

Una Aplicación Interactiva para Visualizar las Hipótesis Generadas por Algoritmos de Aprendizaje Computacional

Santiago David Villalba Bartolomé, Juan José Rodríguez Díez*

*Área de Lenguajes y Sistemas Informáticos
Universidad de Burgos
e-mail: sdvb@wanadoo.es, jjrodriguez@ubu.es

Resumen

Se presenta una herramienta que puede ser empleada en la docencia de asignaturas de Minería de Datos, Aprendizaje Computacional, o de Inteligencia Artificial, que incluyan estas cuestiones dentro de su temario. Inspirada en diversas aplicaciones que visualizan las superficies de decisión generadas por algún tipo de clasificador, la presente herramienta es capaz de mostrar gráficamente la hipótesis lanzada por una vasta gama de clasificadores, desde árboles de decisión hasta redes neuronales, en el marco de un problema bidimensional (sencillo aunque visualizable) y multiclase.

1. Introducción

OAIDTB (Otra Aplicación Interactiva Demostrando Técnicas de Boosting) es una suite de programas para la aplicación y evaluación de técnicas de clasificación en general, y de boosting en particular, a problemas de aprendizaje computacional. Los programas que incluye la suite son los siguientes:

- Una biblioteca de clases java que implementan diversos algoritmos de boosting, extendiendo a la biblioteca WEKA [11]; a este conjunto de clases las llamaremos a partir de ahora boosters. Pueden ser utilizados desde la línea de comandos o embebidos en otro código java.
- Una aplicación gráfica para la selección, configuración, aplicación y evaluación de los métodos de aprendizaje tanto de WEKA como de OAIDTB.
- Una aplicación gráfica docente, *Bidimensional Generalization*, que permite visualizar, de una

manera general, las hipótesis lanzadas por los clasificadores de WEKA y OAIDTB en el marco de un problema bidimensional y multiclase.

Existen diversas herramientas capaces de visualizar las superficies de decisión generadas por algún sistema de clasificación, como redes neuronales [13], máquinas de vectores soporte (SVM) [12], boosting [9][10], etc. La aportación principal de la herramienta que se presenta en este trabajo es que es capaz de visualizar una amplia gama de clasificadores, todos los incluidos en la biblioteca WEKA [11], así como aquellos que se implementen dentro del marco definido por esta biblioteca. En particular, si en la asignatura se exige la implementación de algún algoritmo, los alumnos son capaces de visualizar las hipótesis generadas por el clasificador que han desarrollado sin necesidad de programar la parte gráfica.

OAIDTB es software de código abierto y se distribuye bajo la licencia GNU General Public License. Se puede encontrar la última versión en <http://pisuerga.inf.ubu.es/lsi/Asignaturas/MD/>.

2. Un poco de teoría

A pesar de que se presupone en el lector un cierto conocimiento de los fundamentos necesarios para comprender el presente texto, se dan a continuación unas breves definiciones, más o menos precisas, de varios conceptos fundamentales.

- *Minería de datos*: Aplicación de algoritmos específicos para la extracción de patrones desde los datos.

- *Aprendizaje computacional*: En el contexto en que nos encontramos, este término se refiere al conjunto de teorías y algoritmos que forman la base técnica para la minería de datos.
- *Instancia*: La entrada para un esquema de aprendizaje computacional es un conjunto de instancias. Estas instancias son las “cosas” que deben ser clasificadas, agrupadas o asociadas. Cada instancia es un ejemplo individual e independiente del concepto que debe ser aprendido y cada instancia está caracterizada por los valores que toman un conjunto predeterminado de atributos.
- *Clasificador*: un algoritmo de aprendizaje clasificador trata de extraer información de un conjunto de instancias de entrenamiento, de manera que es capaz de predecir la clase a la que pertenecen ejemplares desconocidos (distintos de los utilizados para entrenarlo).
- *Boosting*: Los algoritmos de aprendizaje computacional basados en boosting (potenciación) buscan iterativamente una combinación lineal ponderada de clasificadores base que haga predicciones correctas en datos desconocidos.

Las clases que implementan a los algoritmos de boosting son el principal componente de la suite. Los algoritmos implementados son AdaBoostM1 [4], AdaBoostM1W [2], AdaBoostOC [6], AdaBoostECC [8], AdaBoostMH [5], RealAdaBoost [3], GentleAdaBoost [3], AdaCost [1] y CSBx [7].

3. Las aplicaciones gráficas

En el terreno de la didáctica son las aplicaciones gráficas de la suite OAIDTB las que tienen algo de mucho que decir.

La interfaz gráfica ha sido diseñada con el afán de ser cómoda e intuitiva para el usuario, proporcionando ayuda sobre qué es y para qué sirve cada componente mediante “tooltips”. El funcionamiento de las aplicaciones se basa, fundamentalmente, en el empleo del ratón, aunque se proporcionan teclas de acceso rápido a la mayoría de las funciones. Además, las aplicaciones no pierden en ningún momento la interactividad con el usuario, pudiendo siempre ser cancelados los procesos lanzados a pesar de

que puedan, eventualmente, tener una importante carga computacional.

Sin más preámbulos, echemos un vistazo a lo que nos ofrecen estos programas.

3.1. El panel de clasificación

El panel de clasificación (Fig. 1) es una herramienta que permite seleccionar, o cargar un clasificador de OAIDTB o de WEKA, configurarlo, entrenarlo sobre un conjunto de instancias, guardarlo, evaluar su precisión proporcionando variadas estadísticas y una representación textual del clasificador y, en el caso de los boosters, mostrar detalles tales como la evolución del error, mediante gráficas cartesianas, o la precisión de los clasificadores base entrenados.

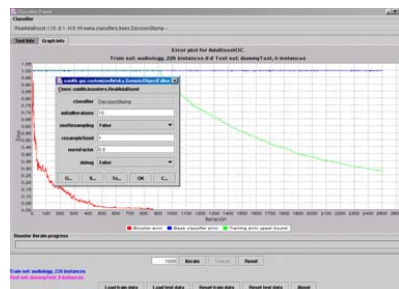


Figura 1. El panel de clasificación

La aplicación es flexible, ya que añadirle nuevos esquemas de clasificación es sencillo. Basta con implementar el algoritmo deseado en el marco proporcionado por la biblioteca WEKA y añadir el nombre de la clase a un fichero de texto que permite configurar la aplicación. La utilización de las capacidades de java para descubrir los miembros de las clases en tiempo de ejecución proporciona la posibilidad de incorporar estas clases, con sus características particulares, sin tener que recompilar el programa.

3.2. “Generalización Bidimensional”

Esta aplicación permite la visualización de las hipótesis lanzadas por diversos algoritmos de

aprendizaje computacional en el marco de un problema de datos bidimensional y multiclase.

En concreto, se trata de clasificar instancias representadas por puntos en un espacio bidimensional y cuyo atributo clase es el color de dicho punto. Con ella es posible enseñar, de una manera práctica y visual, cómo funcionan dichos algoritmos, mostrando propiedades teóricas de los mismos y conectando las hipótesis que lanzan con un problema que, aunque limitado, es fácil de entender y “entra por los ojos”.

Una vez ejecutada, la ventana de la aplicación se divide en varias zonas (Fig. 2).

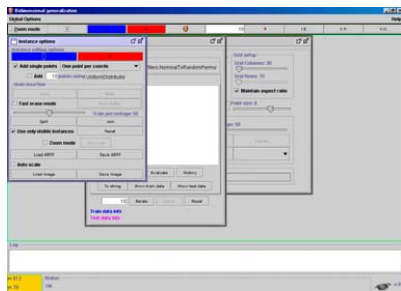


Figura 2. Pantalla principal de la aplicación

En la parte superior existe un pequeño menú con algunas opciones de configuración de la aplicación, y una barra de herramientas con botones para las acciones más frecuentes. En el centro se sitúa el área de dibujo, que es la región de la pantalla en que se dibujan las instancias (puntos de un determinado color) que introduce el usuario mediante pulsaciones del ratón. En la parte inferior existe un área de mensajes, en la que se le muestran al usuario las informaciones pertinentes, desde tiempos de ejecución hasta los posibles errores o excepciones producidas.

Flotando sobre todos estos componentes hay tres ventanas cuyo tamaño, para comodidad del usuario, puede ser alterado.

La primera ventana se corresponde con el panel de clasificación, ya comentado en la sección anterior.

La segunda ventana es el panel de edición de instancias. Permite introducir las instancias mediante pulsaciones de ratón en el área de dibujado, o cargarlas desde un archivo, bien sea en formato propio (ARFF – Formato de Archivo Atributo-Relación) o, y esta característica es experimental, desde un archivo de imagen gif o jpeg. Si se utiliza la inserción directa mediante pulsaciones del ratón, este panel permite la selección del color (clase) de los puntos (instancias) introducidos, así como permite la posibilidad de introducir un número arbitrario de instancias. Algunas funciones ofrecidas a través de los componentes de este panel son:

- Deshacer / rehacer la inserción de instancias.
- Configurar la función empleada en la inserción masiva de instancias, pudiendo elegir entre diversas funciones de distribución (normal, uniforme... o incluso las creadas por un usuario avanzado mediante del empleo de una simple API) para poder crear problemas interesantes o visualmente atractivos.
- Trasladar y escalar las coordenadas de las instancias para poder mostrarlas correctamente o más claramente en el área de dibujado.
- Dividir las instancias en conjuntos de entrenamiento y de prueba.
- Salvar las instancias en formato ARFF o una instantánea del contenido del área de dibujado.

La tercera ventana se corresponde con el panel de opciones visuales, que es el encargado de manejar todo lo referente a la visualización del problema, desde qué aspectos del mismo se muestran u ocultan hasta el tamaño de los puntos que representan las instancias, y de lanzar la creación de la imagen que represente la hipótesis del clasificador seleccionado y construido en el panel de clasificación, pudiendo configurar su nivel de precisión

Otro de los puntos fuertes del programa es la capacidad de hacer “zoom ilimitado” sobre cualquier región del universo del problema y

construir la imagen de hipótesis sólo sobre dicha región, de manera que es posible revelar las peculiaridades del clasificador en cuestión, como por ejemplo los límites de decisión de un árbol o de un clasificador basado en instancias.

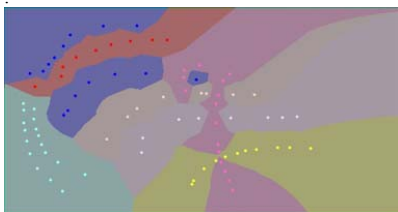


Figura 3. Hipótesis obtenida de un clasificador basado en instancias.

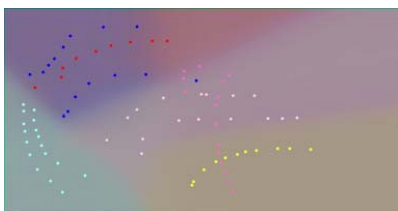


Figura 4. Hipótesis generada por una red neuronal



Figura 5. Zoom sobre una región de la figura 3

4. Conclusión

Se ha presentado una herramienta que permite representar, de manera gráfica y general, las hipótesis lanzadas por algoritmos de aprendizaje computacional basados en clasificación.

Referencias

- [1] Fan, Stolfo, Zhang y Chan. *AdaCost: Misclassification cost-sensitive boosting*. Proceedings of The Sixteenth International Conference on Machine Learning. pp. 97-105. San Francisco: Morgan Kaufmann.
- [2] Günther Eibl y Karl Peter Pfeiffer. *How to make AdaBoost.M1 work for weak base classifiers by changing only one line of the code*. ECML'02 - European Conference on Machine Learning
- [3] Jerome Friedman, Trevor Hastie y Robert Tibshirani. *Additive logistic regression: a statistical view of boosting*. The Annals of Statistics, 38(2): 337-374, April 2000.
- [4] Robert E. Schapire y Yoav Freund. *A decision-theoretic generalization of on-line learning and an application to boosting*. Journal of Computer and System Sciences, 55(1):119-139, 1997. van Leunen, M.C. *A handbook for scholars*. Oxford University Press, 1992
- [5] Robert E. Schapire y Yoram Singer. *Improved boosting algorithms using confidence rated-predictions*. Machine Learning, 37(3):297-336, 1999
- [6] Robert E. Schapire. *Using output codes to boost multiclass learning problems*. In Machine Learning: Proceedings of the Fourteenth International Conferences, pages 313-321, 1997.
- [7] Ting, K.M.. *Cost Sensitive Classification Using Decision Trees, Boosting and MetaCost*. Book chapter in Heuristic and Optimization for Knowledge Discovery. Edited by Sarker, R., Abbass, H. & Newton, C. Idea Group Publishing. 2002.
- [8] Venkatesan Guruswami y Amit Sahai. *Multiclass Learning, Boosting and Error Correcting Codes*. Proceedings of COLT'99
- [9] [<http://www.cs.huji.ac.il/~yoavf/adaboost/index.html>], visitada el 23-1-03.
- [10] [<http://www.cs.technion.ac.il/~rani/LocBoost/>], visitada el 23-1-03.
- [11] [<http://www.cs.waikato.ac.nz/ml/weka/>], visitada el 23-1-03.
- [12] [<http://svm.dcs.rhnc.ac.uk/pagesnew/GPat.shtml>], visitada el 23-1-03.
- [13] [http://neuron.eng.wayne.edu/java/AHK/EP_M_pps.html], visitada el 23-1-03