Práctica: Minería de Flujos de Datos 2024-2025

Master Ciencia de Datos

Universidad de Granada

Práctica: Minería de Flujos de Datos

Miguel García López

Índice

Índice de figuras

Índice de cuadros

1. Cuestiones

1.1. Explica en qué consisten los diferentes modos de evaluación/validación para clasificación en flujos de datos

En clasificación, concretamente en entornos de flujo de datos, los métodos de evaluación difieren de los enfoques estáticos tradicionales debido a la naturaleza dinámica, infinita y potencialmente no estacionaria de los flujos.

Holdout Se toman instantáneas en diferentes momentos durante el entrenamiento del modelo para ver cómo varía la métrica de calidad. Sólo es válido si el conjunto de *test* es similar a los datos actuales (sin *concept drift*) [1].

Test-Then-Train Este enfoque procesa cada nuevo dato en dos fases secuenciales: primero evalúa el modelo (*test*) y luego lo actualiza (*train*). Simula entornos reales de flujos continuos y proporciona métricas en tiempo real, como precisión acumulada.

Prequencial Variante de *Test-Then-Train* que calcula métricas en ventanas deslizantes o bloques. Utiliza dos enfoques: ventanas fijas (evalúa últimos nn datos, como 1000 ejemplos) o ventanas adaptativas (ajusta el tamaño según detección de *drift*, como el algoritmo *ADWIN* [2]). Su ventaja principal es reducir el sesgo hacia datos antiguos. En [3], se aplicó en flujos financieros para medir la adaptabilidad de modelos ante cambios de mercado.

Interleaved Validation Adaptación de la validación cruzada tradicional: divide el flujo en bloques temporales y los rota para entrenamiento y prueba. Este método es útil para evaluar robustez frente a *drift*. En [4], se empleó para comparar algoritmos como *VFDT* y *Hoeffding Adaptive Tree* en presencia de cambios sintéticos en la distribución.

Ventanas Deslizantes (Sliding Windows) Evalúa el modelo solo en datos recientes. Las ventanas pueden ser fijas (mantienen tamaño constante, como los últimos 10000 ejemplos) o adaptativas (ajustan dinámicamente el tamaño usando umbrales de error [5]). Un ejemplo clásico es *VFDT* [6], que usa ventanas para limitar el uso de memoria en flujos infinitos, descartando datos obsoletos.

1.2. Describe tres algoritmos de clasificación en flujos de datos y compara ventajas/desventajas

El primer algoritmo y uno de los más usados es el **VFDT** (*Very Fast Decision Tree*) o Árbol de *Hoeffding*. El **VFDT**, propuesto por *Domingos* y *Hulten* (2000), es un algoritmo incremental que construye árboles de decisión utilizando el *Hoeffding bound* (HB), un límite estadístico que garantiza con alta probabilidad que la mejor división en un nodo, basada en una muestra de datos, será la misma que si se usara el flujo completo. Opera en tiempo constante por muestra y memoria limitada. La cota *Hoeffding* se describe como:

$$HB = \sqrt{\frac{R^2 ln(1/\delta)}{2n}}$$

Donde R es el rango de clases (diferencia máxima posible en las métricas de división, como ganancia de información), n el número de muestras en el nodo, y δ la probabilidad de error [6].

Usa la cota de *Hoeffding* para garantizar que, con alta probabilidad, la mejor elección de división con una muestra será la misma que si se usara todo el flujo de datos. Esto permite hacer divisiones rápidamente sin necesidad de almacenar todos los datos históricos. Para realizar la división se tiene que:

$$\Delta H(X_i) \leq HB < \tau$$

Lo que indica que la diferencia de ganancia de información entre los dos mejores atributos es menor o igual que la cota de *Hoeffding* y esta a su vez es menor que un umbral fijo τ.

Otro ejemplo de algoritmo, basado en el ya mencionado **VFDT**, es el *Optimized Very Fast Decision Tree* (**OVFDT**), el cual implementa un mecanismo de *split* de nodos más eficiente [7]. Este algoritmo utiliza tres tipos de clasificadores en los nodos hoja

- Clase Mayoritaria (MC): Predice la clase más frecuente (igual que **VFDT**).
- Naive Bayes (NB): Calcula probabilidades condicionales para mejorar la precisión en datos ruidosos.
- Naive Bayes Ponderado (WNB): Añade pesos a las clases para mitigar distribuciones desbalanceadas.

Reemplaza el umbral fijo τ de **VFDT** por un valor dinámico calculado como la media de los *Hoeffding Bounds* históricos en cada hoja. A diferencia del algoritmo original, que elegía este valor de forma fija, **OVFDT** lo calcula automáticamente, se puede adaptar al ruido.

$$\tau_{\text{adaptativo}} = \frac{1}{k} \sum_{i=1}^{k} \mu_l \times \text{HB}_i$$

Donde k es el número de evaluaciones de *splits* realizadas en la hoja l, μ_l es una variable binaria que indica que HB se calculó para esa hoja y HB_i es la cota de *Hoeffding* para esa i-ésima evaluación.

Por último, destacar que se han mencionado dos algoritmos basados en árboles. En clasificación básica, los algoritmos basados en técnicas como *bagging* son mucho más potentes que los modelos simples, un ejemplo de esto son los árboles de decisión y los *random forest*.

El algoritmo *Adaptive Random Forest* o **ARF** es un algoritmo que combina la robustez de *Random Forest* con técnicas adaptativas para flujos de datos [8]. Este es capaz de detectar y adaptarse a los cambios de concepto (*concept drift*) mediante monitoreo individual de cada árbol. Cada árbol en el ensemble tiene un detector **ADWIN** que monitorea su error de clasificación.

Si se detecta *drift*, el árbol afectado se marca como obsoleto y se reemplaza por un nuevo árbol entrenado en datos recientes.

VFDT destaca por su velocidad y bajo consumo de memoria, pero carece de adaptación a *concept drift* y es sensible a ruido. **OVFDT** lo mejora con un umbral de desempate adaptativo y hojas funcionales (NB/WNB), logrando mayor precisión y control de tamaño del árbol que a diferencia de las hojas clásicas (que solo predicen la clase mayoritaria), estas incorporan modelos de clasificación locales para tomar decisiones más inteligentes, aunque sigue limitado a un único modelo. **ARF**, al ser un *ensemble* adaptativo, ofrece robustez superior frente a ruido, cambios de concepto y desbalance, pero requiere más memoria para albergar todos los modelos mayor potencia computacional para realizar el monitoreo.

1.3. Explica en qué consiste el problema de concept drift y describe qué técnicas conoces para resolverlo en clasificación, clustering y patrones frecuentes

El *concept drift* se refiere a una dificultad que surge del aprendizaje de datos según estos fluyen con el tiempo. El cambio de concepto de los datos implica que un modelo aprendido en el pasado ya no es consistente con los datos recibidos en el presente.

Un tratamiento adecuado con este efecto involucraría la actualización o creación de un nuevo modelo que tenga en cuenta nuevas características que entren en conflicto con asunciones pasadas, pero conservando asunciones basadas en datos antiguos que sigan cumpliéndose. El *concept drift* es **real** si altera la frontero de decisión y es **virtual** si solo varía la distribución de los datos sin alterar esa frontera.

CVFDT (*Concept-adapting Very Fast Decision Tree*) [9] es una extensión de los árboles de decisión que mantiene estadísticas alternativas en cada nodo, permitiendo crear subárboles alternativos cuando detecta cambios significativos en los patrones de los datos. **Técnica:** Utiliza **ventanas deslizantes** sobre el flujo de datos y test de Hoeffding para comparar distribuciones entre ventanas temporales. Sin embargo, presenta limitaciones como la incapacidad de manejar atributos continuos y un mayor consumo computacional respecto a su versión base.

El algoritmo **CluStream** divide su proceso en dos componentes principales [10]. En la fase *online*, mantiene modelos resumidos mediante estadísticos llamados *microclusters*, que contienen información cuantitativa (número de datos, suma de valores, suma de cuadrados) y temporal (suma de tiempos y sus cuadrados) [10]. Cada nuevo dato se asigna al *microcluster* más cercano; si no coincide con ningún, se crea uno nuevo, eliminando o fusionando los más antiguos cuando se excede la capacidad. Emplea **detección implícita** mediante actualización continua de *microclusters*. En la fase *offline*, utiliza los *microclusters* almacenados para reconstruir agrupaciones mediante algoritmos como *k-means*. Los *microclusters* se organizan en instantáneas temporales con estructura piramidal, optimizando el almacenamiento y permitiendo análisis multiescala. **CluStream** se adapta implícitamente a cambios en los datos mediante actualizaciones continuas, aunque **no detecta activamente el *concept drift***.

StreamDD mejora este enfoque añadiendo detección explícita de *drift* [11]. Emplea el *test* de *Page-Hinkley* para monitorizar la distancia máxima promedio entre *microclusters*. Combina **detección explícita** (test estadístico) con **ventanas adaptativas** que se redimensionan según la magnitud del *drift* detectado. Al superar un umbral en la suma acumulada de cambios, activa una actualización del modelo en la fase *offline*.

En el ámbito de los patrones frecuentes, estos se relacionan con los ítems de las reglas de asociación. El algoritmo **Fuzzy-CSar** destaca por su procesamiento incremental con reglas difusas que manejan imprecisión [12]. Genera reglas dinámicamente mediante un *operador de cobertura* (combina variables y términos lingüísticos aleatoriamente) y

competencia en nichos por variable de salida, priorizando confianza y soporte. Aplica **adaptación reactiva** mediante algoritmos genéticos que reconfiguran reglas cuando su soporte cae bajo umbrales dinámicos. Además, utiliza algoritmos genéticos (AG) para cruce y mutación de reglas, ajustando condiciones y términos lingüísticos en tiempo real.

2. Bibliografía

[1] J. Casillas, *Minería en Flujo de Datos*, Máster en Ciencias de Datos e Ingeniería de Computadores, Universidad de Granada, 2025. URL: `http://decsai.ugr.es/~casillas`.

[2] A. Bifet y R. Gavaldà, "Learning from Time-Changing Data with Adaptive Windowing," en *Proceedings of the 7th SIAM International Conference on Data Mining*, 2007, págs. 443-448.

[3] J. Gama, *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 2010, ISBN: 978-1439826119.

[4] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy y A. Bouchachia, "A Survey on Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, n.º 4, págs. 1-37, 2014.

[5] A. Bifet, G. Holmes, B. Pfahringer y R. Gavaldà, "Adaptive Hoeffding Trees for Learning from Data Streams," en *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM)*, 2009, págs. 139-147.

[6] P. Domingos y G. Hulten, "Mining High-Speed Data Streams," en *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, págs. 71-80.

[7] H. Yang y S. Fong, "Incremental Optimization Mechanism for Constructing a Decision Tree in Data Stream Mining," *Mathematical Problems in Engineering*, vol. 2013, págs. 1-14, 2013. DOI: `10.1155/2013/580397`.

[8] H. M. Gomes, J. Read, A. Bifet, J. P. Barddal y J. Gama, "Adaptive random forests for evolving data stream classification," en *Machine Learning*, vol. 106, Springer, 2017, págs. 1469-1495.

[9] G. Hulten, L. Spencer y P. Domingos, "Mining time-changing data streams," en *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2001, págs. 97-106.

[10] C. Aggarwal, J. Han, J. Wang, P. Yu, T. Watson y R. Ctr, "A Framework for Clustering Evolving Data Streams," jun. de 2003.

[11] H. Leila, B. Hichem y B. Karima, "StreamDD: Stream clustering with Drift Detection," *Procedia Computer Science*, vol. 246, págs. 1240-1249, 2024, 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024), ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2024.09.550`. URL: `https://www.sciencedirect.com/science/article/pii/S187705092402595X`.

[12] A. Orriols-Puig, F. J. Martínez-López, J. Casillas y N. Lee, "A soft-computing-based method for the automatic discovery of fuzzy rules in databases: Uses for academic research and management support in marketing," *Journal of Business Research*, vol. 66, n.º 9, págs. 1332-1337, 2013, Advancing Research Methods in Marketing, ISSN: 0148-2963. DOI: https://doi.org/10.1016/j.jbusres.2012.02.033. URL: https://www.sciencedirect.com/science/article/pii/S0148296312000653.