Trabajo de Fin de Grado

Integración de Algoritmos de Minería de Secuencias Discriminativas en ProM

Discriminative Sequence Mining Algorithms Integration for ProM

Carmen María Santos García

La Laguna, 5 de septiembre de 2016

D. **Pedro Antonio Toledo Delgado**, con N.I.F. 45.725.874-B profesor Ayudante adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor

D. **Vanessa Muñoz Cruz**, con N.I.F. 78.698.687-R profesora Contratada Doctor adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutora

C E R T I F I C A (N)

Que la presente memoria titulada:

"Integración de Algoritmos de Minería de Secuencias Discriminativas en ProM".

ha sido realizada bajo su dirección por D. **Carmen María Santos García**, con N.I.F. 78.632.809-H.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 5 de septiembre de 2016.

Agradecimientos

A Pedro:

Por haberme soportado y seguir teniendo la paciencia necesaria para explicarme las cosas una docena de veces más.

A mi madre:

Porque siempre me ha ayudado en todo, porque gracias a ella he logrado ser lo que soy. Porque siempre me ha motivado para que consiga algo mejor.

A Airam:

Por siempre confiar en que acabaría la carrera, por mantener la confianza de que lo iba a lograr cuando hacía rato que yo la había perdido. Y, sobre todo, por ayudarme a que no me rinda nunca.

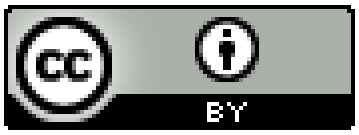
Al resto de profesores de la escuela:

Porque gracias a ellos he aprendido más de lo que esperaba, ya que no sólo me han enseñado informática, sino también he aprendido a comprender mejor el mundo que me rodea.

A mis amigos:

Por haberme escuchado hablar de la carrera una y otra vez.

Licencia



Resumen

La minería de datos y la minería de procesos son dos de las técnicas de análisis de datos más utilizadas hoy en día en el campo de la Inteligencia Empresarial. La Inteligencia Empresarial se basa en estas técnicas de análisis para correr menos riesgos a la hora de tomar de decisiones. Es en la combinación de estas técnicas en las que residen los nuevos avances en este campo.

El objetivo de este trabajo es la integración de dichas técnicas mediante la implementación de algoritmos de minería de secuencias discriminativas y su integración en la herramienta de minería de procesos ProM.

Con esto se consigue unir las diferentes técnicas usadas en la Inteligencia Empresarial (La minería de secuencias y la minería de procesos).

Para realizar esta tarea se han utilizado diferentes variantes de un algoritmo de minería de secuencias frecuentes denominado BIDE. Con estas variantes se ha implementado un plugin que se incorpora como herramienta de minería de procesos, tal cual hace ProM.

El resultado es un plugin para ProM que permite realizar minería de secuencias discriminativas a partir de logs de eventos, pudiendo seleccionar entre distintas variantes del algoritmo, y permitiendo la visualización de los resultados.

Palabras clave: Minería de Patrones, Minería de Procesos, ProM, secuencias discriminativas.

Abstract

Data mining and Process Mining are two of the most currentlly used data analysis techniques in the Business Intelligence field. Business Intelligence is based on these analysis techniques to take fewer risks when making decisions. It is in the combination of these techniques in which reside the new developments in this field.

The goal of this work is the integration of these techniques implementing discriminative sequence mining algorithms and integrating them into the ProM tool.

This way it is possible to unify different techniques used in Business Intelligence in a unique tool and perspective (sequence and processes mining).

To perform this task we have used different variants of a frequent sequence mining algorithm named BIDE. These variants have been implemented in plugin which is was incorporated in the process mining tool ProM.

The result is a plugin for ProM that allows to do discriminative sequence mining from event logs, selecting different variant of the base algortihm, and offering the possibility of result visualization.

Keywords: Pattern Mining, Process Mining, ProM, discriminative sequences.

Índice General

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

En la actualidad, todo nuestro sistema social y profesional está basado en procesos. Los servicios de conducción y funcionalidades internas de las empresas, organismos gubernamentales y organizaciones de todo el mundo se basan en procesos.

Si bien hay un montón de sistemas disponibles para el apoyo a la ejecución de tales procesos, las prácticas actuales para el seguimiento y análisis de esta ejecución todavía deja mucho que desear. La minería de procesos es una técnica capaz de llenar ese vacío, proporcionando medios revolucionarios para el análisis y seguimiento de los procesos de la vida real. Esta técnica se basa en la extracción de conocimiento sobre un proceso a partir de los registros de ejecución de estos. Se trata por lo general de tomar como entrada un conjunto de secuencias de eventos obtenidos a partir de la repetición de la ejecución de un determinado proceso y tratar de obtener a partir de él la representación explícita y formal de dicho proceso, utilizando para ello algún lenguaje de modelado de workflows, como por ejemplo redes de Petri en los casos más tradicionales.

Dada la extrema agresividad del mercado en términos de competencia y calidad, queda claro que es necesario el análisis de los datos. Para ello se usa la minería de datos y la minería de procesos.

La minería de datos es un campo de la estadística y las "ciencias de la computación" referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos.

Los algoritmos de minería de datos usados se clasifican en supervisados, que predicen un dato desconocido a partir de otros conocidos, o no supervisados, que descubren patrones y tendencias en los datos. En los algoritmos no supervisados se produce el agrupamiento de los datos según algún determinado criterio normalmente basado en una medida estadística de similitud. Dentre de

este ámbito no supervisado, se conoce como minería de patrones al análisis de conjuntos de items, agrupando en subsconjuntos aquellos que se puede observar que aparecen más frecuentemente en los datos originales. Estas aproximaciones también han sido extendidas a otras estructuras de datos más complejas como secuencias, secuencias generalizadas, grafos, etc..

Uno de los algoritmos de minería de itemsets frecuentes más conocidos es el algoritmo apriori (Agrawal & Srikant, 1994), el cual establece una forma sistemática de recorrer el conjunto de todos los posibles itemsets frecuentes, a partiendo de los conjuntos de ítems dispuestos individualmente y combinándolos posteriormente, de manera que la frecuencia de aparición de los nuevos conjuntos encontrados será menor con cada iteración. Originalmente la publicación en la que se propuso buscaba conjuntos de enlaces a páginas que aparecen frecuentemente juntos en varias visitas. En el caso de la minería de secuencias frecuentes uno de los algoritmos más conocidos se denomina BIDE, y será explicado con más detalle más adelante.

Además, dentro del ámbito de problemas supervisados de minería de datos, también es posible encontrar técnicas de minería de patrones. Concretamente, en algunas ocasiones, los conjuntos de ítems, secuencias o grafos pueden estar asociados a una etiqueta que determinaría la clase a la que pertenecen. El problema a resolver en esta ocasión sería la extracción de patrones discriminativos, en lugar de patrones frecuentes como en el caso anterior. Se trata, por tanto, de obtener aquellos subconjuntos de ítems, subsecuencias o subgrafos que mejor discriminan las entradas de datos que pertenecen a una clase frente a los que pertenecen a otra de las clases. Debe diferenciarse, al igual que ocurre con los clasificadores en las técnicas de minería de datos, que el caso de dos clases es la variante más sencilla para ser resuelta, teniendo muchas veces que encontrar extensiones o modificaciones de técnicas más sencillas para poder abordar el problema multiclase. Por otro lado, la forma en la que se mide el poder discriminativo de un patrón es un problema en sí mismo, relacionado con el estadístico a utilizar como métrica, y su repercusión en el rendimiento del algoritmo y en los resultados esperados.Para el trabajo tanto en el ámbito de la minería de patrones como en la minería de procesos, no obstante, existen diferentes herramientas que ayudan a trabajar en cada una de estos tipos de minería por separado, y que evitan tener que realizar implementaciones de los

distintos algoritmos desde cero. Por ejemplo en la minería de datos Weka o RapidMiner y en la Minería de Procesos Disco, Process Mining o ProM.

No obstante, se puede plantear la hipótesis de que se conseguirían mejoras significativas en las labores de análisis si pudieran combinarse técnicas de ambos campos para así alcanzar un mayor beneficio.

Es, por ello, que se pretende hacer una nueva aproximación en esta línea integrando la minería de secuencias discriminativas y la minería de procesos. Esta decisión se justifica de la siguiente manera. Se asume que ha sido posible obtener un modelo de proceso a partir de los algoritmos de minería de procesos correspondientes. Si se quisiera utilizar ese modelo para realizar predicciones sobre nuevas casos, sería conveniente tener características que pudieran diferenciar entre las variantes que han sido observadas anteriormente. Para ello podría ser útil utilizar minería de secuencias discriminativas sobre el mismo conjunto de datos original. De la fusión de ambos modelos, procesos y secuencias, se podría construir un mejor modelo predictivo. El primer paso, por tanto es la integración en una misma herramienta de ambos procedimientos de minería. Esto se logró integrando variantes en un plugin para una herramienta de minería de procesos, ProM. Toda esta aproximación ayuda a mejorar el conocimiento que se obtiene de todos los procesos que forman parte de la vida diaria.

I.1.1 Antecedentes y estado del Arte

ProM (que es la abreviatura del framework Process Mining) es un framework extensible que soporta una amplia variedad de técnicas de minería de procesos en forma de plug-ins. Proporciona una plataforma para los usuarios y desarrolladores de los algoritmos de minería de procesos que es fácil de usar y fácil de extender bajo licencia GPL. Los plugins son independientes de la plataforma y actualmente hay más de 600 plugins distribuidos en 120 paquetes.

Uno de los primeros algoritmos desarrollados en un plugin de Prom fue Alfa Miner (van der Aalst, Weijiters, & Maruster, 2003). Uno de los algoritmos más conocidos y actuales sería Genetic Miner (Medeiros, 2006).

Por otro lado, dentro de la minería de patrones se han desarrollado diferentes aproximaciones, según la estructura de los datos que se vayan a analizar:

- La minería de Items frecuentes se basa en examinar colecciones de elementos (items) y comprobar cuales se compran juntos frecuentemente. Un ejemplo muy común es la cesta de la compra, con el cuál se pretende conocer que elementos frecuentemente se compran juntos para así ubicarlos juntos o hacer ofertas.

  Compra 1: {verduras, carne, champú}

  Compra 2: {papas, carne, verduras, refresco}

  Compra 3: {carne, pasta, verduras}

  Según el ejemplo anterior la carne y las verduras se compran juntos frecuentemente, así que se podrían ubicar cerca o hacer una oferta conjunta.

- La minería de secuencias frecuentes consiste en encontrar patrones estadísticamente relevantes en colecciones de datos que están representados de forma secuencial. Puesto que aparecen con mucha frecuencia en datos de escenarios reales, esta técnica constituye uno de los métodos más conocidos de descubrimiento de patrones. En el siguiente ejemplo, se tiene diferentes secuencias asociadas a una clase (A o B):

  A: (1,3)(1,2)(5)(2,4)

  B: (2,4)(5)(1,4)(1,3)(1)

  A: (1,4)(2,3)(5)(2,4)

En el ejemplo anterior, un caso de patrón frecuente sería (2)(5), ya que se repite en todas las secuencias, pero no sería discriminativo ya que no sirve para diferenciar a las secuencias de tipo A de las secuencias de tipo B. Mientras que por ejemplo el patrón (1)(5) es frecuente y discriminativo, ya que solo está en las secuencias de tipo A.

- La minería de secuencias discriminativas consiste en encontrar patrones que sirvan para distinguir unas secuencias de otras, los datos sobre los que se aplican son secuenciales. En el siguiente ejemplo, se tiene diferentes secuencias asociadas a una clase (A o B):

  A: (1,3)(1,2)(5)(2,4)

  B: (2,4)(5)(1,4)(1,3)(1)

A: (1,4)(2,3)(5)(2,4)

Un caso de patrón discriminante sería (1)(5), ya que sirve para diferenciar a las secuencias de tipo A de las secuencias de tipo B.

- En la minería de grafos se conoce que una de sus principales tareas es encontrar patrones frecuentes dentro de un grafo.

En los ámbitos anteriores de la minería, existen varias métricas estadísticas útiles para evaluar el poder discriminativo que tiene una subsecuencia o un patrón, entre ellas destacan la Ganancia de información (IG), X^2 o la correlación.

Por otro lado, entorno al objetivo en que se centra el trabajo cabe destacar el precedente de Trabajo de Fin de Grado (Mattogno, 2013/14). Este trabajo tenía como objetivo desarrollar una herramienta que integrara las técnicas de la minería de datos y la minería de procesos. Para ello, construyó un prototipo de plugin Weka para la plataforma ProM. Este prototipo está compuesto por el grafo del proceso y una lista de caminos más frecuente. Dicho modelo facilita añadir información extra al flujo de procesos como los costes o las personas involucradas en las actividades, entre otras cosas.

I.1.2 Objetivos

El objetivo principal del Trabajo se ha basado en la realización de un plugin para el framework ProM [1] en el que se utiliza un algoritmo de Minería de Secuencias Discriminativas para enriquecer el conocimiento aportado por un modelo de proceso. Para lograr esto, es necesario realizar los siguientes puntos:

1. Seleccionar uno o varios algoritmos de minería de secuencias discriminativas.
2. Desarrollar los algoritmos anteriormente seleccionados.
3. Validar la implementación de los algoritmos.
4. Llevar a cabo la integración del algoritmo en ProM.
5. Crear la interfaz de visualización.

I.1.3 Tecnologías Utilizadas

Para el desarrollo de este TFG se han utilizado las tecnologías siguientes:

- El lenguaje de programación Java.
- El entorno de desarrollo Eclipse (eclipse, s.f.).
- El framework ProM (ProM, s.f.)
- La librería SPMF (Fournier).
- SVN Subversión (The Tortoise Team, s.f.).

Capítulo 2. Algoritmos de Minería de Secuencias Discriminativas

En este capítulo se definen las variantes del algoritmo BIDE (BIDE-Discriminative, BIDE-D y BIDE-DC) así como sus diferencias.

I.2.1 Elección del algoritmo

La elección de este algoritmo se hizo para poder continuar las líneas de Trabajos de Fin de Grado hechos en cursos anteriores que integraban la minería de secuencias y la minería de procesos a través de ProM

Como se explica en el artículo (Fradkin & Mörchen, 2014) es necesario obtener patrones que se puedan utilizar para clasificar problemas. En este artículo se definen varias variantes del algoritmo BIDE. Todas estas variantes ofrecen soluciones a problemas multiclases y las primeras que se definen son las que producen menos patrones y son más rápidas que BIDE. Es por esto que se ha elegido estas variantes como algoritmos de minería de secuencias a implementar.

Es un algoritmo eficiente y útil en la clasificación de datos que puede verse beneficiado al unirlo a la minería de procesos mediante la herramienta ProM.

I.2.2 Algoritmo BIDE

Un algoritmo de minería de secuencias frecuentes se utiliza para minar datos secuenciales obteniendo los patrones más repetidos entre los datos.

BIDE (BI-Directional Extension) (Jianyong Wang, 2004-03-30) es un algoritmo para el minado de secuencias frecuentes cerradas que hace uso del método pseudo projection para proyectar las bases de datos. Además, utiliza

un nuevo método, el BI-Directional Extension closure checking para determinar si una secuencia es cerrada y el método BackScan junto con la técnica de optimización ScanSkip para podar el espacio de búsqueda.

Su pseudocódigo es:

```
Require: Sequential Pattern P = {pi}, projected database D|P, minimun support µ
01: F – set of frequent closed patterns (global variable)
02: l = |P|
03: Ls = sStepFrequentItems(P,D|P,µ);
04: Li = iStepFrequentItems(P,D|P,µ);
05: if !(freqCheck(Ls,P)|| freqCheck(Li,P)) then
06:     if backscan(P,D,true) then
07:         F = F u P
08: for itemset p ∈ Ls do
09:     P' = p1,...,pl,p
10:     if backscan(P',D| P',false) then
11:         bide(P',D| P', µ);
12: for itemset p ∈ Li do
13:     P' = p1,...,pl,pl u p
14:     if backscan(P',D| P',false) then
15:         bide(P',D| P', µ);
16: return F
```

Si examinamos el pseudocódigo anterior observamos lo siguiente:

- F (línea 1), es la lista de patrones cerrados que se obtiene como resultado (línea 16).
- En la línea 3, se intenta extender el patrón P añadiendo un nuevo conjunto al final del patrón P, obteniendo un nuevo patrón de la siguiente forma: $P' = p_1 ... p_n p_{n+1}$. En el que p_{n+1} es la extensión del patrón.
- En la línea 4, se intenta extender el patrón P añadiendo un nuevo elemento al último conjunto del patrón P, obteniendo un nuevo patrón de la siguiente forma: $P' = p_1 ... p'_n$, siendo $p'_n = p_n$ u {nuevo ítem}.

- Las líneas 5, 6 y 7 comprueban las extensiones del patrón y en caso de que no sea extensible, es decir sea un patrón cerrado, se guarda en F.
- Entre las líneas 8-11 y 12-15 se realiza el mismo proceso, pero extendiendo el patrón de diferente manera, agregando un conjunto (Ls) o un ítem (Li). En caso de que el patrón se extienda se hace una llamada recursiva a BIDE(11 y 15) para examinar el nuevo patrón.

I.2.3 Variantes del algoritmo BIDE

Existen diferentes variantes del algoritmo BIDE.

I.2.3.1 BIDE-Discriminative

La variante BIDE-Discriminative usa selecciona los k patrones más discriminativos de entre todos los patrones frecuentes. Para ello se utiliza un valor conocido como poder discriminativo, en este caso la métrica estadística con la que se calcula es la Ganancia de Información.

A continuación, se muestra el pseudocódigo del Algoritmo BIDE-Discriminative:

Require: Sequential Pattern P = {p_i}, projected database D|P, minimun support μ, k – number of patterns to be selected

01: F – set of discriminative patterns (global variable)

02: dt = 0 – minimal threshold for discriminative score of a pattern (global variable)

03: if $d_{UB}(P) < dt$ then

04: return

05: l = |P|

06: Ls = sStepFrequentItems(P,D|P,μ);

07: Li = iStepFrequentItems(P,D|P,μ);

08: if d(P) ≥ dt then

09: if !(freqCheck(Ls,P)|| freqCheck(Li,P)) then

10: if backscan(P,D,true) then

11: F = F u P

12: if |F| > k then

13: F = F-argmin $_X \epsilon_F$ d(X)

```
14:                              dt = min X∈F d(X)
15: for itemset p ∈ Ls do
16:   P' = p1,...,pl,p
17:   if backscan(P',D| P',false) then
18:         BIDEDiscriminative(P',D| P', µ);
19: for itemset p ∈ Li do
20:   P' = p1,...,pl,pl u p
21:   if backscan(P',D| P',false) then
22:         BIDEDiscriminative(P',D| P', µ);
23: return F
```

Como se puede apreciar es muy parecido al algoritmo original, BIDE. Las diferencias son:

- Línea 2: dt establece el valor del poder discriminativo.
- Líneas 3 y 4: Se comprueba que el umbral superior(d_{UB}) de un patrón es mayor que el poder discriminativo. Si es mayor se sigue evaluando ese patrón, en caso contrario se descarta.
- Línea 8: Se comprueba que la ganancia de información del patrón sea superior al poder discriminativo.
- Líneas 12, 13 y 14: Tras haber guardado el nuevo patrón, se comprueba que si el tamaño de la lista de patrones(F) es mayor que los k patrones que se quieren obtener. En caso de que sea mayor, se elimina el patrón que tenga menor poder discriminativo y se actualiza dt al patrón de F con menor poder discriminativo.

Esta variante del algoritmo BIDE sólo se puede ejecutar con dos etiquetas de clase diferentes. Para un problema de dos etiquetas, el umbral superior de la Ganancia de Información es una función convexa. Lo cual implica que la IG no puede ser formulada, ya que la función no admite más elementos.

Para resolver esto se han conseguido dos

I.2.3.2 BIDE-D y BIDE-DC

La variante del BIDE-Discriminative sólo se puede ejecutar con dos etiquetas de clase diferentes ya que, para un problema de dos etiquetas, el umbral superior de la Ganancia de Información es una función convexa. Lo cual implica no puede ser formulada, ya que la función no admite más elementos (clases).

Para resolver esto en el artículo (Fradkin & Mörchen, 2014) se han desarrollado dos enfoques del BIDE-Discriminative para predecir patrones en problemas multiclases.

La variante BIDE-D ejecuta el conjunto de datos una vez igual que BIDE-Discriminative, pero redefinir las puntuaciones discriminantes y los límites superiores de los patrones para hacer máxima sobre todos los problemas binarios.

Mientras que, en la otra variante, BIDE-DC, se separan los datos de manera que se ejecuta |C| (número de clases distintas) veces tratándolo cada vez como un problema one-versus-rest. De esta manera, se producen k|C| patrones, ya que la ejecución de BIDE-Discriminative sobre cada una de las clases se obtienen los k patrones más discriminativos.

I.2.4 Punto de partida

El comienzo de la implementación fue a partir del algoritmo BIDE. La implementación de este algoritmo está en la librería (Fournier), de manera abierta.

Antes de empezar a programar las variantes era necesario adaptar todo lo que trabajaba con la entrada de datos. Esto es debido a que el algoritmo BIDE original no usa etiquetas de clase, mientras que sus variantes necesitan etiquetas de clase para la discriminación de secuencias.

I.2.5 Validación de la implementación

Este apartado trata la validación de la implementación hecha anteriormente. Para ello ha sido necesario simular con diferentes muestras de datos y analizar los resultados.

Esto se ha realizado con datos sintéticos. Todos los resultados muestran los 5 patrones más discriminativos que tienen un soporte mínimo de 2.

- BIDE-D: Primero se ha realizado una prueba con pocos datos, 4 secuencias (repartidas así 2,1,1), para asegurarnos que el algoritmo funciona bien.

  Los resultados están en B.1, y tarda 24 ms en realizar el algoritmo. Con 30 secuencias, repartidas de la siguiente manera 10, 12 y 8. Los resultados están en B.2 y tarda 43 ms.

- BIDE-DC: Al igual que en el caso anterior, primero se ha realizado una prueba con pocos datos, 4 secuencias (3 de una clase y una de la otra), para asegurarnos que el algoritmo funciona bien.

  Los resultados están en B.3, y tarda 70 ms en realizar el algoritmo.

  Con 30 secuencias, repartidas de la siguiente manera 10, 12 y 8. Los resultados están en B.4.

Capítulo 3. Integración

En este capítulo se explica la integración de los algoritmos de Minería de Secuencias Discrimantivas en ProM.

I.3.1 Interfaz de ProM

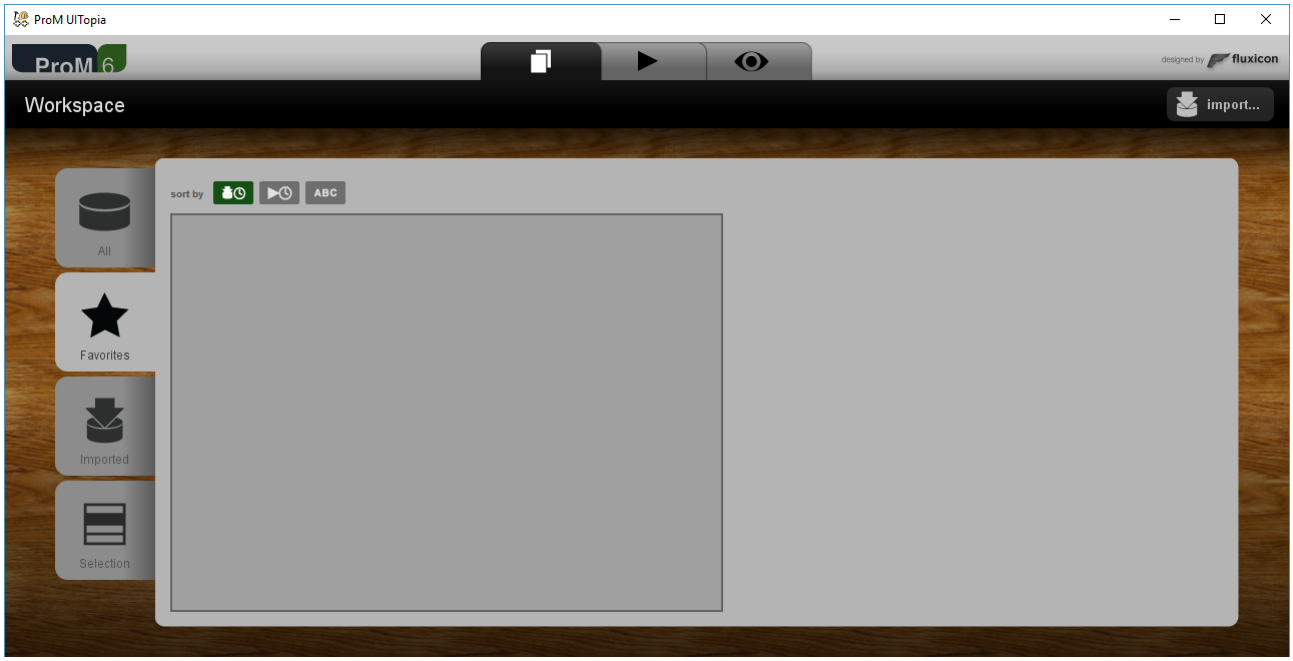


Ilustración 1: ProM.

Como se puede ver, al iniciar ProM 6 se ven tres pestañas, de derecha a izquierda son Workspace (espacio de trabajo), Action(acción) y View(visualización). En la primera pestaña, Workspace, se almacenan los datos con los que trabaja la herramienta. En la segunda, Action, se listan todos los plugins instalados, incluidos el desarrollado en este proyecto. Mientras que en la tercera se pueden visualizar los resultados.

I.3.2 Tipos de plugins

Como ya se ha mencionado anteriormente, el framework ProM permite incorporar algoritmos a través de plugins. Un plugin para ProM suele estar compuesto de varios plugins, según su uso, estos pueden ser:

- Plugins de procesamiento: es sobre estos plugins sobre los que recae la acción de arrancar el código del algoritmo que este incorporado.
- Plugins de visualización: son los que permiten una correcta visualización de los datos obtenidos del algoritmo.
- Plugins de importación: permiten importar los datos de entrada para la ejecución del algoritmo.
- Plugins de exportación: permiten exportar los resultados.

I.3.3 Desarrollo del plugin

A la hora de hacer un plugin para ProM hay que tener en cuenta varios puntos:

- Archivo de entrada: El archivo de los datos tiene que estar en formato Xlog para que ProM pueda aceptarlo. También existe la posibilidad de usar un archivo .csv y en ProM exportarlo a Xlog. Esta es la solución desarrollada en este proyecto.
- Incorporación del código: Esto requiere seguir una estructura básica descargable (ProM, s.f.) para todos los plugins e ir agregando anotaciones al código. Estas anotaciones indican a ProM qué el código es un plugin.
- Visualización: Es necesario crear una visualización de los datos obtenidos para poder observarlos adecuadamente.

I.3.4 Ejecución del Plugin

Para ejecutar el plugin hay que abrir el proyecto y ejecutar “ProM with UITopia.launch”. Este archivo lanza ProM con el plugin. Ahora es necesario importar el archivo de datos seleccionando el botón “Import” que se muestra en la siguiente imagen:

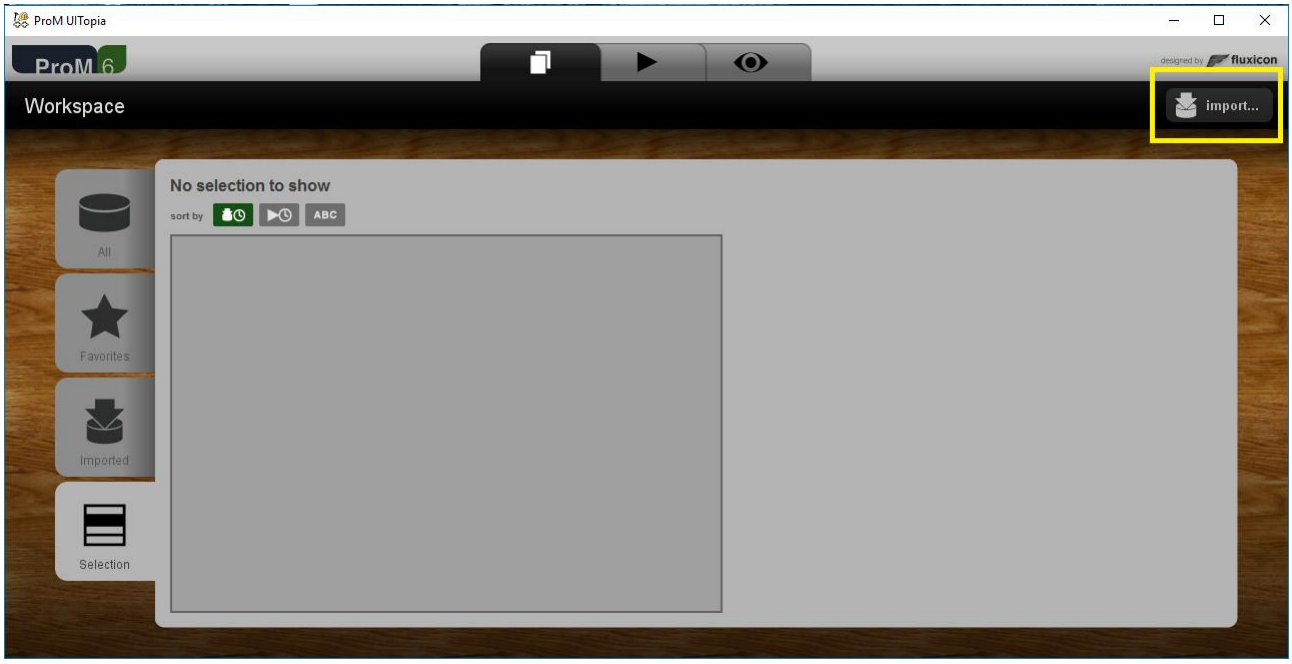


Ilustración 2: ProM. Botón "Import".

A continuación, clickamos en la pestaña de “Workspace”. En esta ocasión seleccionamos el archivo XES Event log, y clickamos en “use resource”, donde se muestra:

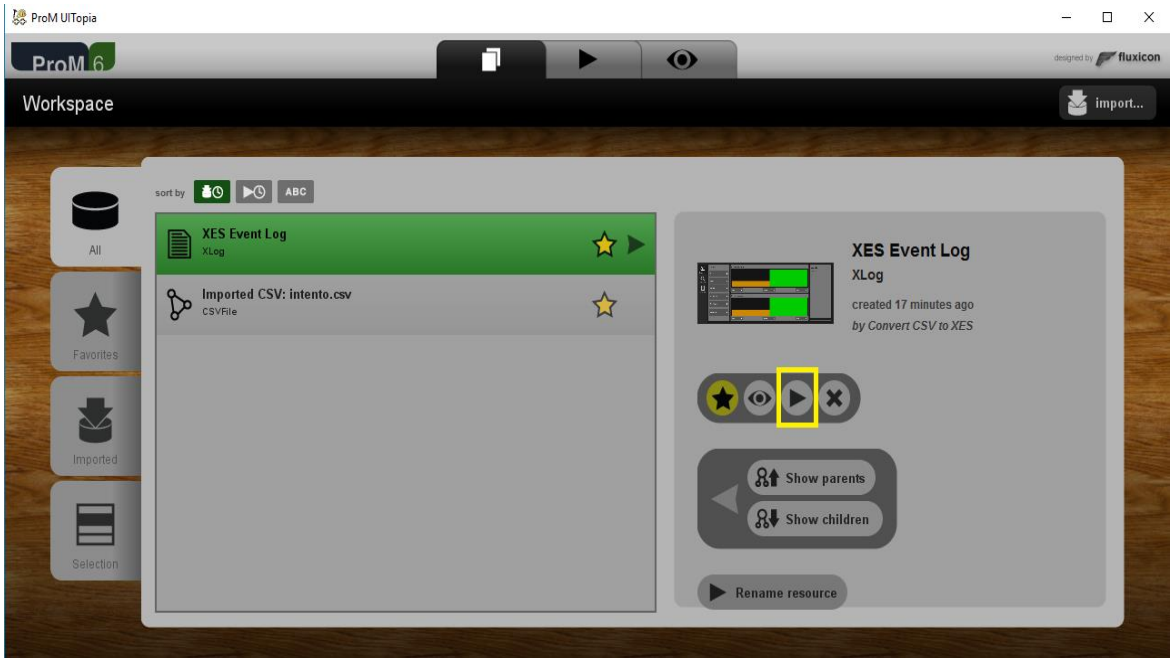


Ilustración 3: ProM. Uso de un recurso.

Abriéndose así:

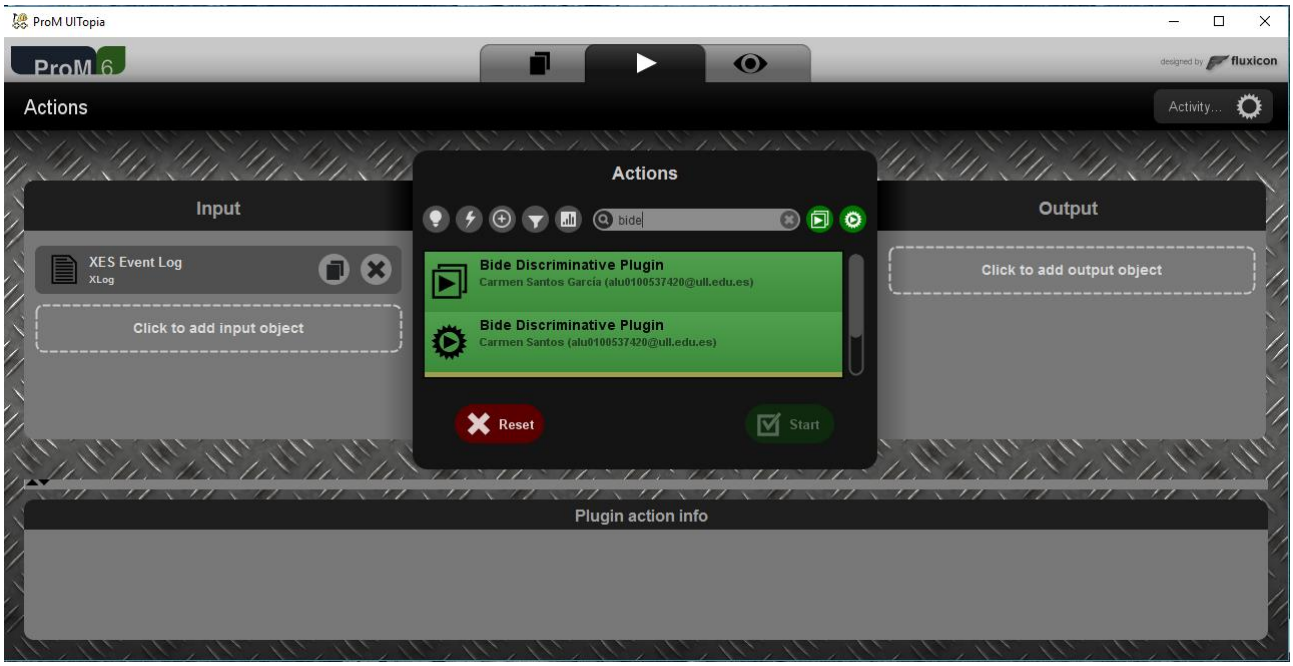


Ilustración 4: ProM. Actions.

Si en el cuadro de búsqueda se busca “bide” aparecen los plugins de BIDE que se han diseñado. Se elige el que se desea y se clicka en el botón “Start” produciendo así que se ejecute el plugin con el archivo XES Event log.

Al terminar, observamos la pestaña de “View” en la que se muestran los resultados.

Capítulo 4.
Conclusiones y líneas futuras

Al comenzar este trabajo, se pretendía dar un paso más en la combinación de técnicas de análisis de datos, para lograr obtener más información de los procesos que forman parte de la vida diaria.

Se desarrollaron dos variantes de un algoritmo de secuencias (BIDE), la variante BIDE-D y la variante BIDE-DC, la cual supone una ligera ventaja sobre la primera. Ambas variantes implementan dos enfoques distintos para afrontar la resolución de problemas de secuencias discriminativas multiclases.

Así mismo, se ha implementado un plugin para una herramienta de procesos que incorpora las variantes nombradas. Dichas variantes permiten obtener patrones discriminativos que pueden ayudan a diferenciar los procesos.

Tras haber logrado los objetivos y haber visto todo lo que puede llegar a ofrecer la integración de estas técnicas queda claro que el campo de investigación es aún muy amplio.

Es necesario seguir explorando está vía, para conocer el alcance que tiene. Como primera línea futura a seguir se debería probar este algoritmo de minería de secuencias con datos reales.

Otra línea sería implementar más algoritmos de minería de secuencias discriminativas e incorporarlos al plugin para así obtener mayores beneficios de esta combinación de técnicas.

Continuar implementando la combinación de estas técnicas (como la minería de datos y la minería de procesos) es el camino para optimizar los procesos empresariales.

Capítulo 5.
Summary and Conclusions

At first, it was intended to go one step further in combining data analysis techniques in order to obtain more information about our daily lives.

Two sequence algorithm variants were implemented, BIDE-D and BIDE-DC. The last one means a slight advantage over BIDE-D. Both variants have two different focus to face multi-class discriminative sequence problema solving.

In the same way, these variants were attached to a custom plugin for ProM. Them allow to obtain discriminative patterns which allow us to distinct process.

Having achieved our main goals and after we took in consideration all the integration of these techniques advantage is now clear for us that the research field is still very wide.

It is necessary to keep exploring these techniques. As first line to follow, tests with this sequence mining algorithm should be done with real data.

Another work line could be develop more discriminative sequences mining algorithms and incorporate it to the plugin in order to obtain greater benefits from this combination of techniques.

Continues implements of these techniques (such as data mining and processes mining) is the way to optimize business processes.

Capítulo 6. Presupuesto

Este capítulo detalla el presupuesto necesario para llevar a cabo este TFG y la justificación del mismo.

I.6.1 Presupuesto

Todo el software utilizado en el TFG son herramientas de software libre. Por lo que no aumentante el coste. En el presupuesto se incluye el ordenador y el trabajo del personal.

Referencia	Cantidad	Costes
Amortización de Equipo (Ordenador)	3/48	33,78€
Personal	300h	6900€
	Total:	6933,78€

Tabla 1: Presupuesto.

I.6.2 Justificación del presupuesto

Como se ha explicado anteriormente todo el software es libre por lo que solo se contabiliza la mano de obra y el ordenador.

En referencia al coste del ordenador se ha hecho una amortización de lo utilizado en este TFG, ya que previamente se disponía de él. Suponiendo que el tiempo total de uso sea de unos 3 meses, lo que sería un 6,25%. Al aplicar ese porcentaje al precio del ordenador (510 euros) que el total de 33.78 euros.

El coste del personal corresponde a una persona trabajando unas 300 horas a 23 euros/hora con lo que hacen un total de 6900 euros sin considerar ningún tipo de impuesto. Con lo que se obtiene un total de 6933,78€ de presupuesto final de este trabajo.

Capítulo 7. Anexo A: Datos

I.7.1 A.1. 4 secuencias

A : (1)(1 2 3)(1 3)(4)(3 6)
E : (1 4)(3)(2 3)(1 5)
A : (5 6)(1 2)(4 6)(3)(2)
I : (5)(7)(1 6)(3)(2)(3)

I.7.2 A.2. 30 secuencias

A : (4)(7)(1)(1 6)(7 9)(1)
E : (6)(1 2)(1)(4 9)(4)
I : (1)(6 6)(7 8)(2)(1 4)
A : (8 8)(2)(3 3 7)(1)
E : (5)(2 5 7)(1)(3 4 5)(3)
I : (1)(1 4)(2)(1 3)
A : (1 3)(2)(1)(2 7)(2 4)
E : (1)(4 5)(4 9)(1 2)
I : (3 4 4)(1)(2 5)
E : (6 8)(2)(1)(1 7)
I : (2 3)(3 4 8)(5 6)
A : (1)(9)(8 8)(1)(2 9)
E : (3 5)(3 4)(1)(1 8)
E : (5 5 7)(2 4)(8 9)(1 8)
A : (1)(8)(2 7 7)(2 5 6)(3 7)(9)
I : (2 3 9)(2 9 9)(1 6)
E : (3 5 9)(1 2)(2 9)
A : (5 7)(2 3)(2 6)(8)
I : (2)(1 4)(2 2)(8)(5 6 9)
A : (2 7 9)(1)(1 2)(3)
E : (1 9)(4 4)(3)(3 5 6)
E : (1)(1)(3 4 8)(2 5 6)(1)

A : (2)(2 6)(1)(6)(3)
I : (4)(2 9)(2 7)(2 4)
E : (1)(4)(4 7)(9)
A : (3 4)(3 9)(1 2)(7 7)
E : (4 6 7)(5 6 9)(1 7)
A : (1)(1)(1 7)(3 6)(3 8 9)
E : (1)(7)(1)(3 4 6)
I : (1 9)(1)(3 5)(3 3)(4)

Capítulo 8. Anexo B: Resultados

I.8.1 B.1: BIDE-D: 4 secuencias

```
----------FREQUENT SEQUENTIAL PATTERNS -------
L0
L1
pattern 1:  (6 )    support :  0,75 (3/4) --IG: 0.8112781244591328
L2
pattern 2:  (4 )(3 )    support :  0,75 (3/4) --IG: 0.8112781244591328
pattern 3:  (1 2 )(6 )    support :  0,5 (2/4) --IG: 1.0
L3
pattern 4:  (6 )(2 )(3 )    support :  0,5 (2/4) --IG: 0.5
pattern 5:  (1 2 )(4 )(3 )    support :  0,5 (2/4) --IG: 1.0
------------------------------- Patterns count : 5
```

I.8.2 B.2: BIDE-D: 30 secuencias

```
----------FREQUENT SEQUENTIAL PATTERNS -------
L0
L1
pattern 1:  (1 4 )    support :  0,1 (3/30) --IG: 0.21447985947595694
pattern 2:  (4 )    support :  0,66667 (20/30) --IG: 0.21957271413169133
L2
pattern 3:  (1 )(4 )    support :  0,36667 (11/30) --IG: 0.14528314229035988
L3
pattern 4:  (1 )(4 )(4 )    support :  0,1 (3/30) --IG: 0.14448434380562802
pattern 5:  (1 )(1 )(7 )    support :  0,1 (3/30) --IG: 0.17523196051238354
------------------------------- Patterns count : 5
```

I.8.3 B.3: BIDE-DC: 4 secuencias

```
c: A
 ----------FREQUENT SEQUENTIAL PATTERNS -------
  L0
  L1
  pattern 1:  (6 )    support :  0,75 (3/4) --IG: 0.31127812445913283
  L2
  pattern 2:  (4 )(3 )    support :  0,75 (3/4) --IG: 0.31127812445913283
  pattern 3:  (1 2 )(6 )    support :  0,5 (2/4) --IG: 1.0
  L3
  pattern 4:  (1 )(3 )(3 )    support :  0,75 (3/4) --IG: 0.31127812445913283
  pattern 5:  (1 2 )(4 )(3 )    support :  0,5 (2/4) --IG: 1.0
 -------------------------------- Patterns count : 5
c: E
 ----------FREQUENT SEQUENTIAL PATTERNS -------
  L0
  L1
  pattern 1:  (6 )    support :  0,75 (3/4) --IG: 0.8112781244591328
  L2
  L3
  pattern 2:  (6 )(2 )(3 )    support :  0,5 (2/4) --IG: 0.31127812445913283
 -------------------------------- Patterns count : 2
c: I
 ----------FREQUENT SEQUENTIAL PATTERNS -------
  L0
  L1
  L2
  pattern 1:  (4 )(3 )    support :  0,75 (3/4) --IG: 0.8112781244591328
  L3
  pattern 2:  (4 )(3 )(2 )    support :  0,5 (2/4) --IG: 0.31127812445913283
 -------------------------------- Patterns count : 2
```

I.8.4 B.4: BIDE-D: 30 secuencias

```
c: A
----------FREQUENT SEQUENTIAL PATTERNS -------
L0
L1
pattern 1:  (4 )    support :  0,66667 (20/30) --IG: 0.2179719978333251
L2
pattern 2:  (4 )(5 )    support :  0,2 (6/30) --IG: 0.1344008287335674
pattern 3:  (2 )(2 6 )    support :  0,1 (3/30) --IG: 0.17523196051238366
L3
pattern 4:  (2 )(2 6 )(3 )    support :  0,06667 (2/30) --IG: 0.11271663672563392
pattern 5:  (2 )(2 )(3 )    support :  0,1 (3/30) --IG: 0.17523196051238366
-------------------------------- Patterns count : 5
c: E
----------FREQUENT SEQUENTIAL PATTERNS -------
L0
L1
pattern 1:  (6 )    support :  0,75 (3/4) --IG: 0.8112781244591328
L2
L3
pattern 2:  (6 )(2 )(3 )    support :  0,5 (2/4) --IG: 0.31127812445913283
-------------------------------- Patterns count : 2
c: I
----------FREQUENT SEQUENTIAL PATTERNS -------
L0
```

```
L1
L2
pattern 1:  (4 )(3 )    support :  0,75 (3/4) --IG: 0.8112781244591328
L3
pattern 2:  (4 )(3 )(2 )    support :  0,5 (2/4) --IG: 0.31127812445913283
-------------------------------- Patterns count : 2
```

Capítulo 9. Bibliografía

[1] Arís, E. P. (2007). *La gestión tecnológica del conocimiento.* EDITUM.

[2] eclipse. (s.f.). *Eclipse.* Obtenido de https://eclipse.org/luna/

[3] Fournier, P. (s.f.). *Librería SPMF.* Obtenido de SPMF An Open-Source Data Mining Library: http://www.philippe-fournier-viger.com/spmf/index.php

[4] Fradkin, D., & Mörchen, F. (15 de Mayo de 2014). *SpringerLink.* Obtenido de http://link.springer.com/article/10.1007/s10115-014-0817-0

[5] Jianyong Wang, J. H. (2004-03-30). BIDE: Efficient Mining of Frequent Closed Sequences. *IEEE Computer Society Washington, DC, USA ©2004*, 79. Obtenido de http://www.cenatav.co.cu/doc/RTecnicos/RT%20SerieGris_016web.pdf

[6] Mattogno, M. A. (2013/14). *riull.ull.es.* Obtenido de http://riull.ull.es/xmlui/bitstream/handle/915/607/Inteligencia%20Empresarial%20combinando%20tecnicas%20de%20Mineria%20de%20Procesos%20y%20Mineria%20de%20Datos.pdf?sequence=1&isAllowed=y

[7] Medeiros, A. K. (2006). *http://www.processmining.org/.* Obtenido de http://www.processmining.org/blogs/pub2006/genetic_process_mining

[8] ProM. (s.f.). *http://www.processmining.org/.* Obtenido de http://www.processmining.org/_media/presentations/processminingtutorialesscass-2009.pdf

[9] ProM. (s.f.). *ProM Tools.* Obtenido de http://www.promtools.org/doku.php

[10] The Tortoise Team. (s.f.). *TortoiseSVN.* Obtenido de https://tortoisesvn.net/downloads.html