Algoritmos de clasificación en Weka

Weka es una colección de algoritmos de aprendizaje automático para tareas de minería de datos. Los algoritmos pueden ser aplicados directamente a un conjunto de datos o llamados desde código Java. Weka contiene herramientas para pre-procesamiento de datos, clasificación, regresión, clustering, reglas de asociación y visualización. También es adecuado para desarrollar nuevos esquemas de aprendizaje automático [atUoW]. En el problema de estudio se utiliza el conjunto de algoritmos de clasificación de Weka [WFHP16]. Los algoritmos de clasificación de Weka que se utilizarán son los siguientes [htt]:

Cuadro 1: Clasificadores Weka

Categoría del clasificador	***Nombre del clasificador***	***Modelo, técnica o algoritmo que implementa***
Clasificadores bayesianos	BayesNet	Bayes Network (Red Bayesiana) [BFH$^+$16]
Clasificadores bayesianos	NaiveBayes	Naive Bayes [JL95]
Clasificadores bayesianos	NaiveBayesMultinomial	Naive Bayes multinomial [MN98]
Clasificadores bayesianos	NaiveBayesMultinomialUpdateable	Naive Bayes multinomial actualizable [MN98]
Clasificadores bayesianos	NaiveBayesUpdateable	Naive Bayes actualizable [JL95]
Basado en funciones	Logistic	Regresión Logística [lCvH92]

Basado en funciones	MultilayerPerceptron	Red Neuronal con *back propagation*
Basado en funciones	SimpleLogistic	Regresión Logística lineal con LogitBoost [LHF05] [SFH05]
Basado en funciones	SMO	Sequential Minimal Optimization con Support Vector [Pla98] [KSBM01] [HT98]
Clasificadores perezosos	IBk	K-nearest neighbours (K vecinos más cercanos) [AK91]
Clasificadores perezosos	KStar	K* con función de distancia basada en entropía [CT95]
Clasificadores perezosos	LWL	Locally Weighted Learning (Aprendizaje Ponderado Localmente) [FHP03] [AMS96]
Meta algoritmos	AdaBoostM1	Adaboost M1 [FS96]
Meta algoritmos	AttributeSelectedClassifier	Selección de atributos
Meta algoritmos	Bagging	Bagging [Bre96]
Meta algoritmos	ClassificationViaRegression	Métodos de regresión [FWI$^+$98]
Meta algoritmos	CVParameterSelection	Selección de parámetros [Koh95a]
Meta algoritmos	FilteredClassifier	Filtro arbitrario
Meta algoritmos	IterativeClassifierOptimizer	Optimización del número de iteraciones
Meta algoritmos	LogitBoost	Regresión Logística aditiva [FHT98]
Meta algoritmos	MultiClassClassifier	Metaclasificador

Meta algoritmos	MultiClassClassifierUpdateable	Metaclasificador actualizable
Meta algoritmos	MultiScheme	Selección del clasificador
Meta algoritmos	RandomCommittee	Conjunto aleatorizado de clasificadores base
Meta algoritmos	RandomizableFilteredClassifier	Clasificador arbitrario con filtro arbitrario
Meta algoritmos	RandomSubSpace	Árbol de decisión [Ho98]
Meta algoritmos	Stacking	Combinación de clasificadores utilizando apilamiento [Wol92]
Meta algoritmos	Vote	Combinación de clasificadores [Kun04] [KHDM98]
Meta algoritmos	WeightedInstancesHandlerWrapper	Soporte de instancias ponderadas
Sistema de reglas	DecisionTable	Tabla de decisión simple [Koh95b]
Sistema de reglas	JRip	"Repeated Incremental Pruning to Produce Error Reduction" (RIPPER) [Coh95]
Sistema de reglas	OneR	Clasificador 1R [Hol93]
Sistema de reglas	PART	Divide y vencerás para construir un árbol de decisión C4.5 parcial [FW98]
Sistema de reglas	ZeroR	Clasificador 0-R
Árboles de decisión	DecisionStump	Decision stump in conjunction with a boosting algorithm

Árboles de decisión	HoeffdingTree	Algoritmo de inducción incremental del árbol de decisión [HSD01]
Árboles de decisión	J48	Árbol de decisión C4.5 podado o no podado [Qui93]
Árboles de decisión	LMT	"Árboles de Modelos Logísticos" o "Logistic Model Trees" (LMT) [LHF05] [SFH05]
Árboles de decisión	RandomForest	"Bosque de Árboles Aleatorios" o "Forest of Random Trees" [Bre01]
Árboles de decisión	RandomTree	Considera K atributos elegidos al azar en cada nodo. No realiza poda.
Árboles de decisión	REPTree	Construye un árbol de decisión/regresión utilizando la información de ganancia/varianza

Los algoritmos utilizados para construir los clasificadores son los proveídos por la herramienta WEKA según tabla y los parámetros de configuración establecidos por defecto. Otra forma de categorizar los clasificadores incluidos en WEKA es como sigue:

- Bayesianos: BayesNet, NaiveBayes, NaiveBayesUpdateable.
- Basados en funciones: Logistic, MultilayerPerceptron, SimpleLogistic, SMO.
- Basados en reglas: OneR, DecisionTable, JRip, PART, ZeroR.
- Basados en árboles: DecisionStump, J48, LMT, RandomForest, RandomTree, REPTree.

Bibliografía

[AK91] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[AMS96] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *AI Review*, 1996.

[atUoW] Machine Learning Group at the University of Waikato. Weka 3: Data mining software in java.

[BFH+16] Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, and David Scuse. *WEKA Manual for Version 3-8-0*, April 2016.

[Bre96] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Bre01] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[Coh95] William W. Cohen. Fast effective rule induction. In *Twelfth International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[CT95] John G. Cleary and Leonard E. Trigg. K*: An instance-based learner using an entropic distance measure. In *12th International Conference on Machine Learning*, pages 108–114, 1995.

[FHP03] Eibe Frank, Mark Hall, and Bernhard Pfahringer. Locally weighted naive bayes. In *19th Conference in Uncertainty in Artificial Intelligence*, pages 249–256. Morgan Kaufmann, 2003.

[FHT98] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Stanford University, 1998.

[FS96] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Thirteenth International Conference on Machine Learning*, pages 148–156, San Francisco, 1996. Morgan Kaufmann.

[FW98] Eibe Frank and Ian H. Witten. Generating accurate rule sets without global optimization. In J. Shavlik, editor, *Fifteenth International Conference on Machine Learning*, pages 144–151. Morgan Kaufmann, 1998.

[FWI+98] E. Frank, Y. Wang, S. Inglis, G. Holmes, and I.H. Witten. Using model trees for classification. *Machine Learning*, 32(1):63–76, 1998.

[Ho98] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[Hol93] R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

[HSD01] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 97–106. ACM Press, 2001.

[HT98] Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1998.

[htt] http://weka.sourceforge.net/doc.dev/weka/classifiers/Classifier.html. Interface classifier.

[JL95] George H. John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, San Mateo, 1995. Morgan Kaufmann.

[KHDM98] J. Kittler, M. Hatef, Robert P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[Koh95a] R. Kohavi. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*. PhD thesis, Stanford University, Department of Computer Science, Stanford University, 1995.

[Koh95b] Ron Kohavi. The power of decision tables. In *8th European Conference on Machine Learning*, pages 174–189. Springer, 1995.

[KSBM01] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649, 2001.

[Kun04] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley and Sons, Inc., 2004.

[lCvH92] S. le Cessie and J.C. van Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.

[LHF05] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. 95(1-2):161–205, 2005.

[MN98] Andrew Mccallum and Kamal Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 Workshop on 'Learning for Text Categorization'*, 1998.

[Pla98] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[Qui93] Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[SFH05] Marc Sumner, Eibe Frank, and Mark Hall. Speeding up logistic model tree induction. In *9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 675–683. Springer, 2005.

[WFHP16] Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. *Data Mining - Practical Machine Learning Tools and Techniques - Fourth Edition*. Cuarta edition, 2016.

[Wol92] David H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.