscr{C}_3$ could be divided in subcases perfectly discriminant and more generally, we could describe each possible value of x by a discriminant chronicle for the positive sequences and another for the negative sequences but those chronicles will have a probability to occur in a sequence of their class of $1/9 \simeq 11.11\%$ that is below the minimal support constraint set up to 20%.

As the chronicle model is not able to catch the proportional behavior used to generate this dataset, we observe another trade-off between the interpretability of the pattern set and the accuracy. By securing the simulatability of the pattern set by constraining the chronicles to occur at least in 20% of the sequences of the class that they discriminate, the accuracy is not perfect. It is worth mentioning that it is possible to discriminate perfectly the dataset with 9 chronicles describing the 9 possible values for x. If those chronicles discriminate the positive sequences, the classifier will have to label a sequence s as positive if one of those chronicles occurs in s and as negative if not.

The accuracy of the discriminant chronicles used as features in a linear *SVM* is lower for the second dataset than for the first one but still good. For the same parameter set as for the first dataset, the discriminant chronicle model obtained the accuracy of 0.7861 with a standard deviation of 0.0214. The discriminant chronicles used as rules obtained the accuracy of 0.7555 with a standard deviation of 0.0473. Without the strategy to predict one of the most represented classes, here the positive one, this accuracy fell to 0.6222 with a standard deviation of 0.0321. Like for the first dataset, a non-negligible part of the accuracy is obtained by the classifier and not

the chronicle themselves. The gain of accuracy obtained by this strategy, here 0.1333 is near to be a third of the not good classified sequences. It suggests that the not good classified sequences are mostly uniformly distributed on the sequences. Constraining the size of the extracted chronicles to be greater or equal to 4 has still no significant impact on the accuracy. The chronicle $\mathscr{C}_1$ of figure 5.8 is still extracted for this dataset. The other extracted chronicles discriminate subset of the search space mostly like $\mathscr{C}_2$ and $\mathscr{C}_4$ and some represent the trade-off between the frequency and the growth rate like $\mathscr{C}_3$. The mean number of extracted chronicles is 6 for a minimal support of 20%.

The discriminant chronicle model is an understandable model that can obtain a good accuracy even on dataset containing behaviors that cannot be caught directly by chronicles. This good accuracy implies an impact on the simulatability of the pattern set. For the first dataset, 4 chronicles were used to classify the dataset were 1 linear constraints could perfectly discriminate the sequences. For the second dataset, 6 chronicles were used to represent the behavior of 2 linear constraints. We could then generalize the notion of informativeness of a pattern to the informativeness of a pattern set. Are the extracted chronicle set informative enough to deliver to the user the proportional behavior used to generate the dataset?

linear *SVM* patterns As expected, using linear *SVM* with default classification parameters produce a perfect classifier for the first synthetic dataset. Using a parameter of minimal size set up to 4 did not change this result but generate a unique discriminant pattern. Thereby, the interpretability of the pattern set is reduced to the interpretability of a pattern while the accuracy is the best. For the second dataset the linear *SVM* classifier is not perfect even if we modify the parameter *C* of the *SVM*. Some sequences are misclassified and this leads to a mean accuracy of 0.9551 with a standard deviation of 0.0089. Like for the first dataset, using a minimal size constraint set up to 4 did not impact significantly the accuracy.

We then obtain one unique pattern for each dataset that can be used as or in a classifier to perfectly label the first dataset or to label with a very good accuracy the second one. By choosing a model that we consider less interpretable than the two others, we maximize the interpretability of the pattern set and the accuracy. Like for the generalization of the notion of informativeness to a pattern set, we might want to compare the interpretability of a pattern set and the interpretability of a pattern if this pattern is the only one extracted. To be more specific, we might want to compare the interpretability of a linear *SVM* pattern with the interpretability of a chronicle set if they both respond to the same question like in this example.

5.2.2.2 BIDE-D datasets

In this section, we present the results of classification using linear *SVM* patterns as features on the *BIDE-D* datasets. The classification is done for several parameter sets. The results are then compared with the previous results obtained by classification using discriminant chronicles as features or as rules. Finally, we discuss those results and the differences implied by each parameter set to highlight the advantages and disadvantages of both approaches.

The two figures 5.10 and 5.9 show the results of the experiments in classification on the dataset *asl-bu* using linear *SVM* patterns as features. Figure 5.10 is dedicated to the experiments using $k = 90$ for the topk constraint applied on the multisets and the figure 5.9 is dedicated to the experiments not using this constraint. Table 5.3 shows the 10 best parameter sets sorted by accuracy.

Firstly, we can see on table 5.3 that the best parameter set is: minimal support = 0.2, top k = 0, max size = 3, eclat target = *c* and global *SVM C* = 0.1 for an accuracy of 0.7298 with a standard deviation of 0.0542. As we were not able to compute the accuracy of a classifier using discriminant chronicles as features on $asl - bu$ for a minimal support of 0.2, we will compare the

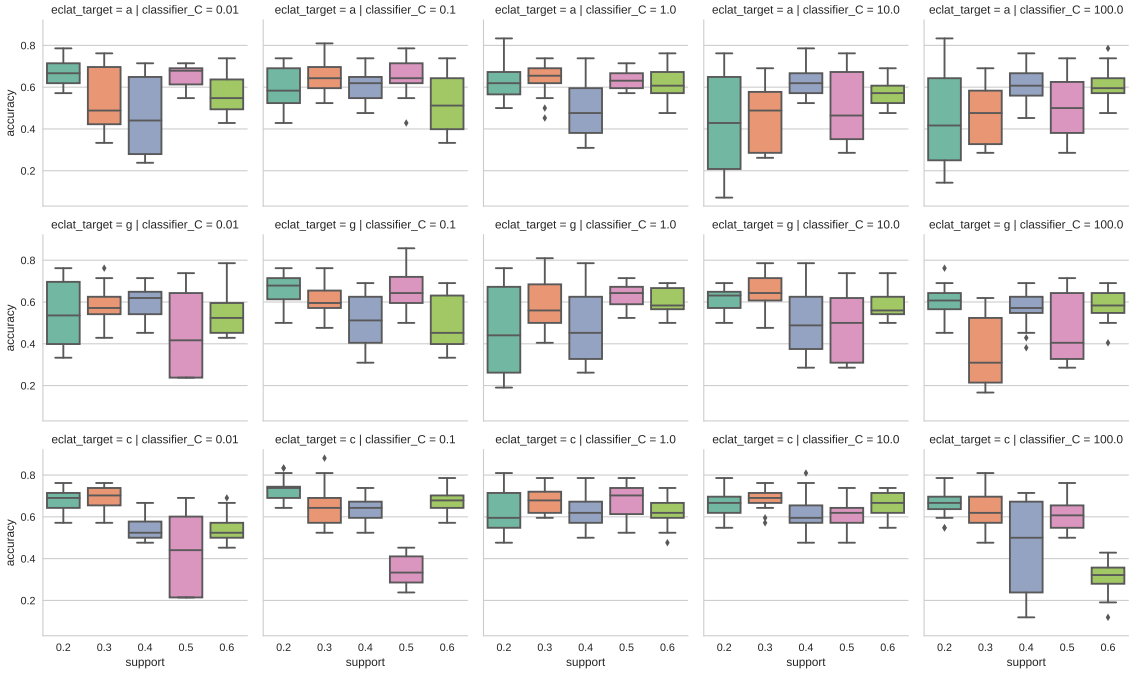


Figure 5.9: Accuracy boxplot for temporal patterns using linear *SVM* as temporal constraints used as features in a linear *SVM* on the *asl-bu* dataset for different minimal support. Each boxplot represents 20 runs for those parameters obtained by two 10-cross validation. Different facets are generated for the use of closed multisets (eclat_target=c), generator multisets (eclat_target=g) or all frequent multisets (eclat_target=a) and different values for the C parameter of the classifier using the patterns as features A topk constraint was used to limit the number of patterns with $k = 90$.

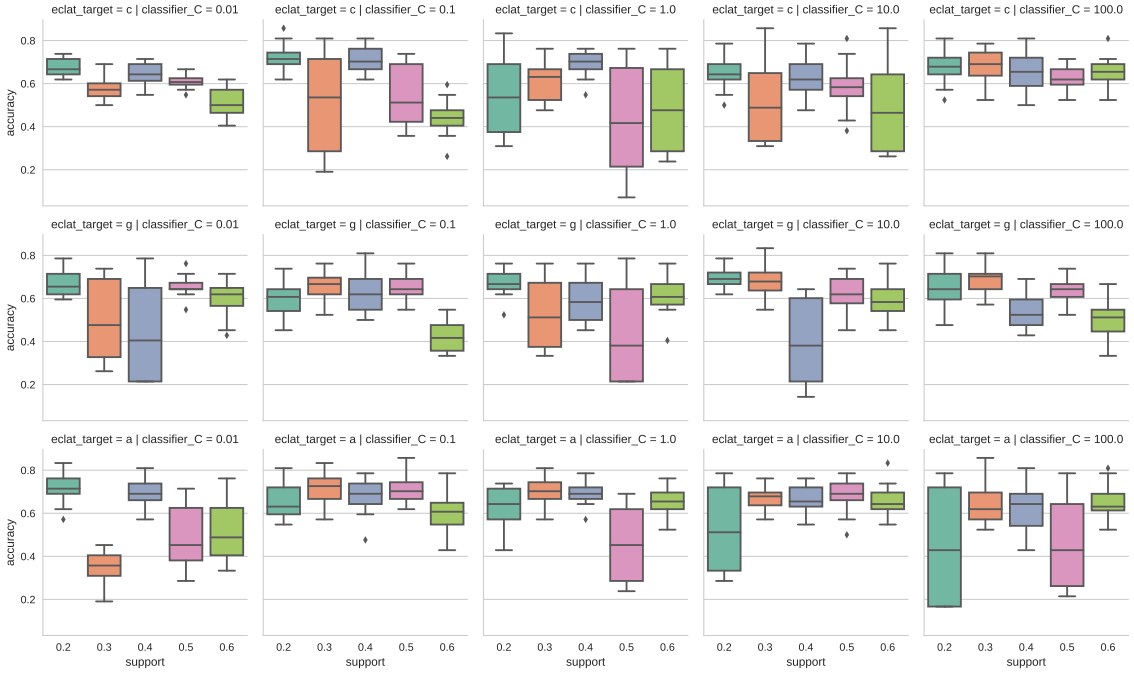


Figure 5.10: Same experiment that for the figure 5.9 without the topk constraint.

support	topk	max_size	eclat_target	classifier_C	accuracy mean	std
0.2	90	3	c	0.1	0.7298	0.0542
0.3		4			0.7286	0.0548
0.2	0	3			0.7250	0.0560
0.3		5			0.7250	0.0702
0.2	90	4		1	0.7238	0.0730
				0.01	0.7202	0.0463
0.3	0			0.1	0.7179	0.0652
0.2		3	a	0.01	0.7179	0.0590
0.3				0.1	0.7167	0.0646
0.2		5	c	100	0.7143	0.0921

Table 5.3: The top 10 parameter sets in terms of accuracy (mean and standard deviation) on the *asl-bu* dataset.

results of linear *SVM* patterns and discriminant chronicles for a minimal support equals to 0.3. For a minimal support equals to 0.3, the best accuracy produced by the linear *SVM* patterns is 0.7286 with a standard deviation of 0.0548. The other parameters associated to this result are the same that for the previous result except that the max size is set to 4 instead of 3. Those results are then much better than the best result of discriminant chronicles that was 0.6736 for a standard deviation of 0.0555. We can notice that the best result of discriminant chronicles is obtained for a minimal support of 0.6. For this support, the best accuracy of the linear *SVM* patterns is 0.6762 with a standard deviation of 0.0634. The results of the discriminant chronicles are not significantly better than this result but we can see that all results of discriminant chronicles are greater than this result. Thereby, the classification using linear *SVM* patterns as features does not produce a significant better result than the classification using discriminant chronicles as features for each minimal support.

We can observe on figures 5.9 and 5.10 that the accuracy results are not always proportional with the minimal support. This behavior could be observed for discriminant chronicles but not in such proportions. For example, on Figure 5.10 we can see that for the parameters eclat target = c and global *SVM* $C = 0.1$, the accuracy results obtained for a minimal support equals to 0.2 or 0.4 are greater than 0.6 but for a minimal support equals to 0.3 lower than 0.4. This seems to be non-intuitive because we expect to obtain a better accuracy or at least not a lower accuracy by adding features to the classifier.

Another observation that we can do on those plots is the influence of the minimal support on the accuracy. The naive intuition is that more the minimal support is low, more the accuracy is high, or at least, not lower than for a higher minimal support. The reason behind this intuition is that a lower minimal support will produce more patterns and so more knowledge. With this new knowledge we can then hope to obtain a good accuracy for more sequences. At least, we can hope to obtain a similar accuracy because the pattern set obtained for a higher minimal support is contained in the new pattern set. The necessary knowledge to obtain the same accuracy as a higher minimal support is also extracted for a lower minimal support and should lead at least to this accuracy. But this intuition is not always validated by figures 5.9 and 5.10.

We can for example see a stable accuracy on Figure 5.9 for parameters eclat target = c and *SVM* $C = 1$. The accuracy is not better on this plot for the lowest minimal support threshold but they are not significantly lower either. We can then think that the same knowledge was used to

obtain the same accuracy. But for the same eclat target, we can observe that for $SVM\ C = 0.1$, there is a great difference in term of accuracy between a minimal support of 0.6 and a minimal support of 0.5. In the first case we obtain the accuracy of 0.6726 with a standard deviation of 0.0583 and in the second the accuracy of 0.3452 with a standard deviation of 0.0743. We can notice that the accuracy for the next minimal support is equal to 0.6393 with a standard deviation of 0.0605. There is then one configuration that produce bad results but close to other good configurations. This is not a problem in an optimization context. We just have to use cross-validation to test all the parameter sets and select the most optimal. But this is a problem in terms of interpretability of the model. How can we interpret this difference of accuracy obtained by different values for parameter C. We can obtain a difference greater than 0.3 with only changing a parameter of the global classifier. But we cannot link only this value to a bad accuracy because it is included in the parameter set used to obtain the best accuracy. This relies then to a bad simulatability of the global classification process because it is difficult to understand why a parameter set will produce a good accuracy, and a similar other one a bad accuracy.

We observe the same behavior comparing the two figures 5.9 and 5.10. Even if the best results are obtained with the top k constraint set to 90, we can see on those figures than it is difficult to choose if it is good for the accuracy to use it. Using or not this constraint to improve the accuracy depends on the rest of the parameters and it can even be different if we change the minimal support.

Finally, which of closed multiset set, generator multiset set or all the frequent multisets are the best sets is also not easy to answer. The closed multiset set produces the best results and none of the parameter sets based on the generator multiset is in the top 10 of the best parameter set 5.3. But it also depends on the other parameters and there is not a best multiset set for all the parameter sets even if we set the minimal support.

Those results on real datasets highlight the fact that a more complex model does not always improve significantly the accuracy results. When the goal of a classifier is to be interpretable, it seems then that a simpler model producing similar accuracy results is preferable. Furthermore, the best results obtained with linear SVM patterns are obtained for some specific parameter sets. Using a model with fewer parameters and so potentially producing more simulatable pattern sets will be preferable.

5.3 Conclusion and perspectives

The interpretability is, for learning models or a pattern models, a desired property but difficult to prove. For example, I firstly stated that chronicle is an interpretable pattern model without knowing exactly what this interpretability meant. This interpretability was then extended to the pattern-based classification interpretability. For this type of interpretability, I noticed a difference between: (i) the interpretability of a linear SVM classifier using chronicle as features and (ii) the interpretability of a simpler model using the most relevant discriminant chronicle in a sequence as a rule. I manage then to determine which properties I wanted to find in an interpretable model to define the interpretability concept.

From Lipton's definitions [Lipton, 2016], I determined two definition categories of the interpretability for patterns: (i) pattern models interpretability and (ii) pattern sets interpretability and by extension the use of pattern sets in classification. Those definitions lead to a trade-off when choosing the good pattern model for a specific task. At first glance, we can see a trade-off between the pattern model interpretability and the accuracy of the classifier using patterns as features. But it is often possible to produce a sufficiently big pattern set to obtain a good accuracy with a classifier using simple patterns as features. We observe then a second trade-off between the interpretability of the pattern model and the interpretability of the pattern set. A more interpretable pattern model

can produce a less interpretable pattern set if it requires more patterns to represent knowledge.

We have the choice to improve the accuracy with a more complex pattern model and potentially less interpretable or more patterns and a less interpretable pattern set. The accuracy is then used as an evaluation of this trade-off. We need more patterns to obtain a good accuracy with a simple pattern model and a more complex pattern model with a small pattern set.

Finally, those intuitions were experimentally tested. Firstly, we generated synthetic datasets that could highlight the trade-off between the complexity of the model and the size of the pattern set. We extracted then from those datasets several types of temporal patterns that we group in the generalized chronicle framework. This experiment showed that some patterns are more appropriated for certain types of datasets. We obtained a great difference between the results obtained with discriminant partially ordered patterns and with discriminant chronicles.

The synthetic datasets were generated from a behavior that could not be caught by these pattern models. However, we obtained a good accuracy of classification using discriminant chronicles as features and a bad one using the discriminant partially ordered patterns as features. This can be explained by the fact that the discriminant behaviors in the synthetic datasets are only correlated with temporal information. The sequential representation of the time done by the partially ordered patterns is then not sufficient enough to catch frequent discriminant behaviors in the dataset. On the other hand, even if the chronicles are not designed to catch the discriminant behaviors of the dataset, the specific cases that they caught were sufficiently frequent to be used as features. Moreover, the sequentiality could not represent certain specific cases of the datasets where it will be possible to generate a discriminant chronicle for each sequence of the dataset *i.e.* overfitting the dataset and then obtain a perfect classifier. Then, there is a trade-off for some pattern model between the complexity of the model and the size of the pattern set to produce a good classifier but too simple model on certain types of dataset cannot produce good classifiers even with a pattern for each sequence.

The second observed difference is between the chronicles and the linear SVM patterns. For linear SVM patterns, we obtain a perfect accuracy for the simple synthetic datasets and a near to be perfect accuracy for the other synthetic datasets. It is worth noting that those accuracy results are obtained using only one pattern as feature. In this case the interpretability of the pattern set is reduced to the interpretability of the pattern and switching the classifier has hardly any impact on the result. For chronicles, the accuracy is quite good but not perfect. Several chronicles are needed to produce a good classifier. We see then a different gap between those two models of patterns than between partially ordered patterns and chronicles. Here, the chronicle model can produce a perfect model with a bigger pattern set but the minimal support parameter regulates the size of the pattern set. If we want to obtain a perfect classifier, we then have to choose between two options: (i) a single pattern of a model that we consider less interpretable or (ii) numerous patterns of a model that we consider more interpretable. It is not obvious to answer which of a pattern of chronicles or a single linear SVM pattern is the most interpretable. Even if chronicles are considered more interpretable, too big a pattern set of chronicles will be less interpretable than a unique linear SVM pattern.

Finally, we classified real datasets using both chronicles and linear SVM patterns. This experiment shows that a classifier based on linear SVM patterns is not significantly more accurate than a classifier based on chronicles in each case. Indeed, at the opposite of the previous case where the partially ordered patterns were not an interesting choice for our synthetic dataset because they do not catch the temporal information, it can be useless to use too complex a model if it does not improve the accuracy.

All those experiments lead to two conclusions:

1. There is not a best pattern that optimizes in all cases the trade-off between the two categories of interpretability and the accuracy.

2. A model considered less interpretable can produce more interpretable pattern sets if it is the most suitable for a type of dataset.

These results can be used as guidelines to choose the most suitable pattern model for a task. It is useless to choose too complex a model if a simple one does the same task. It is also useless to choose too simple a model that will generate a deluge of pattern that could finally not be interpretable.

The perspectives of this work could then be twofold: (i) Explore the domain of the interpretable pattern models and (ii) explore the generalization of chronicles that are not interpretable.

Even using the definitions given in Section 5.1 and the other Lipton's definitions, it remains difficult to define objectively a pattern model or a pattern set as interpretable or not. Thereby, we still lack an objective measure for interpretability. Such measure could replace the use of the accuracy in classification, and cardinality of the pattern set to evaluate the results of a pattern model. We could then categorize the different pattern models in a hierarchy of interpretable models.

Pattern-based classification is initially designed to produce interpretable classifiers or at least classifiers based on interpretable features. However, pattern-based classification can produce very good accuracy results as shown in recent paper [De Smedt et al., 2017]. The second perspective will be then to explore the possible choices of pattern model for pattern-based classification without the interpretability constraint.

Chapter 6

Case study: Discriminant chronicle mining for pharmaco-epidemiology

Pharmaco-epidemiology is the study of the uses and effects of drugs and healthcare products in real life within well-defined populations. In other words, the main goal of pharmaco-epidemiology studies is to analyze the correlations between occurrences or co-occurrences of medical events (drug consumption, hospitalization, diseases, etc.) for specific populations. From these analyses, the pharmaco-epidemiologists attempt, for example, to prevent bad uses of drugs or adverse effects of drug co-occurrences.

An example of pharmaco-epidemiology study is the analysis of anti-epileptic drug substitution for epileptic people (GENEPI study [Polard et al., 2015]). A substitution to another anti-epileptic drug can occur with generic drugs. An epileptic person could then switch from a treatment based on a brand-named anti-epileptic drug to a treatment based on similar generic drug (same molecule and same dosage). The question raised by pharmaco-epidemiologists is: "Can a generic substitution imply an epileptic seizure?".

The standard way to conduct such study starts by elaborating a hypothesis to assess. Its first specifies the patient to study. After this step, cohorts are created and studied within a desired period. After the data collection, a statistical model is constructed to valid or not the initial hypothesis of the study. This straightforward way of conducting studies explore only one hypothesis at a time.

But they would also be interested in discovering new type of substitution and explore multiple hypotheses that may explain epileptic seizures. For instance, the substitution for a dosage to another or from a molecule to another. And maybe more complex care pathways than just two static behaviors.

6.1 Pharmaco-epidemiology based on the SNDS

The SNDS is a French medico-administrative database. Medico-administrative databases collect data concerning the medical domain with administrative purposes. In the case of the SNDS, all the reimbursement made by the French public healthcare insurance are recorded. Figure 6.1 illustrates what is contained by such databases. In this illustration, the main database regroups all information about the insured people. This is a true and fair view of the real database. In this database, basic information like surnames, forenames, dates of birth or gender are stored. Information collected through logs is stored in the additional subbases. For example, one base is dedicated to pharmacy service reimbursements. The sequence of all the pharmacy services of a specific insured can be built based on the information of those two bases. The figure 6.2 illustrates the data that we obtain

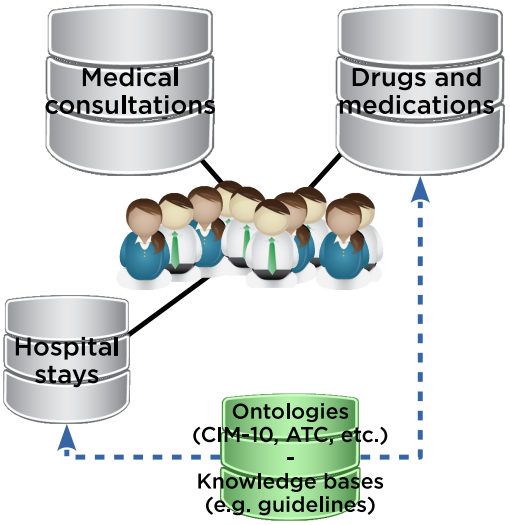


Figure 6.1: Illustration of the information diversity contained in medico-administrative databases. The grey subbases represent the datalogs and the green subbase represent the knowledge used to design the base or already known.

from the SNDS. The data are represented by a healthcare pathway list denoted *patients*. In this global list, each pathway is represented by three different event pathways and metadata. These three lists, denoted *prestations*, *prestationsMedicament* and *prestationsBiologie*, are relative to medical consultations, drug deliveries and biology laboratory services, respectively. In these lists, events are described by generic information such that, begin and end timestamps, the performer or the prescriber ids, and by specific information. For example, the event in *prestationsMedicament*, described from line 18 to line 24, begins and ends at timestamp 20090826 and is a drug denoted A12AX in the Anatomical Therapeutic Chemical (ATC) Classification System and 3387873 in the CIP hierarchies.

The advantage of using the SNDS instead of standard pharmaco-epidemiology cohorts is twofold:

1. The data are already available. Pharmaco-epidemiology studies do not have to wait the end of data collection to study them.

2. The data covers a large population. The French insured population is about several tens of millions of people. Well-defined population like, for example, the epileptic people, will then be well represented by those bases even if a great majority of the whole database is not relative to epileptic people. Studying such data ensures then to avoid the bias implied by to small cohort.

The first advantage of using the SNDS for pharmaco-epidemiology is the data availability. Data have been collected in the SNDS for many years. Using such database will save the time that would have been spent by collecting the data from a cohort. Furthermore, this can also reduce the study cost because the construction of a cohort needs some material and human resources.

The second advantage of such approach is the large population covered by this database. Indeed, the French insured population is several tens millions of people for several years. A great part of data that could be obtained by cohorts can be substituted by subsets of this database.

The main drawback of using medico-administrative databases for pharmaco-epidemiology studies is that these databases are not designed for this purpose but only for reimbursement management. The data have to be processed for removing useless information for pharmaco-epidemiology

```
{'patients': [{
  'birthYear': '1960',
  'localisation': '018218',
  'sexe': '2',
  'uid': '1',
  'prestations': [{
        'begin': '20101123',
        'end': '20101123',
        'performer': {'type': 'S.1', 'uid': 'A1933529.1.0'},
        'prescriber': {'type': 'S.1', 'uid': 'A1933529.1.0'},
        'type': '1111'}, {
        'begin': '20100315',
        'end': '20100315',
        'performer': {'type': 'S.1', 'uid': 'A1933529.1.0'},
        'prescriber': {'type': 'S.1', 'uid': 'A1933529.1.0'},
        'type': '1111'}, ...],
  'prestationsMedicament': [{
        'begin': '20090826',
        'end': '20090826',
        'medicament': {'atc': 'A12AX', 'cip': '3387873'},
        'performer': {'type': 'A.50', 'uid': 'A1543517.0.50'},
        'prescriber': {'type': 'S.1', 'uid': 'A1533560.1.0'},
        'quantityMedicament': '1',
        'type': '3313'}, {
        'begin': '20090701',
        'end': '20090701',
        'medicament': {'atc': 'N03AX09', 'cip': '3389843'},
        'performer': {'type': 'A.50', 'uid': 'A1543517.0.50'},
        'prescriber': {'type': 'S.1', 'uid': 'A1533560.1.0'},
        'quantityMedicament': '3',
        'type': '3313'}, ...],
  'prestationsBiologie': [{
        'begin': '20091009',
        'biologie': '519',
        'end': '20091009',
        'performer': {'type': 'A.30', 'uid': 'A1960457.0.30'},
        'prescriber': {'type': 'S.1', 'uid': 'A1933529.1.0'},
        'type': '3211'}, {
        'begin': '20091009',
        'biologie': '592',
        'end': '20091009',
        'performer': {'type': 'A.30', 'uid': 'A1960457.0.30'},
        'prescriber': {'type': 'S.1', 'uid': 'A1933529.1.0'},
        'type': '3211'}, ...]}, ...]}
```

Figure 6.2: Data example obtained from the SNDS. The data are represented in JSON format. This example represents a list of patient healthcare pathways. In this list, we can see one pathway. The pathway description begins with metadata as the birth year (line 2), the gender (line 4) or the reference patient ID (line 5). The different recorded events of this example are then described in the three lists: *prestations*, *prestationsMedicament* and *prestationsBiologie*. They correspond respectively to the medical consultations, to the drug deliveries and to the biology laboratory services, respectively.

studies like prices and reimbursement rates but also for completing them [Moulis et al., 2015].

Indeed, there is a gap between the administrative semantic and the medical information. The translation from one to the other requires strong knowledge in the domain. For example, we can see on Figure 6.1 that several ontologies are linked with the SNDS bases. Several of these ontologies are used to understand what type of events are logged in the SNDS. For example, the drugs are described by their code in the international ATC taxonomy. In the Anatomical Therapeutic Chemical (ATC) classification system, the active substances are divided into different groups according to the organ or system on which they act and their therapeutic, pharmacological and chemical properties. Drugs are classified in groups at five different levels. An issue with such taxonomy is that a drug or a chemical substance can have several codes. For example, the *alprostadil* is classified by codes beginning by C01EA and G04BE. It means that if we consider the *alprostadil* delivery as an event, all the codes linked to it have to be processed using the ATC taxonomy. Thereby, the processing of such data to be used for a specific study can be difficult. Issues may be encountered with disease encoding.

Furthermore, a part of the data is not accurate, incorrect or incomplete. For example, in the case of the SNDS, only the delivery of a drug is known and not the real medication time. It is due to the fact that the logged information dedicated to the reimbursement is done in the drugstore and not when the medication is really taken. Thereby, an event of drug delivery could be logged but the drug never took. Hypotheses then have to be done based on known guidelines to deduce the real medication time. Diagnoses made by doctors are also missing from the SNDS, a reimbursement was done for a specific out-hospital consultation but the reason is unknown. We only have disease information for hospital stays. Finally, the human factor may also impact the data quality. Practitioners have to fill this information accurately. This is our assumption even if we know that is not necessarily true. The quality of the data requires a great effort to combine them with expert knowledge.

The ANSM/PEPS project challenge was to conduct such pharmaco-epidemiology studies based on the SNDS data. The work done in this thesis is dedicated to improve the data exploration that is difficultly doable manually due to the data complexity and volume.

PEPS is the French acronym for Pharmaco-Epidémiologie des Produits de Santés that means pharmaco-epidemiology of the healthcare products. This project, supported by the University Hospital of Rennes, gathers different scientific partners in a consortium which conducts pharmaco-epidemiology studies. This project is funded by ANSM (Agence National de Sécurité du Médicament et des produits de santé) which is the French Agency for the Safety of Health Products. The goal of this funding for the ANSM is to improve the carrying out of studies independent of the pharmaceutical industry.

6.2 GENEPI

The GENEPI study (GENeric substitution of anti-EPIleptic drugs) is a typical example of pharmaco-epidemiology study conducted as part of the PEPS project. The expected outcomes were to evaluate the potential of the SNDS for pharmaco-epidemiolgy studies and to develop new data analytic tools dedicated to epidemiologist needs. Details and results of this study are described in [Polard et al., 2015].

The goal of this study was to further assess the association between seizure-related hospitalization and generic substitution and provide the necessary evidence to inform clinician and patient decisions. Clinicians working on this study tried to validate or invalidate the hypothesis that a switch from a brand-named anti-epileptic drug to a generic anti-epileptic drug can imply an epileptic seizure.

To do this, the studied population had to be defined. Indeed, the whole set of people for whom

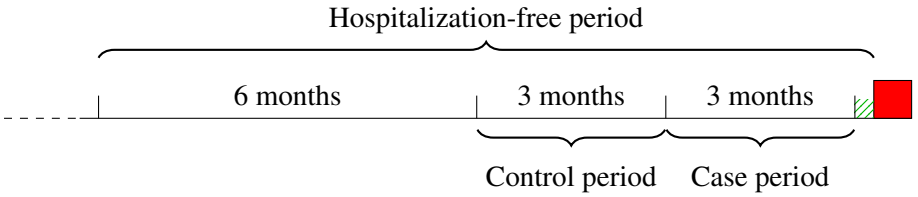


Figure 6.3: Illustration of the heathcare pathway of patients concerned by GENEPI. The index event is the hospitalization with epilepsy diagnosis represented by the red rectangle on the illustration. Before this, a 3-day induction period is removed from the data. The case and control periods are the two 3-month exposure window occurring before the induction period.

we know such switch or seizure is not adapted to test the hypothesis. For example, it is difficult to conclude from care pathways representing people who have seizures after substitutions if they often have seizures. The clinicians chose then to consider only the controlled epilepsy patients *i.e.* people known to have a stable treatment before than the seizure occurred.

To be sure to obtain data concerning generic substitution, a larger population was requested from the SNDS. The data obtained from the SNDS concern all adult patients affiliated to the French national health insurance scheme, 18 years old or older in January 2009, who had at least one reimbursement between 2009 and 2011 for at least one of the following anti-epilieptic drugs: *carbamazepine*, *lamotrigine*, *levetiracetam*, *topiramate*, or *valproic acid*, referred to as selected anti-epilieptic drugs (see Appendix B). These drugs had a brand name and at least one "A-rated" generic form available on the French market by this time (2009-2011) and were widely prescribed for epilepsy, allowing identification of epilepsy patients. They were extracted from the SNDS database using their ATC codes: N03AF01, N03AX09, N03AX14, N03AX11 or N03AG01.

The collected data were then pruned to remove specific populations that could bias the results. Patients with a medical history of cancer and women who gave birth within the study period were excluded because of the high risk of repeated seizures as well as patients receiving fatty acid derivatives through formulations registered as mood-stabilizing drugs such as valpromide (DEPAMIDE) or divalproex sodium (DEPAKOTE) who may not be epilepsy patients. The events related to cancer or birth were detected by their ICD-10 codes. The International Statistical Classification of Diseases and Related Health Problems (ICD), is a taxonomy similar to ATC but dedicated to the diseases.

Finally, the target population is selected from the remaining data. To study the effect of generic substitution on epileptic seizures, patients known to be hospitalized for such seizure between January 2010 and December 2011 were selected. Such selection is made by selecting all patients concerned by an event with the ICD-10 codes G40.x (epilepsy) or G41.x (status epilepticus) as codes of interest in primary or secondary hospital discharge diagnosis position. The controlled epilepsy patients were at last defined to have a hospitalization-free period for the epilepsy ICD-10 codes of at least a year before the selected hospitalization. Furthermore, a regular dispensation was defined as at least 10 dispensation claims within a year for the same anti-epileptic drug.

After all those pre-processing steps, a statistical model was designed to test the hypothesis. For each patient, a case cross-over was done. In this case, the case period *i.e.* the three months before the hospitalization for epileptic seizure was compared with the control period *i.e.* three months before. Figure 6.3 illustrates the studied periods of the healthcare pathway. The statistical model lies in four cases: the people for whom a generic substitution occurs in both phases, those for whom a generic substitution occurs only before the seizure, those for whom a generic substitution occurs only in the first phase and those for whom no generic substitution occurred. Table 6.1 regroups the different number of patients in each case. Intuitively, the epileptic seizure seems to

		Case period	
		No	Yes
Control	No	7222	478
period	Yes	491	188

Table 6.1: Number of patients according to the occurrence of brand-to-generic substitution in case and control periods. The majority of patients is not concerned by generic substitutions. The number of patients for which a generic substitution occurs only in the case period is similar to the number of patients for which generic substitution only occurs in the control period.

not be correlated with generic substitution occurring in case period. The conclusion of this study was negative *i.e.* the hypothesis was invalidated [Polard et al., 2015].

6.3 Discriminant chronicle mining for pharmaco-epidemiology

Based on the previous description of the GENEPI study, we can highlight some weakness in the standard process of a pharmaco-epidemiology study:

- The selection of the targeted population takes a lot of times.
- The temporal dimension is weakly taken into account.
- Only one kind of substitution was studied for only 5 different types of event.

Regarding the first point, a way to reduce the time allocated to the selection of patients is to use an expressive formalism to discriminate the patients. Furthermore, this formalism has to be usable through an efficient tool extracting a set of patients from data. The chronicle model is a good formalism for this task. The tool in this case is not a discriminant chronicle mining algorithm but a chronicle matching algorithm. Matching chronicles in big sets of patients can be efficiently done and we propose then to use such formalism in the project.

The chronicle model can also improve the description in temporal behavior for the study. Firstly, the chronicle model can be used to represent the behavior of controlled epilepsy. It can be used to represent regular treatment periods used in GENEPI study as illustrated by the left chronicle on Figure 6.4. But we could imagine more constrained regular treatment period as illustrated by the right chronicle on Figure 6.4. This illustrates how a simple model can empower epidemiologists in their analysis. Secondly, chronicles could represent more complex behavior than the occurrence or not of a generic substitution in a period.

The discriminant chronicle mining is the logical continuation of the use of a temporal formalism for matching patients. Instead of just describing one hypothesis with a chronicle, we want to automate this approach by extracting a set of chronicle that could be interesting hypotheses. The care pathways in which such hypotheses represented by chronicles occur could then be selected to be further analyzed. The involvement of the clinician will then be greatly reduced along the study pipeline. The chronicles extracted by our algorithm not only introduced the temporal dimension in the hypotheses but also specified what the interesting temporal behaviors that could be studied are. From an extracted set of chronicles, each hypothesis represented by a chronicle could be automatically assessed as statistically significant or not. The next parts of this chapter present the extraction of discriminant chronicles from the GENEPI dataset, their meanings as hypotheses and the conclusion that we can draw.

The following sections present the application of DCM to study the care pathways of epileptic patients contained by the GENEPI dataset. To extend this study and exploit the dataset beyond

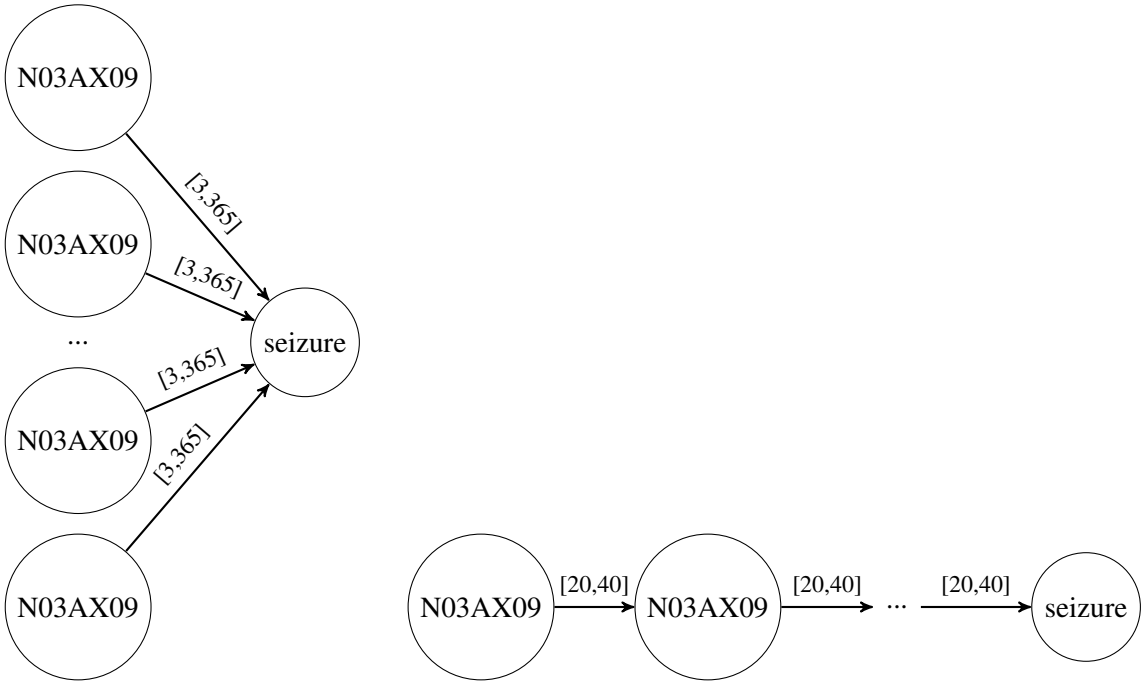


Figure 6.4: The chronicle on the left illustrates the behavior matched for the regular treatment period in GENEPI. The chronicle on the right is an example of a more complex behavior that could be wanted for regular treatment period.

the original study, we are interested in discovering other drug co-occurrences that could imply epileptic seizures using a more precise description of the temporal dimension, the DCM algorithm is used to extract patterns of drug deliveries that discriminate seizure occurrences.

6.4 Positive and negative datasets construction

This section presents the transformation of medical database into a dataset of labeled sequences. This transformation is required to apply the DCM algorithm and has important consequences on the results and their interpretation. In fact, depending on input data (event types, sequence lengths, etc.), the extracted patterns may change a lot. We compared several sequence encodings and the one proposed below appears to be the most interesting.

Our dataset was obtained from the SNDS database which contains a lot of tables with a large number of attributes (*cf.* section 6.1). Two important choices are required to obtain datasets of labeled sequences: 1) the choice of the set of events on which sequences will be built and 2) the sequence labeling.

The care pathway of each patient is the collection of timestamped drugs deliveries and hospitalization diagnosis from 2009 to 2011. All other care events have been ignored in this study. For each drug delivery, the event identifier is a tuple $\langle m, grp, g \rangle$ where m is the ATC code of the active molecule, $g \in \{0,1\}$ such that $g = 1$ when drug is generic and $g = 0$ when it is a brand-named drug and grp is the speciality group. The speciality group identifies the drug presentation (international non-proprietary name, strength per unit, number of units per pack and dosage form). Some drugs are not available in a generic format or have only consumption mode. For some others, the information seems to be missing. The event identifier representing such drugs is in this case equivalent to their ATC drugs.

Two other formats representation of drug deliveries were used before this one: by ATC and by

CIP[1] codes. The issue with ATC is the lack of information of this representation. For example, the ATC code does not contain the information that a drug was delivered in its generic or brand-named format. Discriminant behaviors like, for example, those describing generic substitutions are impossible to extract from such data. Furthermore, the representation is too coarse to show interesting behaviors and no discriminant chronicles could be extracted.

The issue with the CIP representation is at the opposite of the issue for the ATC representation. With the CIP, all drugs that are not contained by same packaging are different. For example, two drugs containing the same molecule and of the same specificity are different if the pill numbers in the packages are different. This leads to few frequent behaviors and then to no extracted discriminant chronicles. The choice of the triplet as event identifier is a trade-off between the CIP information and the ATC generalization.

A case-crossover protocol, presented in Section 6.2, has been adapted to label sequences. This protocol is often used in pharmaco-epidemilogy studies. It studies the possible causes of an outcome using a patient as his/her own control. In our case, the patient outcomes are seizure events of epileptic patients. But in our case, the patient is not only his/her own control. All the case sequences are compared with all the control sequences. We have so a global control and a global case datasets.

A case and a control sequence are generated from each patient healthcare pathways. Drugs delivered within the 90 days before induction yield the case sequences and those delivered within the 90 days before the case sequence, *i.e.* the 90 to 180 days before induction, yield the control sequences. A 3-day induction period is defined before the first seizure of each patient. For the comparison, the experiment was also done with 60, 120 and 150 days instead of 90. In the original study, they evaluated the sensibility of the analysis with both 28-day periods for control and case periods and 1 and 5-day induction period. The results were unchanged.

At the end of the data preparation, the dataset contained two sets of 8,379 labeled sequences, and contains 127,191 events corresponding to 1,716 different drugs.

6.5 Experiments and results

Set up with $\sigma_{min} = 5.10^{-3}$, *i.e.* 42 patients[2], and $g_{min} = 1.4$, we generated 777 discriminant chronicles that discriminate the case sequences from the control sequences of 90 days. Chronicles involved 510 different multisets and 128 different event types. Similar results were obtained with sequence durations of 60, 120 and 150 days. For comparison, 535 chronicles were extracted for the dataset containing sequences of 150 days with the same parameters. The main difference between those two datasets is that chronicles are discriminant in the "150 days" dataset for higher g_{min}. For example, 487 chronicles are discriminant for $g_{min} = 1.8$. Our intuition is that regular treatments are more easily discriminated from punctual deliveries when the sequences are larger. Larger discriminant chronicles are extracted from the 150 days dataset than from the 90 days dataset and these chronicles describe more accurately the underlying discriminant behavior.

Three types of pattern are of specific interest for clinicians: (1) sequences of anti-epileptic generic and brand-named drug deliveries, (2) sequences of same anti-epileptic drug deliveries, (3) sequences with anti-epileptic drug deliveries and other types of drug deliveries. According to these criteria, we selected 55 discriminant chronicles involving 16 different multisets to be discussed with clinicians. We choose to focus the remaining of this section on chronicles related to *valproic acid* (*N03AG01* ATC code, with different presentations) because it is the most frequent

[1]The CIP code is a French code related to marketing authorization. Thereby, this code can be used to describe a specific drug packaging with specific dosage and speciality group.

[2]This number of patients have been initially estimated important by epidemiologist to define a population of patients with similar care sequences associated to seizure.



Figure 6.5: Four discriminant chronicles describing switches between same type of valproic acid (N03AG01) generic (G 438) and brand-named (R 438). $supp(\mathscr{C}_i, \mathscr{S}^+)$ respectively for $i = 1$ to 4 equals 43, 78, 71 and 43 and $supp(\mathscr{C}_i, \mathscr{S}^-)$ equals 23, 53, 39 and 30.

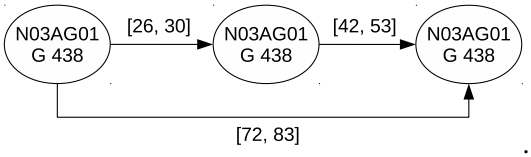


Figure 6.6: Extracted chronicle describing repetitions of *valproic acid* (N03AG01) generic (G 438). This chronicle is one of the most related to epileptic seizures with a growth rate greater than 2: $supp(\mathscr{C}, \mathscr{S}^+) = 50$, $supp(\mathscr{C}, \mathscr{S}^-) = 17$.

anti-epileptic drug occurring in the dataset but our results contain similar chronicles related to other anti-epileptic drugs like *levetiracetam* or *lamotrigine*.

6.5.1 Taking into account time in brand-to-generic substitution

We start with patterns representing switches between different presentation of *N03AG01*. Figure 6.5 illustrates all discriminant patterns that have been extracted. It is noteworthy that all chronicles have temporal constraints, this means that multisets without temporal constraints are not discriminant. These results are consistent with [Polard et al., 2015] which concluded that brand-to-generic anti-epileptic drug substitution was not associated with an elevated risk of seizure-related hospitalization. But temporal constraints was not taken into account in the later. The four extracted chronicles suggest that for some small patient groups, drug switches with specific temporal constraints are more likely associated to seizure.

The first two chronicles represent delivery intervals lower than 30 days, from brand-to-generic and generic-to-brand respectively. The third one represents an interval between the two events that are greater than 30 days but lower than 60 days. The discriminant temporal constraints of the last one could be interpreted as $[67, 90]$ because of the bounded duration of the study period (90 days). This chronicle represents a switch occurring more than 60 days but most of the time less than 90 days.

These behaviors may correspond to unstable treatments. In fact, anti-epileptic deliveries have to be renewed every month, thus, a regular treatment corresponds to a delay of ≈ 30 days between two anti-epileptic drug deliveries.

We next present in Figure 6.6 an example of discriminant chronicle that involves three deliveries of *N03AG01* (no chronicle involves more deliveries of this anti-epileptic drug).

The growth rate of this chronicle is high (2.94). It is easy to understand and, with their discriminant temporal constraints, it can be represented on a timeline (see Figure 6.7). It is noteworthy that the timeline representation loses some constraint information. The first delivery is used as starting point (t_0), but it clearly illustrates that the last delivery occurs too late after the second one (more 30 days after). As well as previous patterns, this chronicle describes an irregularity in deliveries. More precisely, the irregularity occurs between the second and the third deliveries as



Figure 6.7: Timeline representation of the chronicle illustrated by Figure 6.6.

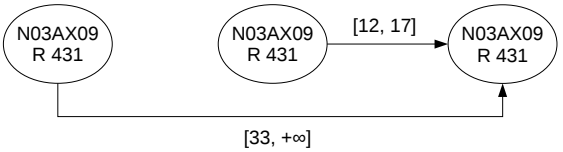


Figure 6.8: Chronicle describing repetitions of *lamotrigine* (N03AX09) brand-named (R 431). This chronicle occurs twice more times in case sequences than in control sequences: $supp(\mathscr{C},\mathscr{S}^+) = 42$, $supp(\mathscr{C},\mathscr{S}^-) = 21$.

described by the discriminant temporal constraints $[42,53]$ and $[72,83]$.

Similar chronicles were extracted for other types of anti-epileptic drugs. Figure 6.8 illustrates such chronicle. The chronicle of Figure 6.8 is similar to chronicle of Figure 6.6 but with lamotrigine delivery events instead of valproic acid delivery events. It has a lower growth rate than the previous chronicle (2 against 2.94). This chronicle is still very discriminant as it occurs twice more times in the case sequences than in the control sequences. The behavior described by the chronicle is also more deviant regarding the guideline. In France, anti-epileptic drugs are delivered monthly. This chronicle describes cases where two lamotrigine deliveries occur between 12 and 17 days one after the other. We conclude that this chronicle also describes delivery irregularities that are deviant from the delivery guidelines. Such irregularity in epileptic treatment represents an unstable treatment and is known by clinicians to be correlated with seizures.

We conclude from observations on the previous two types of patterns that the precise numerical temporal information discovered by DCM is useful to identify discriminant behaviors. Analyzing pure sequential patterns does not provide enough expression power to associate switch of same anti-epileptic deliveries with seizures. Chronicles, specifying temporal constraints, allow us to describe the conditions under which a switch of same anti-epileptic deliveries is discriminant for epileptic seizures.

6.5.2 Example of a complex chronicle leading to new hypotheses

Clinicians found the chronicle presented in Figure 6.9 interesting, as it is at first sight not corresponding to an expert knowledge. It illustrates a potential adverse drug interaction between an anti-epileptic drug and a drug non-directly related to epilepsy, more especially aspirin (*B01AC06*), prescribed as an anti-thrombotic treatment. The discriminant temporal constraints imply that aspirin and paracetamol (*N02BE01*) are delivered within a short period (less 28 days). There is no temporal relations between these deliveries and the deliveries of *valproic acid*. But their co-occurrence within the period of 90 days is part of the discriminatory factor.

After a more thorough analysis of patient care pathways supporting this chronicle, clinicians made the hypothesis that these patients were elderly people, treated for brain stroke. It is known for seriously exacerbating epilepsy and increasing seizure risk.

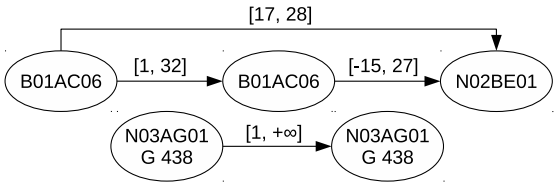


Figure 6.9: A chronicle describing co-occurrences between anti-thrombosis drugs (*B01AC06*) and *valproic acid* which is more likely associated to seizure: $supp(\mathscr{C}, \mathscr{S}^+) = 42$, $supp(\mathscr{C}, \mathscr{S}^-) = 20$.

6.6 Conclusion

We showed in this chapter that discriminant chronicle mining is a good method to explore pharmaco-epidemiology data and more especially to generate interesting hypotheses. In comparison with the efforts needed to produce a statistical model answering the question "Can an anti-epileptic generic substitution implies an epileptic seizure?", our discriminant chronicle mining algorithm generated easily a set of interesting pharmaco-epidemiology hypotheses. The set of discriminant chronicles was sufficiently small to be manually explored and analyzed.

However, the pre-processing of the dataset was not as simple as we imagined it. The main difficulty was to choose the good representation for the events. Indeed, a too general description of events leads to inefficient discriminant behaviors and too precise a description of events and to not enough frequent behaviors. We provide, after some iterations, an event representation that allows us to extract interesting discriminant chronicles.

Several iterations were required to effectively extract interesting patterns. These iterations were based on event representation using ATC or CIP taxonomies. However, the automation of the mining process allows us to be little involved in this process. Parameters was set up and, if the results were not good, they were adjusted to obtain the desired results. This approach involves the expert to evaluate the results and to guide the exploration but not to describe all the hypotheses and construct their models. Using the discriminant chronicle mining is finally a great time saving for the experts.

Finally, discriminant patterns have been presented to clinicians who conclude to their potential interestingness to explore care pathways supported by chronicles. The main interest of this experiment was to show that some interesting patterns can be extracted and that clinician can actually understand them. Contrary to black-box approaches that can efficiently discriminate different dataset type, this approach enables to bring insights to the expert about large data at hand. Furthermore, discriminant chronicle mining shows results in accordance with the previous results obtain through a standard study process. No discriminant chronicle linked only a generic substitution with an epileptic seizure and improved the study by describing the cases where a substitution (generic to brand-named and brand-named to generic) was related with an epileptic seizure. The extracted discriminant chronicles were also not limited to the substitution and described, for example, brain stroke cases.

This application of discriminant chronicle mining on care pathways shows that this approach is really useful for the automation of pharmaco-epidemiology study and hypothesis generation.

Chapter 7

Conclusion

The contributions of this thesis are threefold: (i) the discriminant chronicle mining task, (ii) the study of chronicles interpretability and (iii) a pharmaco-epidemiology case study.

The discriminant chronicle mining was introduced through the evaluation of the chronicle model to be relevant to represent temporal behaviors. Our application objective was to identify such temporal behaviors in care pathways and thus to conduct pharmaco-epidemiology studies. The chronicle model was chosen because it can represent complex and insightful temporal behaviors.

We proposed the discriminant chronicle mining task to improve the interestingness of the extracted patterns. Algorithms were already implemented to extract frequent chronicles efficiently. But in many different contexts, sequences can be labeled. The supervised dimension of the discriminant chronicle mining task allows the user to describe which behaviors are interesting. An effective and modular algorithm called DCM was proposed to solve this task. The discriminant chronicle model is effectively **extractable**.

Thereby, we brought a new **expressive** model of discriminant temporal patterns and an algorithm to effectively extract them. Because of the ability to express quantitative temporal constraints and the flexibility of the constraint order, discriminant chronicles subsumes the state of art of the discriminant temporal patterns. This model is more expressive than discriminant sequential patterns [Fradkin and Mörchen, 2015] and partially ordered patterns [Fabrègue et al., 2014] thanks to the temporal constraints. It is also more expressive than temporal annotated sequences (TAS) [Giannotti et al., 2006] with the management of non-sequential constraints.

The efficiency of discriminant chronicles to represent the dataset from where they were extracted was compared to sequential patterns through the comparison of pattern-based classification accuracy. Results have shown that discriminant chronicles are efficient as features in pattern-based classification. Such results on various types of datasets highlight the **genericity** of our approach.

Comparison of different temporal pattern models in terms of classification accuracy leads us to the question of the extracted pattern set interpretability. Indeed, we choose the chronicle model, among other possible models (for example neural networks or linear discriminant analysis), for its **interpretability**. The intuition was that the chronicle model is more interpretable than accurate model like deep neural network. This intuition is reinforced by the justifications used to promote pattern mining and pattern-based classification. For example, the abstract of [Bringmann et al., 2011] presents pattern mining as:

> Pattern mining can help to obtain models for structured domains, such as graphs and sequences, and has been proposed as a means to obtain more accurate and more interpretable models.

But we did not define what an interpretable model is before. A legitimate question is so: "Is the

chronicle model effectively interpretable?" To confirm this intuition, we needed to compare models based on their interpretability. To focus the comparison of several temporal pattern models, we generalized the notion of discriminant chronicle to a set of discriminant temporal pattern models. We defined then generalized discriminant chronicle interpretability through state-of-the-art definitions [Lipton, 2016] and highlighted the difference between several generalized discriminant chronicle models in terms of interpretability. We highlighted a trade-off between the interpretability of a model and its classification accuracy. We compared the accuracy and interpretability of several models experimentally on synthetic and real datasets.

Then, we analyze care pathways in the context of pharmaco-epidemiology studies. Through this application, we highlighted the importance of using chronicle mining in such studies:

- It improves the temporal representation of interesting behaviors.
- It enables to automatically explore a large search space of potentially interesting chronicles.

Indeed, complex discriminant behaviors were extracted as discriminant chronicles from the GENEPI dataset. In the original study, only sequentially behaviors were characterized (*i.e.* generic substitution). The discovered chronicles identified similar substitution but also temporal information characterizing the care pathways. Such precision in behavior description is more difficult to obtain with the standard process of pharmaco-epidemiology study, and a mandatory to answer their complex questions. Furthermore, the automation of the search space exploration allowed us to generalize the original study. We did not extract discriminant chronicles describing a generic substitution in correlation with an epileptic seizure in accordance to the original study. We extended the search to other discriminant behaviors. Our results on GENEPI study were validated by clinicians. This confirms that the approach is useful to conduct pharmaco-epidemiology study.

This thesis contributes and highlights the importance of temporal pattern mining. Indeed, the insights given by extracted discriminant chronicles to experts are very important. Such insights are difficultly obtainable using standard classification methods like deep neural networks. For example, the patterns extracted in GENEPI dataset are understandable by clinicians. A black box classifier could have better accuracy. In this thesis, the defended idea is that a slight improvement of accuracy must not be worth losing human interpretability.

7.1 Discussion and perspectives

In this section we discuss the different contributions of this thesis and their perspectives. These perspectives range from algorithmic improvements to the imagination of an objective interpretability measure.

7.1.1 Application

We have to think about how to deliver the algorithm to be used by clinicians. At the time of the case-study experiments, we first did experiments and then discussed the results with clinicians. The goal of such release would be that clinicians do the experiments themselves. More generally, we have to think about how to release the DCM algorithm to be used in a epidemiology study platform. Furthermore, the way to represent lots of chronicles have to be elaborate. Indeed, the DCM algorithm produces in some cases numerous patterns.

The initial idea of this work was to design a tool dedicated to support temporal data management. Figure 7.1 illustrates the proposed process scheme. This tool was thought as a whole data mining processing chain from the data pre-processing step (3) and then the pattern mining step (1) to the evaluation of the results (2).

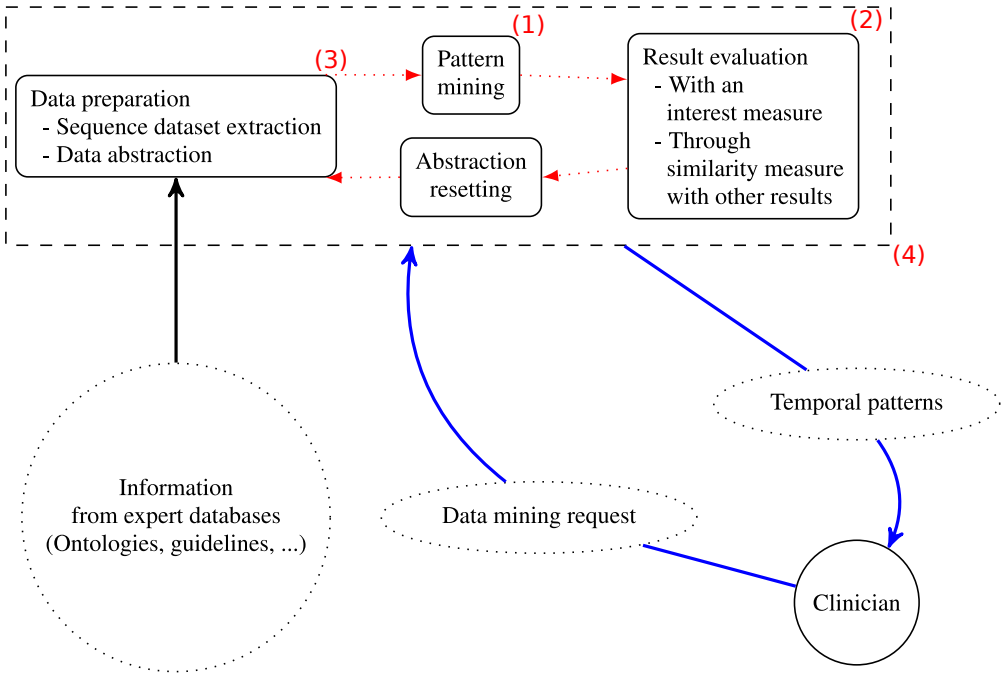


Figure 7.1: Illustration of the sequential pattern mining tools imagined at the beginning of the thesis. The rectangle dashed box was thought to be a language dedicated to the temporal data management.

In this thesis, we proposed a generic pattern mining algorithm that can be used within this tool (4) and proposed some way to evaluate the results. The perspectives dedicated to the generalization of discriminant chronicles and the evaluation of the interpretability would improve these points. Two components of the tool are still missing:

- The management of the pre-processing step (3)
- The user interface

One of the most challenging issues to tackle is the management of event taxonomies. This appears to be very important in the context of pharmaco-epidemiology. Such improvement would lead to merge the pre-processing step and the pattern mining step: the goal would be to extract patterns as much as possible from the raw data. We model a first proposition of such reconciliation exploring declarative pattern mining approach [Guyet et al., 2017].

The remaining works to do is so focus on the user interface. An idea of such interface would be to propose a domain specific language for querying care pathways. Such language has to be expressive enough to allow the user to request complex data management while being easy to use.

7.1.2 Discriminant chronicle mining algorithm

Some issues are remaining in our implementation of the discriminant chronicle mining algorithm.

The first issue concerns the multiple instance problem (see section 4.3). The algorithm $Ripper_k$ is used to extract discriminant numeric rules because it is one of the most efficient algorithms to produce such rules. The issue is that $Ripper_k$ is not dedicated to handle multiple instance problems. Indeed, there is a difference between the number of occurrences covered by a rule and the number of sequences covered by this same rule. Thereby, some not discriminant rules could be extracted instead of discriminant rules because of this problem. We have so no guarantee that $Ripper_k$ will

extract at least one discriminant rule for a multiset even if one exists. We can so highlight two weaknesses in our approach:

- There is no evaluation of the coverage of the search space done by DCM.
- We are not able to propose a solution based on a multiple instance algorithm that offers the same or best results than $Ripper_k$. At the moment, our attempts to use multiple instance classifier failed to improve our results and deeper studies are still to be done.

One attempt was done by using *MITI* [Blockeel et al., 2005] instead of $Ripper_k$ but the evaluation in terms of classification accuracy promote $Ripper_k$ as best algorithms for temporal constraint mining. It is interesting to notice that this work is the first to make the link between temporal sequences and multiple instance learning. There is certainly interesting theoretical questions to explore and algorithms to propose with an explicit management of this problem.

The perspectives concerning the multiple instance management are twofold. On the one hand, the evaluation of the search space coverage by the extracted discriminant chronicles by a formalization of the discriminant chronicle lattice. With a formalization of the discriminant chronicle search space, it would be easier to evaluate which type of chronicle is missing and how to improve or correct the $Ripper_k$ strategy. Such formalization could be obtained by extended those done for sequential patterns in formal concept analysis domain [Buzmakov et al., 2013, Otaki et al., 2014]. Such formalization will not directly solve the multiple instance problem but will allow to evaluate what is lost by not manage it.

On the other hand, the multiple instance domain is large and the right strategy to extract temporal constraint was presumably already studied. The choice of the better algorithm and the better multiple instance strategy can then be done experimentally by comparing a large collection of multiple instance algorithms.

Another issue of the algorithm application concerns the event granularity. We call here event granularity the level at which two different events are considered similar or not. For example, one sequence can be $\langle(banana,1),(orange,2),(bread,3)\rangle$ at a certain event granularity and $\langle(fruit,1),(fruit,2),(bread,3)\rangle$ at another. Case study of chapter 6 shown that it can be difficult to extract discriminant behaviors when the event granularity is too high *i.e.* too many events are considered similar. In the previous example, the event granularity level using $fruit$ as event type is too high if the discriminant behaviors are specific to *banana* or to *orange* but not both. For example, when drugs were only described by their ATC code, no discriminant chronicles were extracted. On the other hand, when drugs were described by their CIP code *i.e.* the most finest grain for drugs, no discriminant chronicles were frequent. We had to choose a compromised granularity where drugs were described by a triplet between ATC and CIP. But even with this compromise, extracted chronicles were rare and not very discriminant. We see in this example how difficult is the data preparation phase.

A perspective of improvement of handling such data would be to use hierarchical events in chronicle mining. The potential benefits would be twofold:

- Extracted discriminant behaviors could cover a larger population. Indeed, some extracted chronicles are rare because of few events that could be generalized. Using similar events but more generalized could improve the frequency of such chronicles without decrease too highly their growth rate. For example, if the discriminant behavior hidden in a dataset is "Buy a banana and then another fruit", the actual algorithm will extract all the chronicle describing the combinations of *banana* and another fruit. If the fruit contained by the dataset are *banana*, *orange* and *raspberry*, one chronicle could be extracted for each of the three

multisets: $\{\!\{banana, banana\}\!\}$, $\{\!\{banana, orange\}\!\}$ and $\{\!\{banana, raspberry\}\!\}$. The discriminant chronicle based on multiset $\{\!\{banana, fruit\}\!\}$ will so describe a behavior more general and so more frequent. It would provide fewer patterns with individual better significance.

- The pre-processing step of data would be easier without choosing the event granularity. All the information of the hierarchy would be contained in the dataset and the best granularity will be chosen by the algorithm for each pattern. It would prevent from doing the tedious data preparation task.

Such improvement could be done by extending sequential pattern mining with hierarchical approaches like *LASH* [Beedkar and Gemulla, 2015]. In the case of an extension of *LASH*, the additional complexity due to hierarchy management could be partially absorbed by the capacity.

7.1.3 Generalized discriminant chronicles

Comparison of pattern model interpretability in section 5 shown that simple models can effectively represent behaviors even in datasets containing complex behaviors. For example, the discriminant partially ordered patterns described behaviors of the synthetic dataset of chapter 5 even if this dataset was constructed on a complex temporal behavior. Thereby, losing interpretability for increasing accuracy by using more complex models would be justified only in case of significant increase of the accuracy.

The generalized discriminant chronicle mining is limited by the choice of the pattern model that will describe discriminant behaviors. If the dataset behaviors are not homogeneous in terms of complexity, improving the accuracy by using a more complex model will artificially penalize the pattern set interpretability. Thereby, a solution is to represent the simplest behaviors using more interpretable models and the most complex behaviors by the most complex model. This would justify a loss of interpretability only for the behaviors that are not or not sufficiently covered by the desired interpretable model. Such approach will then allow to:

- classify datasets with the same accuracy as the best classifiers
- represent each behavior with the most as possible interpretable pattern model

Thereby, there will be no more trade-off between interpretability and accuracy because the least interpretable model will be used to represent behaviors that are not representable by the most interpretable model.

Furthermore, such approach could highlight the genericity of the generalized discriminant chronicle mining algorithm. Indeed, a mixed pattern set can be extracted by the same algorithm. The difference with the actual algorithm would be the choice of which method to use to compute the occurrence mapping functions.

The generalization of discriminant chronicles we proposed was focus on the temporal dimension. Other data dimensions could be used to enrich the chronicle model and could increase the model accuracy. For example, the dataset dedicated to rule extraction done by $Ripper_k$ could be enriched by sequence metadata. In the pharmaco-epidemiology context, an example of metadata for healthcare pathways is the age. The rules extracted by $Ripper_k$ will contain conditions of the $age \leq x$. For example, the condition $age \leq 60$ specifies that the chronicle describes a discriminant behavior that only matches patients younger than 60 years old. The previous example could be generalized to other metadata such that gender or location.

Such approach is extendable to patterns like negative chronicles. In this case, the non-occurrence of events between two events of the multiset will be a dimension to discriminate. Table 7.1 illustrates a negative occurrence dataset from which the discriminant negative chronicle illustrated by

SID	$\mathscr{A}_{A\to B}$	$\mathscr{A}_{B\to C}$	$\mathscr{A}_{A\to C}$	$A \xrightarrow{?D} B$	$B \xrightarrow{?D} C$	$A \xrightarrow{?D} C$	Label
1	2	2	4	False	False	False	+
1	−1	2	1	False	False	False	+
2	5	−2	3	False	True	False	+
3	3	0	3	False	False	True	+
5	−1	3	1	False	False	True	−
6	6	−1	5	True	False	False	−

Table 7.1: Example of dataset describing the occurrences of multiset $\{\!\{A,B,C\}\!\}$. In this dataset there is two types of attributes: numerical attributes and boolean attributes. The duration between two events and the occurrence of an event between two others. For example, $A \xrightarrow{?D} B$ corresponds to the occurrence of D between A and B.

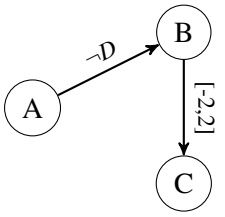


Figure 7.2: Example of discriminant negative chronicle that could be extracted from the dataset on Table 7.1. The constraint $\neg D$ between the events A and B specify that no occurrence of D can be found between A and B.

Figure 7.2 could be extracted. With A = *apple*, B = *bread*, C = *chocolate* and D = *dumplings*, the negative chronicle on Figure 7.2 will a represent a behavior matching the sequence where occur A, B and C but not D between A and B. This allows a greater expressiveness for the model: in this example, we can extract two specific behaviors where the one for which the *dumplings* are bought between *apple* and *bread* is different from the other one. The formalism of the generalized chronicle will be unchanged but in this case it is the occurrence definition that will be extended to any vector of attributes considered interesting to be discriminant.

7.1.4 Measures of interpretability

Interpretable models and, more particularly, interpretable pattern models are presented in this work as useful to give insights to domain experts and stakeholders. The goal of such knowledge discovery is to give insights to stakeholders. It aims as identifying the specificities of different classes described in a dataset. In this case, an interpretable model is not dedicated to automatically classify new examples but to highlight behaviors that would have been difficult to find manually.

Chapter 5 initiates the specification of interpretability. More efforts have to be done to make it more operational. Thereby, the comparison between two models requires a great degree of subjective assessment. A general agreement could define that a pattern model is interpretable or more interpretable than another but such agreement is not common and could differ for different populations. For example, data scientists could argue that a linear model is interpretable but clinicians could not agree.

A perspective of improvement of the work done on generalized discriminant chronicle interpretability is the definition of an objective interpretability measure. Such measure could be interesting to design a generalized discriminant chronicle model for a specific application. The model interpretability would be presented to the final user as the accuracy power of the model.

In accordance with Lipton's thoughts [Lipton, 2017] such measure has to be defined with or at least validated by the stakeholders, for example clinicians in the pharmaco-epidemiology context. Indeed, the definition of interpretable models has to be defined by a larger population than the researchers in data analysis domain. Furthermore, each domain could have its own definition of interpretable models. If the goal of an interpretable model is to be interpretable by domain experts or stakeholders who are not data scientists, the definition of interpretability has to be defined by them.

A way to construct such interpretability measure could be done through experiments on users. Such experiment will be based on the evaluation of several models by domain experts. One difficulty to do such experiment is to find enough relevant datasets and associated domain experts to ensure the quality of the evaluation. Another difficulty is to formalize the evaluation of the participants. However, such experiments would be useful to describe interpretable models and to learn what is effectively required as interpretability for the domain experts. Lipton wrote that "we must move past the word interpretable and decide which definitions to focus on" [Lipton, 2017] and such experiment could guide us to those definitions to focus on.

This work was based on the subjective opinion that temporal pattern mining is one of the most promising approaches to give insights and support stakeholders. Foundations of this opinion are the interpretability and explanability of the patterns. Patterns seem to be interpretable and explainable due to their simplicity and the simplicity of the measure for which they are extracted such that the support. But this assumption remains to be evaluated more objectively. The work presented in this thesis was dedicated to highlight the interestingness and interpretability of temporal pattern mining. An improvement of this work such that a measure of interpretability will help to promote temporal pattern mining as a good assistant tool for stakeholders. Furthermore, in a context where a right to explanation is claimed, temporal pattern mining seems to be a pertinent option face to popular but difficultly explainable classification models such that deep neural networks or bayesian approaches.

Glossary

ANSM Agence Nationale de Sécurité du Médicament (French Agency for the Safety of Health Products). 10, 98

ATC Anatomical Therapeutic Chemical Classification System. 96, 98, 99, 101, 102, 105, 110

CIP Code Identifiant de Présentation (Presentation Identifier Code): code required for drug marketing authorization decision in France. 96, 102, 105, 110

DCM Discriminant Chronicle Mining. 9, 53, 55, 62–68, 73, 100, 101, 104, 107, 108, 110

GENEPI GENeric substitution of anti-EPIleptic drugs: name of a pharmaco-epidemiology study conducted in the University hospital of Rennes [Polard et al., 2015]. 10, 11, 25, 50, 95, 98–101, 108

PEPS Pharmaco-Epidemiologie des Produits de Santé (Pharmaco-Epidemiology of Healthcare Products). 10, 25, 98

SNDS Système National des Données de Santé (National System for Healthcare Data). 4, 20, 95–99, 101

UCI University of California, Irvine: reference to the dataset repository created in this university `https://archive.ics.uci.edu/ml/index.php`. 53

Appendix A

DCM implementation

The DCM algorithm used in chapter 4 to extract discriminant chronicles was implemented in *C++* en based on the original code of $Ripper_k$ writted in *C*. This code was deposited at the Agence de Protection des Programmes under the number: IDDN.FR.001.440018.000.S.P.2017.000.20700. The DCM algorithm sources are available on the INRIA GitLab: `https://gitlab.inria.fr/ydauxais/DCM`.

The following pages are the README file of the DCM git deposit. It provides information about the software usage and its main parameters.

An alternative implementation of DCM has been implemented in Python with the purpose to experiment generalized discriminant chronicles. This version enables to easily change the occurrence mapping function and associated classifiers. The sources of this implementation are available on the INRIA GitLab: `https://gitlab.inria.fr/ydauxais/GDC-PBC`.

DCM

DCM (Discriminant chronicle mining) is a C++ implementation of two chronicle mining task. The extraction is done from set of temporal sequences.

- **Discriminant chronicle mining**: The main purpose of DCM is to extract discriminant chronicles from a positive dataset in comparison to a negative dataset. A chronicle will be considered discriminant if its support in the positive dataset is greater than g_{min} times its support in the negative dataset. The parameter g_{min} has to be defined by the user before the run. The implementation of this task in DCM is not complete as it will not extract the whole set of discriminant chronicles. This task has been detailed in **Discriminant chronicle mining: Application to care pathways**.
- **Frequent chronicle mining**: DCM contains also an implementation of the frequent chronicle mining task. The implementation of this task is complete according to the definition that the bounds used for the temporal intervals of the chronicles must occur in the dataset. This part of the implementation has not been maintained for a while and could contains some bugs. This implementation was used for the experiments in **Chronicles mining in a database of drugs exposures**.

License

This code, except files present in ***Ripper/ripper/code***, is written by **Yann Dauxais** and belongs to the **University of Rennes 1**. This code is licensed under the **BSD** License - see the **LICENSE.md** file for details. This code is deposited at the *Agence de Protection des Programmes* under the number: 'IDDN.FR.001.440018.000.S.P.2017.000.20700'. The files present in ***Ripper/ripper/code*** belong to **AT&T** and were written by **William W. Cohen**.

Setup

To compile DCM, just use the ***CMakeLists.txt*** in the root directory.

```
mkdir build
cd build
cmake ..
make
cd ..
```

If the compilation is successful, the executable ***Extract*** will be present in the ***bin*** directory.

Requirements

The compilation requires a ***GCC*** version greater or equal than 4.9 (supporting the flag `--std=c++0x` and the `<regex>` implementation).

The implementation includes `<boost/program_options.hpp>` that must be installed.

Examples

The directory ***examples*** contains the two subdirectories ***ECG*** and ***BIDE-D***.

The directory ***ECG*** includes the two datasets ***d214lbbb__H141.dat*** and ***d214pvc.dat***. Those datasets was generated from the MIT-BIH Arrhythmia database.

The directory ***BIDE-D*** includes a translation in our format of the datasets provided here.

The sequences in those datasets are represented using the line format.

The uses of the command `./bin/Extract examples/ECG/d214lbbb_H141.dat -d examples/ECG/d214pvc.dat -f 0.8 -g 2` will return 4 discriminant chronicles:

```
C: {["qrs[abnormal]", "qrs[abnormal]", "qrs[abnormal]"]}
0, 1: (686, 881)
0, 2: (-inf, inf)
1, 2: (539, 894)
f: 125/20

C: {["qrs[abnormal]", "qrs[abnormal]", "qrs[abnormal]",
"p_wave[normal]", "p_wave[normal]"]}
0, 1: (-inf, inf)
0, 2: (-inf, 1861)
0, 3: (-173, inf)
0, 4: (706, inf)
1, 2: (653, 1011)
1, 3: (-inf, inf)
1, 4: (-inf, 706)
2, 3: (-inf, inf)
2, 4: (-inf, inf)
3, 4: (-inf, 1653)
f: 116/0

C: {["qrs[abnormal]", "qrs[abnormal]", "p_wave[normal]"]}
0, 1: (1402, 1705)
0, 2: (544, 1384)
```

```
1, 2: (-inf, inf)
f: 119/41

C: {["qrs[abnormal]", "qrs[abnormal]", "p_wave[normal]",
"p_wave[normal]", "p_wave[normal]"]}
0, 1: (-inf, inf)
0, 2: (-inf, inf)
0, 3: (-inf, inf)
0, 4: (-inf, 1564)
1, 2: (-inf, inf)
1, 3: (-inf, inf)
1, 4: (-172, inf)
2, 3: (-inf, inf)
2, 4: (-inf, inf)
3, 4: (-inf, 1355)
f: 117/30
```

Those chronicles are the discriminant chronicles of the positive dataset ***d214lbbb_H141*** in comparison to the negative dataset ***d214pvc*** which occurs in at least 80% of the sequences of the positive dataset and 5 times in the positive than in the negative.

For example, for the first chronicle, `{["qrs[abnormal]", "qrs[abnormal]", "qrs[abnormal]"]}` corresponds to the multiset of the chronicle. The events are separated by commas. The line `0, 1: (686, 881)` corresponds to a temporal constraint of the chronicle. `0` and `1` correspond to indices in the multiset, it is so a temporal constraint between the first and the second occurrences of `qrs[abnormal]`. The temporal interval is defined by `(686, 881)` what means that the temporal constraint is `"qrs[abnormal]" [686, 881] "qrs[abnormal]"`. Finally, `f:125/20` corresponds to a positive support of 125 and a negative support of 20.

Argument options

The DCM parameters are listed in the help of the executable. To print this help, use the parameter `--help` or simply run the executable without parameters.

```
Usage:  Extract input_file fmin [options]
Positional Options (required):
  -i [ --input_file ] arg input file containing dataset to mine (string)
                          - positive dataset if --disc is used
                          positional : input_file
  -f [ --fmin ] arg       minimal frequency threshold (number)
                          Number of sequences if >= 1 (support)
                          Percent of positive sequences number else
                          positional fmin
```

```
General Options:
  --help                    Display this help message
  -d [ --disc ] arg         Extract discriminant chronicles using this fil
                            negative dataset
  -u [ --IBM ]              Use IBM format for files instead of sequence p
                            line
  --mincs arg               Minimum size of extracted chronicles
  --maxcs arg               Maximum size of extracted chronicles
  -c [ --close ]            Extract frequent closed chronicles or discrimi
                            chronicles from closed multisets if --disc is
  -j [ --json ]             Output format is json instead of plain text
  -v [ --verbose ]          The program will speak

Discriminant chronicles Options:
  -g [ --gmin ] arg         Minimal growth threshold
                            default : 2

Frequent chronicles Options:
  -a [ --all_different ]    Extract chronicles with multisets containing a
                            one occurrence of an event
  -w [ --cwm ] arg          Define the maximal windows size for temporal
                            constraints
  -n [ --not_calc_freq ]    If used, doesn't calculate exact frequency it'
                            bigger than fmin
                            ignored if --close is used
```

Appendix B

ATC descriptions

This appendix is dedicated to the event identifiers used in chapter 6.

The first table, Table B.1, associates the different ATC codes and their meanings. The column ATC03 class corresponds to the meaning of the first 3 characters of the ATC code.

The second table, Table B.2, associates the event identifiers in the chronicles presented in chapter 6 and their occurrences in the french drug register. it is worth to notice that the number of drugs represented by those identifiers is quite heterogeneous. Where brand-name identifiers only refer to one specif drug, the others refer to about ten drugs. The paracetamol is here an exception as it refers to a great number of drugs.

ATC	ATC03 class	name
B01AC06	anti-thrombotic agents	acetylsalicylic acid
N02BE01	analgesics	paracetamol
N03AF01	anti-epileptics	carbamazepine
N03AG01	anti-epileptics	valproic acid
N03AX09	anti-epileptics	lamotrigine
N03AX11	anti-epileptics	topiramate
N03AX14	anti-epileptics	levetiracetam

Table B.1: Description of the ATC codes used in chapter 6.

ATC	speciality group	GRS	occurrences
B01AC06			ASPIRINE UPSA 325MG GELULE 28
			ASPIRINE PROTECT 100 MG CPR GASTRORESISTANT 90
			ASPIRINE PROTECT 100 MG CPR GASTRORESISTANT 30
			CARDIOSOLUPSAN 100MG PDR ORALE 28
			KARDEGIC 75MG PDR ORALE SACHET 30/153,45
			ASPIRINE PROTECT 300MG CPR 30
			KARDEGIC 160MG PDR ORALE SACHET 30
			KARDEGIC 300MG PDR ORALE SACHET 30
			CLARAGINE 300 MG CPR 30
			CLARAGINE 300 MG CPR 30
N03AG01	G	438	VALPROATE DE SODIUM QUALIMED LP 500MG CPR 30
			VALPROATE DE SODIUM SANDOZ LP 500MG CPR 30
			VALPROATE DE SODIUM BIOGARAN LP 500MG CPR 30
			VALPROATE DE SODIUM TEVA LP 500MG CPR 30
			VALPROATE DE SODIUM SANDOZ 500 MG CPR SEC 30
			VALPROATE DE SODIUM RPG LP 500MG CPR 30
			VALPROATE DE SODIUM TEVA SANTE LP 500 MG CPR SEC 30
			VALPROATE DE SODIUM TEVA SANTE LP 500MG CPR 30

			VALPROATE DE SODIUM EG LP 500MG CPR 30
			VALPROATE DE SODIUM MERCK LP 500MG CPR 30
			VALPROATE DE SODIUM ZENTIVA LP 500MG CPR 30
			VALPROATE DE SODIUM ARROW 500 MG CPR SEC 30
			VALPROATE DE SODIUM ALTER LP 500MG CPR 30
N03AG01	R	438	DEPAKINE CHRONO 500MG CPR 30
N03AX09	R	431	LAMICTAL 100MG CPR DISP 30
N02BE01			PARACETAMOL NOR 500 MG CPR 16
			PARACETAMOL SANDOZ 500 MG CPR EFF 16
			PARACETAMOL BGR 500 MG CPR 16
			PARACETAMOL RPG 500MG PDR SAC 12
			PARACETAMOL QUALIMED 500MG CPR 16
			BRILIVO 500 MG CPR 16
			PARACETAMOL GNR 500 MG CPR 16
			PARACETAMOL RANBAXY 500 MG GELULE 16
			PARACETAMOL RATIOPHARM 500 MG GELULE 16
			PARACETAMOL RATIOPHARM 1 000 MG CPR EFF SEC 8
			DOLIPRANE 150MG PDR ORALE SACHET 12/0,81 G
			DOLIPRANE 100MG PDR ORALE SACHET 12/0,54 G
			PARACETAMOL ZYDUS 1 G CPR 8
			PARACETAMOL EVOLUGEN 500 MG GELULE 16
			DAFALGAN 500MG GELULE 16
			PARACETAMOL RATIOPHARM 500 MG CPR EFF 16
			PARACETAMOL SANDOZ 500MG GELULE 16
			PARACETAMOL ARROW 1G CPR EFF SEC 8
			PARACETAMOL MYLAN 1G CPR EFF SEC 8
			DAFALGAN 500 MG CPR SEC 16
			PARACETAMOL TEVA 500MG CPR EFF 16
			DAFALGAN PEDIATRIQUE 3 % SOL BUV 1/90 ML
			DOLIPRANE 200MG PDR ORALE SACHET 12/1,08 G
			PARACETAMOL ARROW 500 MG CPR 16
			DOLKO 1G CPR 8
			PARACETAMOL ALMUS 1G CPR 8
			PARACETAMOL SANDOZ 1 G CPR SEC 8
			PARACETAMOL TEVA 1 G CPR EFF SEC 8
			PARACETAMOL EG 500MG CPR 16
			DOLIPRANE 2,4% SS SUCRE SUSP BUV 1/100 ML
			DOLIPRANE 300MG SUPPO 10
			DOLIPRANE 200MG SUPPO 10
			PARACETAMOL EG 1 G CPR 8
			PARACETAMOL BIOGARAN 1G CPR 8
			PARACETAMOL TEVA 500MG CPR 16
			PARACETAMOL EG 1G CPR EFF SEC 8
			PARACETAMOL MYLAN 500MG GELULE 16
			PARACETAMOL TEVA 1G CPR 8
			PARACETAMOL BIOGARAN 500MG CPR EFF 16
			DOLKO 60MG/2ML SOL BUV FL 1/90 ML
			PARACETAMOL TEVA SANTE 1000 MG CPR SEC 8
			DOLSTIC 100 MG/ML SOL BUV 1/60 ML
			PARACETAMOL GNR 1 G CPR SEC 8
			EFFERALGAN 500MG CPR 16
			PARACETAMOL BIOGARAN 500MG GELULE 16
			DOLIPRANE 1G AD SUPPO 8
			DAFALGAN 600MG SUPPO 10
			CLARADOL 500MG CPR EFF 16
			DOLKO 500MG PDR ORALE SACHET 12
			DOLIPRANE 100MG SUPPO 10
			CLARADOL 500MG CPR SEC 16
			PARACETAMOL EG 500MG GELULE 16
			DOLKO 500MG CPR SEC 16
			PARACETAMOL WINTHROP 1 G CPR SEC 8

PARALYOC 250MG LYOPHILISAT ORAL 10
PARACETAMOL IREX 500MG CPR 16
PARACETAMOL BIOGARAN 500MG CPR 16
CLARADOL 120MG ENF NR CPR EFF 12
PARACETAMOL GNR 500 MG CPR 16
PARACETAMOL TEVA SANTE 500 MG CPR 16
DOLIPRANE 500MG PDR ORALE SACHET 12
DOLIPRANE 500MG CPR 16
PARACETAMOL G GAM 1G CPR EFF SEC 8
DOLKO 170MG SUPPO 10
DAFALGAN 150 MG PDRE EFFERV SOL BUVABLE SACHET 12
DAFALGAN 80 MG SUPPO 10
PARACETAMOL GNR 500MG CPR 16
DAFALGAN 80 MG PDRE EFFERV SOL BUVABLE SACHET 12
DAFALGAN 150 MG SUPPO 10
DOLIPRANE 1G PDR ORALE SACHET 8/5,4 G
DAFALGAN 300 MG SUPPO 10
PARACETAMOL ARROW 1G CPR 8
PARACETAMOL ZENTIVA 500 MG CPR 16
DOLIPRANE 1G CPR EFF SEC 8
EFFERALGAN 1G CPR EFF 8
DAFALGAN 1 G CPR 8
PARACETAMOL TEVA 300 MG PDRE SOL BUV SACHET 12
PARACETAMOL ZYDUS 500 MG GELULE 16
PARACETAMOL AHCL 1 G CPR 8
PARACETAMOL ARROW 500MG CPR EFF SEC 16
DOLIPRANE 1000 MG GELULE 8
PARACETAMOL TEVA 500 MG PDRE SOL BUV SACHET 12
DOLIPRANE 50MG PDR ORALE SACHET 12
EXPANDOL 500MG CPR 16
PARACETAMOL BIOGARAN 1G CPR EFF SEC 8
PARACETAMOL SANDOZ 500 MG CPR 16
PARACETAMOL RPG 500MG GELULE 16
PARACETAMOL G GAM 500MG CPR EFF 16
PARACETAMOL ACTAVIS 500 MG CPR 16
PARACETAMOL CRISTERS 500 MG GELULE 16
DOLSTIC 100 MG/ML SOL BUV 1/30 ML
DOLIPRANE 350MG SUPPO 10
PARACETAMOL ISOMED 1 000 MG CPR EFF SEC 8
PARACETAMOL ISOMED 1 000 MG CPR 8
PARACETAMOL ZYDUS 500MG CPR 16
PARACETAMOL RATIOPHARM 1G CPR 8
DOLIPRANE 150MG SUPPO 10
EFFERALGAN 500MG CPR EFF SEC 16
PARACETAMOL BIOGARAN 1 000 MG PDRE SOL BUV SACHET 8
PARACETAMOL RPG 500MG PDR ORALE 12
PARACETAMOL BAYER 500MG CPR EFF 16
DAFALGAN 1G CPR SEC 8
DOLIPRANE 125MG PDR ORALE SACHET 12
DAFALGAN 500 MG CPR 16
PARACETAMOL BIOGARAN 500 MG CPR 16
PARACETAMOL MYLAN 1 G CPR SEC 8
PARACETAMOL CRISTERS 1 G CPR 8
PARACETAMOL ALMUS 500MG CPR 16
PARACETAMOL ALTER 1G CPR 8
PARACETAMOL ACTAVIS 1000 MG CPR SEC 8
PARACETAMOL BIOGARAN 300 MG PDRE SOL BUV SACHET 12
PARACETAMOL ISOMED 500 MG CPR 16
PARACETAMOL BAYER 500MG CPR 16
PARACETAMOL ARROW 500MG CPR 16
DOLIPRANE 250MG PDR ORALE SACH 12

PARACETAMOL ALTER 1G GRANULES EFF SACH 8
PARALYOC 125MG LYOPHILISAT ORAL 10
PARACETAMOL ARROW 500 MG GELULE 16
PARACETAMOL RATIOPHARM 500MG CPR 16
DOLIPRANE 1G CPR 8
PARACETAMOL ARROW 500 MG GELULE 16
PARACETAMOL ISOMED 500 MG CPR SEC 16
PARACETAMOL RPG 1 G CPR SEC 8
PARACETAMOL RPG 500MG CPR EFF 16
DAFALGAN 250 MG PDRE SUSP BUVABLE SACHET 12
PARACETAMOL RPG 500MG CPR 16
EFFERALGAN PEDIATRIQUE 30MG/ML SOL BUV 1/90 ML
DOLIPRANE 170MG SUPPO 10
DOLIPRANE 80MG SUPPO 10
FEBRECTOL 500MG CPR 16
FEBRECTOL 250MG CPR DISPERSIBLE 12
PARALYOC 50MG LYOPHILISAT ORAL 10
PARACETAMOL MYLAN 500 MG GELULE 16
PARACETAMOL IVAX 500MG CPR 16
PARACETAMOL TEVA 500 MG GELULE 16
FEBRECTOL 125MG CPR DISPERSIBLE 12
DOLKO 80MG SUPPO 10
PARACETAMOL GENODEX 500MG CPR 16
BRILIVO 1000 MG CPR SEC 8
PARACETAMOL ACTAVIS 500 MG CPR SEC 16
PARACETAMOL QUALIMED 1 G CPR SEC 8
PARACETAMOL WINTHROP 1 G CPR EFF SEC 8
PARACETAMOL EG 500MG CPR 16
PARACETAMOL EG 500MG CPR EFF 16
PARACETAMOL CRISTERS 500 MG CPR 16
PARACETAMOL RANBAXY 1 G CPR 8
PARALYOC 500MG LYOPHILISAT ORAL 16
ALGODOL 500MG CPR 16
PARACETAMOL MYLAN 500MG CPR EFF 16
PARACETAMOL GRUNENTHAL 1 G CPR 8
PARACETAMOL SANDOZ 1G CPR EFF SEC 8
PARACETAMOL TEVA 1 000 MG PDRE SOL BUV SACHET 8
PARACETAMOL BIOGALENIQUE 500 MG GRAN SUSP BUV SACHET 12
PARACETAMOL MYLAN 500MG CPR 16
PARACETAMOL MERCK 500 MG GELULE 16
PARACETAMOL ARROW 300MG PDR ORALE SACH 12
PARACETAMOL BIOGARAN 500 MG PDRE SOL BUV SACHET 12
DOLIPRANE 500MG GELULE 16
DOLIPRANE 300MG PDR ORALE SACHET 12/1,62 G
PARACETAMOL ARROW 500MG PDR ORALE SACH 12
EXPANDOL 500 MG PDRE SOL BUV SACHET 12
DAFALGAN 1G CPR PELLICULE 8
PARACETAMOL ARROW 1G PDR ORALE SACH 8
GELUPRANE 500MG GELULE 16
BRILIVO 1000 MG CPR 8
PARACETAMOL SET 500 MG CPR 16
DOLIPRANE 500MG CPR EFF 16

Table B.2: Association of the event identifiers used in chapter 6 and the occurrences in the french register.

Appendix C

Publications

[1] Clément Gautrais, Yann Dauxais, and Maël Guilleme. Multi-plant photovoltaic energy forecasting challenge: Second place solution. In *Discovery Challenges co-located with European Conference on Machine Learning-Principle and Practice of Knowledge Discovery in Database*, 2017.

[2] Yann Dauxais, Thomas Guyet, David Gross-Amblard, and André Happe. Discriminant chronicles mining. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 234–244, 2017.

[3] Thomas Guyet, André Happe, and Yann Dauxais. Declarative sequential pattern mining of care pathways. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 261–266, 2017.

[4] Yann Dauxais, David Gross-Amblard, Thomas Guyet, and André Happe. Extraction de chroniques discriminantes. In *Extraction et Gestion des Connaissances (EGC)*, 2017.

[5] Valentin Lemière, Yann Dauxais, Patrice Boizumault, and Arnaud Lallouet. Mining frequent patterns using cp: a comparative study. In *CP Doctoral program*, 2016.

[6] Frédéric Balusson, Marie-Anne Botrel, Olivier Dameron, Yann Dauxais, Erwan Drezen, Alain Dupuy, Thomas Guyet, David Gross-Amblard, André Happe, Nolwenn Le Meur, et al. Peps: a platform for supporting studies in pharmaco-epidemiology using medico-administrative databases. In *International Congress on e-Health Research*, 2016.

[7] Yann Dauxais, David Gross-Amblard, Thomas Guyet, and André Happe. Chronicles mining in a database of drugs exposures. In *ECML Doctoral consortium*, 2015.

Bibliography

[Achar et al., 2012] Achar, A., Laxman, S., and Sastry, P. (2012). A unified view of the apriori-based algorithms for frequent episode discovery. *Knowledge and information systems*, 31(2):223–250.

[Agrawal and Srikant, 1995] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the International Conference on Data Engineering*, pages 3–14.

[Albert-Lorincz and Boulicaut, 2003] Albert-Lorincz, H. and Boulicaut, J.-F. (2003). Mining frequent sequential patterns under regular expressions: a highly adaptative strategy for pushing constraints. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 316–320. SIAM.

[Allen, 1984] Allen, J. F. (1984). Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154.

[Alvarez et al., 2013] Alvarez, M. R., Felix, P., and Carinena, P. (2013). Discovering metric temporal constraint networks on temporal databases. *Artificial Intelligence in Medicine*, 58(3):139 – 154.

[Andrews et al., 2003] Andrews, S., Tsochantaridis, I., and Hofmann, T. (2003). Support vector machines for multiple-instance learning. In *Advances in neural information processing systems*, pages 577–584.

[Ayres et al., 2002] Ayres, J., Flannick, J., Gehrke, J., and Yiu, T. (2002). Sequential pattern mining using a bitmap representation. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 429–435. ACM.

[Batal et al., 2013] Batal, I., Valizadegan, H., Cooper, G. F., and Hauskrecht, M. (2013). A temporal pattern mining approach for classifying electronic health record data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):63.

[Bay and Pazzani, 2001] Bay, S. D. and Pazzani, M. J. (2001). Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246.

[Beedkar and Gemulla, 2015] Beedkar, K. and Gemulla, R. (2015). Lash: Large-scale sequence mining with hierarchies. In *Proceedings of the International Conference on Management of Data*, pages 491–503.

[Blockeel et al., 2005] Blockeel, H., Page, D., and Srinivasan, A. (2005). Multi-instance tree learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 57–64.

[Borgelt, 2003] Borgelt, C. (2003). Efficient implementations of apriori and eclat. In *FIMI'03: Proceedings of the IEEE ICDM workshop on frequent itemset mining implementations*.

[Bornemann et al., 2016] Bornemann, L., Lecerf, J., and Papapetrou, P. (2016). Stife: A framework for feature-based classification of sequences of temporal intervals. In *International Conference on Discovery Science*, pages 85–100. Springer.

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and regression trees*. CRC press.

[Bringmann et al., 2011] Bringmann, B., Nijssen, S., and Zimmermann, A. (2011). Pattern-based classification: a unifying perspective. *arXiv preprint arXiv:1111.6191*.

[Buzmakov et al., 2013] Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S. O., Napoli, A., and Raïssi, C. (2013). The representation of sequential patterns and their projections within formal concept analysis. In *Workshop Notes for LML (PKDD)*.

[Chevaleyre and Zucker, 2001] Chevaleyre, Y. and Zucker, J.-D. (2001). Solving multiple-instance and multiple-part learning problems with decision trees and rule sets. application to the mutagenesis problem. In *Conference of the Canadian Society for Computational Studies of Intelligence*, pages 204–214.

[Cohen, 1995] Cohen, W. W. (1995). Fast effective rule induction. In *Proceedings of the International Conference on Machine Learning*, pages 115–123.

[Cordier et al., 2007] Cordier, M.-O., Le Guillou, X., Robin, S., Rozé, L., and Vidal, T. (2007). Distributed chronicles for on-line diagnosis of web services. In *18th International Workshop on Principles of Diagnosis*, pages 37–44.

[Cram et al., 2012] Cram, D., Mathern, B., and Mille, A. (2012). A complete chronicle discovery approach: application to activity analysis. *Expert Systems*, 29(4):321–346.

[De Smedt et al., 2017] De Smedt, J., Deeva, G., and De Weerdt, J. (2017). Behavioral constraint pattern-based sequence classification. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD*, pages 20–36.

[Dean and Ghemawat, 2008] Dean, J. and Ghemawat, S. (2008). Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113.

[Dechter et al., 1991] Dechter, R., Meiri, I., and Pearl, J. (1991). Temporal constraint networks. *Artificial intelligence*, 49:61–95.

[Dietterich et al., 1997] Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71.

[Dong and Li, 1999] Dong, G. and Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 43–52. ACM.

[Doran and Ray, 2014] Doran, G. and Ray, S. (2014). A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning*, 97(1):79–102.

[Dousson and Duong, 1999] Dousson, C. and Duong, T. V. (1999). Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *Proceedings of International Conference on Artificial Intelligence*, pages 620–626.

[Duivesteijn et al., 2016] Duivesteijn, W., Feelders, A. J., and Knobbe, A. (2016). Exceptional model mining. *Data Mining and Knowledge Discovery*, 30(1):47–98.

[Fabrègue et al., 2014] Fabrègue, M., Braud, A., Bringay, S., Grac, C., Le Ber, F., Levet, D., and Teisseire, M. (2014). Discriminant temporal patterns for linking physico-chemistry and biology in hydro-ecosystem assessment. *Ecological Informatics*, 24:210–221.

[Fabrègue et al., 2013] Fabrègue, M., Braud, A., Bringay, S., Le Ber, F., and Teisseire, M. (2013). Orderspan: Mining closed partially ordered patterns. In *International Symposium on Intelligent Data Analysis*, pages 186–197.

[Foulds and Frank, 2010] Foulds, J. and Frank, E. (2010). A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(01):1–25.

[Fradkin and Mörchen, 2015] Fradkin, D. and Mörchen, F. (2015). Mining sequential patterns for classification. *Knowledge and Information Systems*, 45(3):731–749.

[Garofalakis et al., 1999] Garofalakis, M. N., Rastogi, R., and Shim, K. (1999). Spirit: Sequential pattern mining with regular expression constraints. In *Proceedings of VLDB*, volume 99, pages 7–10.

[Gärtner et al., 2002] Gärtner, T., Flach, P. A., Kowalczyk, A., and Smola, A. J. (2002). Multi-instance kernels. In *Proceedings of ICML*, volume 2, pages 179–186.

[Giannotti et al., 2006] Giannotti, F., Nanni, M., and Pedreschi, D. (2006). Efficient mining of temporally annotated sequences. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 348–359. SIAM.

[Guyet et al., 2017] Guyet, T., Happe, A., and Dauxais, Y. (2017). Declarative sequential pattern mining of care pathways. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 261–266. Springer.

[Guyet and Quiniou, 2011] Guyet, T. and Quiniou, R. (2011). Extracting temporal patterns from interval-based sequences. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 1306–1311.

[Han et al., 2000a] Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., and Hsu, M.-C. (2000a). Freespan: frequent pattern-projected sequential pattern mining. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 355–359. ACM.

[Han et al., 2001] Han, J., Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings of the 17th international conference on data engineering*, pages 215–224.

[Han et al., 2000b] Han, J., Pei, J., and Yin, Y. (2000b). Mining frequent patterns without candidate generation. In *ACM sigmod record*, volume 29, pages 1–12. ACM.

[Herrera et al., 2011] Herrera, F., Carmona, C. J., González, P., and Del Jesus, M. J. (2011). An overview on subgroup discovery: foundations and applications. *Knowledge and information systems*, 29(3):495–525.

[Ho, 1995] Ho, T. K. (1995). Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition*, volume 1, pages 278–282. IEEE.

[Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844.

[Huang et al., 2012] Huang, Z., Lu, X., and Duan, H. (2012). On mining clinical pathway patterns from medical behaviors. *Artificial Intelligence in Medicine*, 56(1):35–50.

[Knobbe et al., 2008] Knobbe, A., Crémilleux, B., Fürnkranz, J., and Scholz, M. (2008). From local patterns to global models: the LeGo approach to data mining. *LeGo*, 8:1–16.

[Lakshmanan et al., 2013] Lakshmanan, G. T., Rozsnyai, S., and Wang, F. (2013). Investigating clinical care pathways correlated with outcomes. In *Business process management*, pages 323–338.

[Lattner et al., 2003] Lattner, A. D., Kim, S., Cervone, G., and Grefenstette, J. J. (2003). Experimental comparison of symbolic learning programs for the classification of gene network topology models. *Center for Computing Technologies–TZI*, 2:1.

[Li et al., 2008] Li, H., Wang, Y., Zhang, D., Zhang, M., and Chang, E. Y. (2008). Pfp: parallel fp-growth for query recommendation. In *Proceedings of the conference on Recommender systems*, pages 107–114. ACM.

[Lipton, 2016] Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.

[Lipton, 2017] Lipton, Z. C. (2017). The doctor just won't accept that! *arXiv preprint arXiv:1711.08037*.

[Lou et al., 2012] Lou, Y., Caruana, R., and Gehrke, J. (2012). Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158.

[Luo and Chung, 2008] Luo, C. and Chung, S. M. (2008). A scalable algorithm for mining maximal frequent sequences using a sample. *Knowledge and Information Systems*, 15(2):149–179.

[Ma and Liu, 1998] Ma, B. L. W. H. Y. and Liu, B. (1998). Integrating classification and association rule mining. In *Proceedings of the fourth international conference on knowledge discovery and data mining*.

[Mabroukeh and Ezeife, 2010] Mabroukeh, N. R. and Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Journal of Computing Survey*, 43(1):1–41.

[Mannila et al., 1997] Mannila, H., Toivonen, H., and Inkeri Verkamo, A. (1997). Discovery of frequent episodes in event sequences. *Data mining and knowledge discovery*, 1(3):259–289.

[Mäntyjärvi et al., 2004] Mäntyjärvi, J., Himberg, J., Kangas, P., Tuomela, U., and Huuskonen, P. (2004). Sensor signal data set for exploring context recognition of mobile devices. In *Proc. of 2nd Int. Conf. on Pervasive Computing (PERVASIVE 2004)*, pages 18–23.

[Masseglia et al., 1998] Masseglia, F., Cathala, F., and Poncelet, P. (1998). The psp approach for mining sequential patterns. *Principles of Data Mining and Knowledge Discovery*, pages 176–184.

[Masseglia et al., 2004] Masseglia, F., Teisseire, M., and Poncelet, P. (2004). Recherche des motifs séquentiels. *Revue Ingénierie des Systemes d'Information (ISI)*, 9(3-4):183–210.

[Mooney and Roddick, 2013] Mooney, C. H. and Roddick, J. F. (2013). Sequential pattern mining – approaches and algorithms. *ACM Journal of Computing Survey*, 45(2):1–39.

[Morin and Debar, 2003] Morin, B. and Debar, H. (2003). Correlation of intrusion symptoms: an application of chronicles. In *International Workshop on Recent Advances in Intrusion Detection*, pages 94–112. Springer.

[Moskovitch and Shahar, 2015] Moskovitch, R. and Shahar, Y. (2015). Fast time intervals mining using the transitivity of temporal relations. *Knowledge and Information Systems*, 42(1):21–48.

[Moulis et al., 2015] Moulis, G., Lapeyre-Mestre, M., Palmaro, A., Pugnet, G., Montastruc, J.-L., and Sailler, L. (2015). French health insurance databases: What interest for medical research? *La Revue de Médecine Interne*, 36(6):411–417.

[Novak et al., 2009] Novak, P. K., Lavrač, N., and Webb, G. I. (2009). Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, 10:377–403.

[Orlando et al., 2004] Orlando, S., Perego, R., and Silvestri, C. (2004). A new algorithm for gap constrained sequence mining. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 540–547. ACM.

[Otaki et al., 2014] Otaki, K., Ikeda, M., and Yamamoto, A. (2014). Pattern structures for understanding episode patterns. In *CLA*, pages 47–58.

[Pagallo and Haussler, 1990] Pagallo, G. and Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine learning*, 5(1):71–99.

[Papapetrou et al., 2005] Papapetrou, P., Kollios, G., Sclaroff, S., and Gunopulos, D. (2005). Discovering frequent arrangements of temporal intervals. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE.

[Pasquier et al., 1999] Pasquier, N., Bastide, Y., Taouil, R., and Lakhal, L. (1999). Discovering frequent closed itemsets for association rules. In *International Conference on Database Theory*, pages 398–416.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pei and Han, 2002] Pei, J. and Han, J. (2002). Constrained frequent pattern mining: a pattern-growth view. *ACM SIGKDD Explorations Newsletter*, 4(1):31–39.

[Pei et al., 2002] Pei, J., Han, J., and Wang, W. (2002). Mining sequential patterns with constraints in large databases. In *Proceedings of the international conference on Information and knowledge management*, pages 18–25.

[Polard et al., 2015] Polard, E., Nowak, E., Happe, A., Biraben, A., and Oger, E. (2015). Brand name to generic substitution of antiepileptic drugs does not lead to seizure-related hospitalization: a population-based case-crossover study. *Pharmacoepidemiology and drug safety*, 24(11):1161–1169.

[Quiniou et al., 2001] Quiniou, R., Cordier, M., Carrault, G., and Wang, F. (2001). Application of ILP to cardiac arrhythmia characterization for chronicle recognition. In *Proceedings of International Conference on Inductive Logic Programming*, pages 220–227.

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

[Quinlan, 1987] Quinlan, J. R. (1987). Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234.

[Quinlan, 1993] Quinlan, J. R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, 38.

[Salle et al., 2009] Salle, P., Bringay, S., and Teisseire, M. (2009). Mining discriminant sequential patterns for aging brain. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 365–369. Springer.

[Santisteban and Tejada-Cárcamo, 2015] Santisteban, J. and Tejada-Cárcamo, J. (2015). Unilateral jaccard similarity coefficient. In *GSB@ SIGIR*, pages 23–27.

[Srikant and Agrawal, 1996] Srikant, R. and Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Advances in Database Technology—EDBT'96*, pages 1–17.

[Starner et al., 1998] Starner, T., Weaver, J., and Pentland, A. (1998). Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1371–1375.

[Subias et al., 2014] Subias, A., Travé-Massuyès, L., and Le Corronc, E. (2014). Learning chronicles signing multiple scenario instances. *IFAC Proceedings Volumes*, 47(3):10397–10402.

[Uno et al., 2004] Uno, T., Kiyomi, M., and Arimura, H. (2004). LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets. In *FIMI*, volume 126.

[Vásquez et al., 2017] Vásquez, J., Travé-Massuyès, L., Subias, A., and Jimenez, F. (2017). Enhanced chronicle learning for process supervision. *IFAC-PapersOnLine*, 50(1):5035–5040.

[Weiss and Indurkhya, 1991] Weiss, S. M. and Indurkhya, N. (1991). Reduced complexity rule induction. In *IJCAI*, pages 678–684.

[Xing et al., 2010] Xing, Z., Pei, J., and Keogh, E. (2010). A brief survey on sequence classification. *ACM Sigkdd Explorations Newsletter*, 12(1):40–48.

[Yang and Kitsuregawa, 2005] Yang, Z. and Kitsuregawa, M. (2005). Lapin-spam: An improved algorithm for mining sequential pattern. In *Data Engineering Workshops, 2005. 21st International Conference on*, pages 1222–1222.

[Yang et al., 2005] Yang, Z., Wang, Y., and Kitsuregawa, M. (2005). Lapin: Effective sequential pattern mining algorithms by last position induction.

[Zaharia et al., 2010] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. (2010). Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10.

[Zaki, 2000] Zaki, M. J. (2000). Sequence mining in categorical domains: incorporating constraints. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 422–429.

[Zaki, 2001] Zaki, M. J. (2001). Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1):31–60.

[Zhang et al., 2001] Zhang, M., Kao, B., Yip, C.-L., and Cheung, D. (2001). A gsp-based efficient algorithm for mining frequent sequences. In *Proc. of IC-AI*, pages 497–503.

Résumé

De nombreuses données sont enregistrées dans le cadre d'applications variées et leur analyse est un challenge abordé par de nombreuses études. Parmi ces différentes applications, cette thèse est motivée par l'analyse de parcours patients pour mener des études de pharmaco-épidémiologie. La pharmaco-épidémiologie est l'étude des usages et effets de produits de santé au sein de populations définies. Le but est donc d'automatiser ce type d'étude en analysant des données.

Parmi les méthodes d'analyses de données, les approches d'extraction de motifs extraient des descriptions de comportements, appelées motifs, caractérisant ces données. L'intérêt principal de telles approches est de donner un aperçu des comportements décrivant les données. Dans cette thèse, nous nous intéressons à l'extraction de motifs temporels discriminants au sein de séquences temporelles, c'est-à-dire une liste d'évènements datés. Les motifs temporels sont des motifs représentant des comportements par leur dimension temporelle. Les motifs discriminants sont des motifs représentant les comportements apparaissant uniquement pour une sous-population bien définie. Alors que les motifs temporels sont essentiels pour décrire des données temporelles et que les motifs discriminants le sont pour décrire des différences de comportement, les motifs temporels discriminants ne sont que peu étudiés. Dans cette thèse, le modèle de chronique discriminante est proposé pour combler le manque d'approches d'extraction de motifs temporels discriminants. Une chronique est un motif temporel représentable sous forme de graphe dont les nœuds sont des évènements et les arêtes sont des contraintes temporelles numériques. Le modèle de chronique a été choisi pour son expressivité concernant la dimension temporelle. Les chroniques discriminantes sont, de ce fait, les seuls motifs temporels discriminants représentant numériquement l'information temporelle.

Les contributions de cette thèse sont : (i) un algorithme d'extraction de chroniques discriminantes (*DCM*), (ii) l'étude de l'interprétabilité du modèle de chronique au travers de sa généralisation et (iii) l'application de *DCM* sur des données de pharmaco-épidémiologie.

L'algorithme *DCM* est dédié à l'extraction de chroniques discriminantes et basé sur l'algorithme d'extraction de règles numériques $Ripper_k$. Utiliser $Ripper_k$ permet de tirer avantage de son efficacité et de son heuristique incomplète évitant la génération de motifs redondants. La généralisation de cet algorithme permet de remplacer $Ripper_k$ par n'importe quel algorithme de machine learning. Les motifs extraits ne sont donc plus forcément des chroniques mais une forme généralisée de celles-ci. Un algorithme de machine learning plus expressif extrait des chroniques généralisées plus expressives mais impacte négativement leur interprétabilité. Le compromis entre ce gain en expressivité, évalué au travers de la précision de classification, et cette perte d'interprétabilité, est comparé pour plusieurs types de chroniques généralisées. L'intérêt des chroniques discriminantes à représenter des comportements et l'efficacité de *DCM* est validée sur des données réelles et synthétiques dans le contexte de classification à base de motifs. Des chroniques ont finalement été extraites à partir des données de pharmaco-epidemiologie et présentées aux cliniciens. Ces derniers ont validés l'intérêt de celles-ci pour décrire des comportements d'épidémiologie discriminants.

Abstract

Data are recorded for a wide range of applications and their analysis is a great challenge addressed by many studies. Among these applications, this thesis was motivated by analyzing care pathway data to conduct pharmaco-epidemiological studies. Pharmaco-epidemiology is the study of the uses and effects of healthcare products in well-defined populations. The goal is then to automate this study by analyzing data.

Within the data analysis approaches, pattern mining approaches extract behavior descriptions, called patterns, characterizing the data. Patterns are often easily interpretable and give insights about hidden behaviors described by the data. In this thesis, we are interested in mining discriminant temporal patterns from temporal sequences, *i.e.* a list of timestamped events. Temporal patterns represent expressively behaviors through their temporal dimension. Discriminant patterns are suitably adapted for representing behaviors occurring specifically in small subsets of a whole population. Surprisingly, if temporal patterns are essential to describe timestamped data and discriminant patterns are crucial to identify alternative behaviors that differ from mainstream, discriminant temporal patterns received little attention up to now. In this thesis, the model of discriminant chronicles is proposed to address the lack of interest in discriminant temporal pattern mining approaches. A chronicle is a temporal pattern representable as a graph whose nodes are events and vertices are numerical temporal constraints. The chronicle model was chosen because of its high expressiveness when dealing with temporal sequences and also by its unique ability to describe numerically the temporal dimension among other discriminant pattern models.

The contribution of this thesis, centered on the discriminant chronicle model, is threefold: (i) a discriminant chronicle model mining algorithm (*DCM*), (ii) the study of the discriminant chronicle model interpretability through its generalization and (iii) the *DCM* application on a pharmaco-epidemiology case study.

The *DCM* algorithm is an efficient algorithm dedicated to extract discriminant chronicles and based on the $Ripper_k$ numerical rule learning algorithm. Using $Ripper_k$ allows to take advantage to its efficiency and its incomplete heuristic dedicated to avoid redundant patterns. The *DCM* generalization allows to swap $Ripper_k$ with alternative machine learning algorithms. The extracted patterns are not chronicles but a generalized form of chronicles. More expressive machine learning algorithms extract more expressive generalized chronicles but impact negatively their interpretability. The trade-off between this expressiveness gain, evaluated by classification accuracy, and this interpretability loss, is compared between several types of generalized chronicles. The interest of the discriminant chronicle model and the *DCM* efficiency is validated on synthetic and real datasets in pattern-based classification context. Finally, chronicles are extracted from a pharmaco-epidemiology dataset and presented to clinicians who validated them to be interesting to describe epidemiological behaviors.