MINERÍA DE SERIES TEMPORALES Y FLUJO DE DATOS

MÁSTER EN CIENCIA DE DATOS E INGENIERIA DE COMPUTADORES

Trabajo autónomo II: Minería de Flujo de Datos

Autores
José Ángel Díaz García
joseangeldiazg02@correo.ugr.es

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

—

Granada, Abril de 2018

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este documento encontramos el resultado final alcanzado durante el estudio del apartado de **minería de flujos de datos**, enmarcado dentro de la asignatura de 'Minería de Series Temporales y Flujos de Datos' del máster en Ciencia de Datos de la Universidad de Granada. En este primer capítulo, veremos una introducción al problema a resolver así como a los objetivos a alcanzar con esta práctica y la organización del trabajo.

1.1. Problema a resolver

El problema a resolver en esta práctica se centrará en la resolución de ciertos problemas de clasificación con flujos de datos, de manera tanto estática como dinámica así como con cambio o sin cambio de contexto. Para resolver estos problemas, se propone el uso del software MOA [1] [2] el cual usaremos con la línea de comandos o con la interfaz gráfica que podemos ver en la figura 1.1.

Los problemas a resolver serán:

1. Entrenamiento offline (estacionario) y evaluación posterior.
2. Entrenamiento online.
3. Entrenamiento online en datos con concept drift.

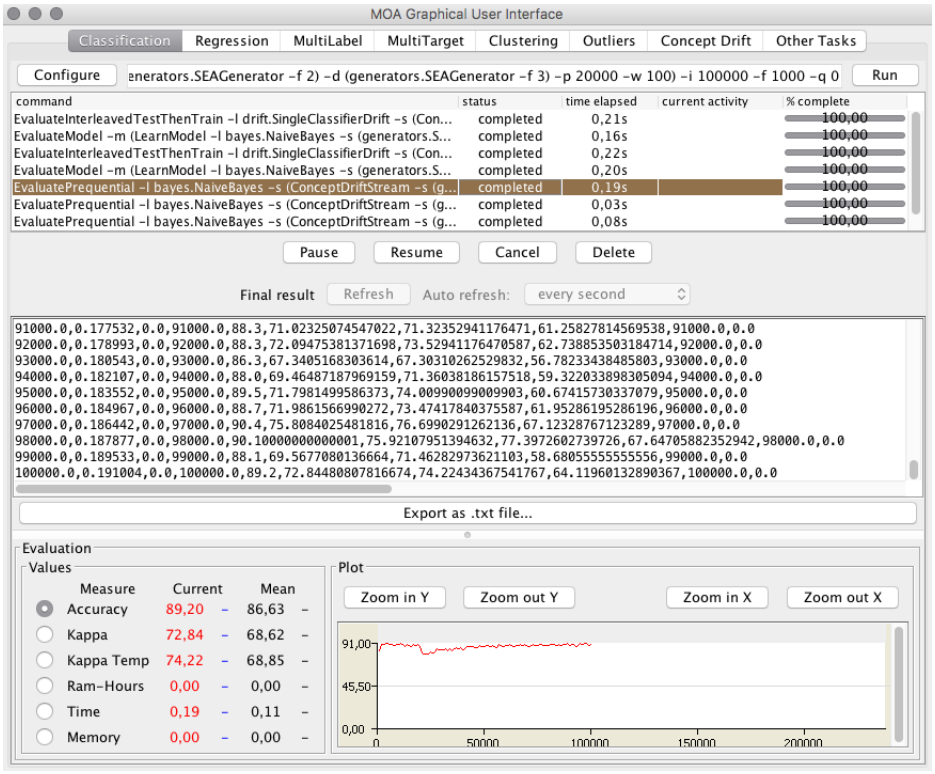

Figura 1.1: Interfaz del software moa.

4. Entrenamiento online en datos con concept drift, incluyendo mecanismos para olvidar instancias pasadas.

5. Entrenamiento online en datos con concept drift, incluyendo mecanismos para reinicializar modelos tras la detección de cambios de concepto.

Posteriormente a la solución de estos problemas, se propone una discusión teórica de los resultados, por lo que encontraremos la práctica dividida en dos apartados, por un lado el teórico (capítulo 3) y por otro el práctico (2).

1.2. Objetivos

Los objetivos de esta práctica serán:

- Asentar y comprender la materia teórica de la minería de flujo de datos vista durante el transcurso de la asignatura.
- Comprender el uso y formas de utilización del software MOA.
- Asentar el conocimiento sobre test estadísticos par comparación de modelos.
- Elaboración de una memoria donde se recojan todos los resultados de manera apropiada.

1.3. Organización del trabajo

La organización del presente documento, se centra en detallar cada uno de los pasos seguidos durante el estudio y resolución del problema planteado en esta introducción, tras la cual tendremos el contenido práctico en el capítulo 2 y el cual representa el grueso de esta memoria. Tras este capítulo encontramos el capítulo 3 donde desde un punto de vista teórico analizamos el concepto de **clasificación** y **concept drift** en minería de flujo de datos. Finalizaremos la memoria con las conclusiones obtenidas en el transcurso de finalización de la misma en el capítulo 4.

Capítulo 2

Práctica

En este capítulo encontramos el desarrollo práctico de este trabajo. El discurso de este capítulo está organizado por secciones, una para cada experimento a resolver con el software MOA.

2.1. Entrenamiento offline y evaluación posterior

Entrenar un clasificador HoeffdingTree offline (estacionario, aprender modelo únicamente), sobre un total de 1.000.000 de instancias procedentes de un flujo obtenido por el generador WaveFormGenerator con semilla aleatoria igual a 2. Evaluar posteriormente (sólo evaluación) con 1.000.000 de instancias generadas por el mismo tipo de generador, con semilla aleatoria igual a 4. Repita el proceso varias veces con la misma semilla en evaluación y diferentes semillas en entrenamiento, para crear una población de resultados. Anotar como resultados los valores de porcentajes de aciertos en la clasificación y estadístico Kappa. Repetir el paso anterior, sustituyendo el clasificador por HoeffdingTree adaptativo. Responda a la pregunta: ¿Cree que algún clasificador es significativamente mejor que el otro en este tipo de problemas? Razone su respuesta.

Para resolver este problema debemos usar un script en el que ejecutaremos varias veces nuestros experimentos para cada uno de los clasificadores. El resultado de ese script sería:

```
for i in ‘seq 1 20‘;
    do
        eval "htnormal=htnormal$i.txt"
        eval "htadaptativo=htadaptativo$i.txt"
        java -cp moa.jar -javaagent:sizeofag-1.0.0.jar moa.DoTask \
         "EvaluateModel -m (LearnModel -l trees.HoeffdingTree -s \
          (generators.WaveformGenerator -i $i)  -m 1000000) -s \
          (generators.WaveformGenerator -i 4)" >  $htnormal \
        java -cp moa.jar -javaagent:sizeofag-1.0.0.jar moa.DoTask  \
        "EvaluateModel -m (LearnModel -l trees.HoeffdingAdaptiveTree -s \
         (generators.WaveformGenerator -i $i)  -m 1000000) -s \
         (generators.WaveformGenerator -i 4)" > >  $htadaptativo
    done
```

2.2. Entrenamiento online

2.3. Entrenamiento online con concept drift

2.4. Entrenamiento online con concept drift, incluyendo mecanismos para olvidar instancias pasadas

2.5. Entrenamiento online en datos con concept drift, incluyendo mecanismos para reinicializar modelos tras la detección de cambios de concepto

Capítulo 3

Teoría

En este capítulo asentaremos los conceptos teóricos utilizados en el transcurso del anterior apartado. Comenzaremos con el concepto de **clasificación** y finalizaremos con el concepto de **concept drift**.

3.1. Clasificación

Explicar el problema de clasificación, los clasificadores utilizados en los experimentos de la sección 2, y en qué consisten los diferentes modos de evaluación/validación en flujos de datos.

El problema de la clasificación es uno de los problemas más ampliamente estudiado en ciencia de datos. Se basa en el entrenamiento y construcción de modelos predictivos en base a un conocimiento previo que viene dado en forma de datasets pre-etiquetados, una vez obtenido estos modelos, se deberá poder predecir la etiqueta o clase de una nueva muestra que se incluya al problema y de la cual no sabemos su clase de pertenencia.

En minería de flujo de datos el problema es similar, salvo por que los datos nos llegan en un flujo continuo, de manera que no podemos tener todo el dataset a priori para aprender y validar con las consiguientes dificultades que esto aporta al problema.

3.2. Concept Drift

Explicar en qué consiste el problema de concept drift y qué técnicas conoce para resolverlo en clasificación.

El ***concept drift***, o cambio de contexto es uno de los principales problemas a los que nos enfrentamos en la minería de flujo de datos y viene a significar que los datos que han llegado en el pasado difieren en mayor o menor medida en los datos que estamos trabajando en este mismo momento por lo que se entra en conflicto con asunciones pasadas sobre los datos y causará un descenso de las medidas de bondad que en algunos casos puede llegar a ser drástico. Detectar estos cambios de contexto apropiadamente y readaptar los modelos será por tanto un gran reto pero necesario en los problemas de minería de flujo de datos. Su importancia en estos problemas es tal que han propiciado una gran línea de investigación con trabajos muy recientes que tratan sobre soluciones en la materia [3] [4] [5] [6].

Como hemos introducido anteriormente, el ***concept drift*** se debe a cambios en los datos, pero estos no vendrán dados siempre de la misma manera sino que podrán presentarse de manera recurrente, gradual, incremental o brusco acorde a los ejemplos que podemos ver en la figura 3.1.

El cambio de contexto podrá deberse a diversos motivos como pueden ser, la variación de características, la presencia de ruido, aparición de nuevas características o la influencia del entorno, siendo este uno de los motivos más delicados. Igualmente, sea cual sea el motivo del cambio de contexto, encontramos los siguientes métodos u algoritmos para para solventar el problema:

- **Aprendizaje Online**: Estos algoritmos continuamente actualizan los parámetros de clasificación mientras el flujo de datos está activo. Hay que tener en cuenta que cada ejemplo solo debe ser procesado una vez, su rendimiento no debería ser distinto a los algoritmos offline y y los requisitos de tiempo y procesado deben ser limitados. Un algoritmo muy común dentro de esta categoría sería el CVFDT [7].

- **Soluciones de ventana**: Son algoritmos que olvidan datos antiguos conforme llegan datos nuevos, asumiendo que estos últimos tienen más relevancia en el problema. Esta solución, hace que de cara a un cambio de contexto el algoritmo reaprenderá este cambio de contexto y aunque tendrá cierto descenso en sus medidas en algún momento temporal, se

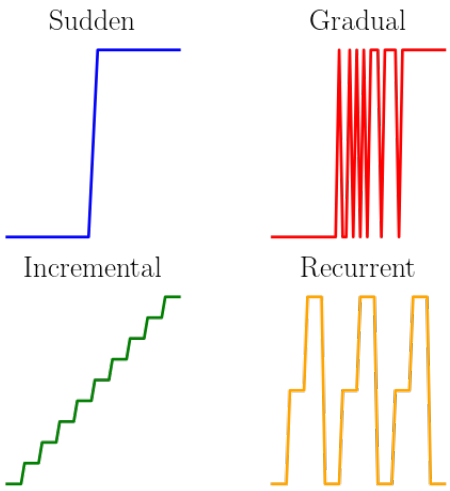

Figura 3.1: Tipos de concept drift.

recuperará y volverá a ofrecer resultados similares a los ofrecidos antes del cambio de contexto. Las soluciones más afamadas son la ventana deslizante, ventana en función de hitos y la ponderación de datos en función de la antigüedad.

- **Ensembles**: Estos son algoritmos que entrenan diversos modelos de clasificadores elementales con ciertas variaciones de manera que la decisión final es una decisión colectiva entre todos ellos. Esta diversidad o variaciones para entrenar pueden deberse a las características, el modelo de clasificación o las instancias usadas para entrenar siendo estos conceptos muy relevantes en entornos cambiantes donde la diversidad puede venir marcada por los nuevos datos del stream, incluido el hipotético cambio de contexto. Los algoritmos más famosos dentro de este área serían los de Street [8] y el de Wang [9].

- **Algoritmos de detección**: Estos algoritmos en lugar de buscar adaptabilidad buscan detectar cuando se producirá cambio de contexto para posteriormente paliar sus efectos. Para detectar un cambio de contexto se suele analizar la disminución de las medidas de bondad. En estos algoritmos encontramos el problema de cuando detectar el cambio de

contexto, si esperamos mucho una vez iniciado el mismo estaremos ante un método que es robusto, pero que perderá accuracy al activar los procesos tarde. Por otro lado, hacen saltar la alarma rápidamente el algoritmo paliará los efectos cuando el cambio de contexto aún no haya perjudicado mucho el modelo, pero por contra se enfrenta al problema de un elevado número de falsos positivos.

Capítulo 4

Conclusión

Bibliografía

[1] Albert Bifet, Geoff Holmes, Richard Kirkby, Bernhard Pfahringer (2010); MOA: Massive Online Analysis; *Journal of Machine Learning Research* 11: 1601-1604

[2] Moa Web Site https://moa.cms.waikato.ac.nz

[3] Sunanda Gamage and Upeka Premaratne. 2017. Detecting and Adapting to Concept Drift in Continually Evolving Stochastic Processes. *Proceedings of the International Conference on Big Data and Internet of Thing (BDIOT2017)*. ACM, New York, NY, USA, 109-114

[4] Sylvio Barbon Junior, Gabriel Marques Tavares, Victor G. Turrisi da Costa, Paolo Ceravolo, and Ernesto Damiani. 2018. A Framework for Human-in-the-loop Monitoring of Concept-drift Detection in Event Log Stream. *Companion of the The Web Conference 2018 on The Web Conference 2018 (WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 319-326.

[5] Sun, Y., Wang, Z., Bai, Y., & Dai, H. A Classifier Graph based Recurring Concept Detection and Prediction Approach.

[6] Wang, Z., Tian, M., & Jia, C. (2017, December). An Active and Dynamic Botnet Detection Approach to Track Hidden Concept Drift. In *International Conference on Information and Communications Security* (pp. 646-660). Springer, Cham.

[7] Hulten, G., Spencer, L., & Domingos, P. (2001, August). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD*

international conference on Knowledge discovery and data mining (pp. 97-106). ACM.

[8] Street, W. N., & Kim, Y. (2001, August). A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 377-382). ACM.

[9] Wang, H., Fan, W., Yu, P. S., & Han, J. (2003, August). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 226-235). AcM.