Very Fast Decision Rules for Multi-class Problems

P. Kosina[1] J. Gama[2]

[1]LIAAD-INESC Porto, FI MU Brno

[2]LIAAD-INESC Porto, FEP-University of Porto

27th Symposium On Applied Computing

Contents

Data Streams

- Highly detailed, automatic, rapid data feeds.
  - Radar: meteorological observations.
  - Satellite: geodetics, radiation.
  - Astronomical surveys: optical, radio,.
  - Internet: traffic logs, user queries, email, financial,
  - Sensor networks: many more *observation points* ...
- Most of these data will never be seen by a human!
- Need for near-real time analysis of data feeds.

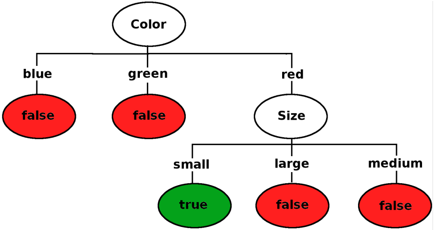
Data Stream Computational Model

Data stream processing algorithms require:

- small constant time per record;
- restricted use of main memory;
- be able to build a model using at most one scan of the data;
- be able to detect and react to concept drift;
- make a usable model available at anytime;
- produce a model with similar performance to the one that would be obtained by the corresponding memory based algorithm, operating without the above constraints.

Decision Trees and Rule Sets

- Decision trees and Rule Sets are *almost* equivalent:



Rules

1: Color = blue -> false
2: Color = green -> false
3: and(Color = red; Size = small) -> true
4: and(Color = red; Size = large) -> false
5: and(Color = red; Size = medium) -> false

- High degree of interpretability

Rule Sets

- Decision trees
  - A decision tree covers all the instance space
  - Each node has a context defined by previous nodes in the path
  - Large decision trees are difficult to understand because of a specific context established by the antecedent nodes
- Rules
  - Each rule covers a specific region of the instance space
  - The Union of all rules can be smaller than the Universe
  - Rules can be interpretable per si:
    - Remove conditions in a rule without removing in another rule.
    - Loss the distinction between tests near the root and near the leaves.
    - Advantage of rule sets: *modularity* and consequently *interpretability*

Rule Learning Systems

- Learning as search
- Two approaches:
  - From the most general to the most specific:
    Top-down
    Model driven
  - From the most specific to the most general:
    Bottom-up
    Data driven
- In this work we focus on top-down learning decision rules in multi-class classification problems.

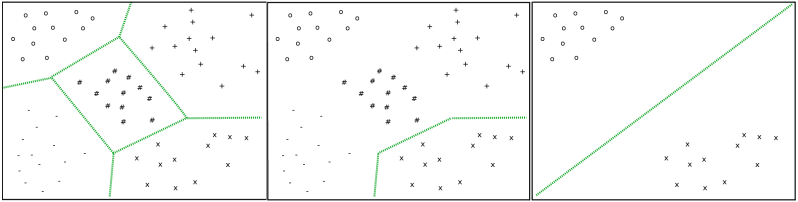
Multi-class Rule Learning Systems

Different strategies to handle multi-class problems:

- direct multi-class
  find the best *literal* that discriminates between all the classes
- one vs. all
  find the best *literal* that discriminates one (positive) class from all the other classes
- all vs. all
  transform c-class problem into $\frac{c \times (c-1)}{2}$ two-class problems
  learn a classifier for each pair of classes.

Rule Learning Systems

direct multi-class one vs. all all vs. all



VFDR Learning

- VFDR is one-pass, any-time, incremental algorithm for data stream classification
- The classifier incrementally learns from labeled examples
  - creates new rules,
  - expands existing ones
- A rule covers an example when all the literals are true for the example
- A rule set is a set of rules plus a default rule
- Default rule's statistics are updated if none of the rules in the set covers the example

VFDR Rule Sets

- VFDR starts from a *default* rule
  $\{\} \rightarrow \mathcal{L}$ where $\mathcal{L}$ is a structure containing
  the sufficient statistics to expand rule **and** the information needed to classify examples.
- A rule has the form of $\{\text{set of literals}\} \rightarrow \mathcal{L}_r$
  A rule covers an example when all the literals are true for the example
- A rule set is a set of rules plus a default rule
- Only the labeled examples covered by a rule update its $\mathcal{L}_r$
- Default rule's statistics are updated if none of the rules in the set covers the example

VFDR Multi-class

- Rule Expansion
  - Rule expansion considers new literals in a *one vs. all* fashion
  - Uses FOIL gain computed on its $\mathcal{L}_r$
  - The expansion of a rule is controlled by Hoeffding bound
- Prediction
  - Simple prediction strategy:
    uses the class distribution in $\mathcal{L}_r$ and selects the majority class
  - Bayes prediction strategy:
    select the class that maximizes posteriori probability given by the Bayes rule using the statistics in $\mathcal{L}_r$.

VFDR Multi-class - FOIL' Gain

- Change in gain between rule r and a candidate rule r'

$$Gain(r', r) = s \times \left(\log_2 \frac{N'_+}{N'} - \log_2 \frac{N_+}{N} \right)$$

N: number of examples covered by r
N_+: number of positive examples covered by r
N': number of examples covered by r'
N'_+ number of positive examples covered by r'

s % of true positives in r that are still true positives in r'

Measures the effect of adding another literal in the rule.

LIAAD
LABORATÓRIO DE INTELIGÊNCIA ARTIFICIAL E APOIO À DECISÃO
INESCPORTO LA
U.PORTO

VFDR Multi-class - FOIL' Gain

Normalized gain:

$$GainNorm(r', r) = \frac{Gain(r', r)}{N_+ \times \left(-\log_2 \frac{N_+}{N}\right)}$$

VFDR Multi-class - Two Approaches

- `VFDR-MC` learns two types of rule sets:
- Unordered
  - Rules are independent
  - All rules that cover an example update their statistics
  - Prediction using a weighted sum classification strategy
- Ordered
  - First rule that covers an example updates its statistics
  - Prediction uses a *first hit* classification strategy

VFDR MC: Illustrative Example

STAGGER SET:

Size = {small,medium,large}, Color = {red,green,blue}, Shape = {circle, square, triangle}

If Size = small and Color = red **then** true **else** false

$$\{\} \longrightarrow \mathcal{L}$$

LIAAD LABORATÓRIO DE INTELIGÊNCIA ARTIFICIAL E APOIO À DECISÃO INESCPORTO LA U.PORTO

VFDR MC: Illustrative Example

STAGGER SET:

Size = {small,medium,large}, Color = {red,green,blue}, Shape = {circle, square, triangle}

If Size = small and Color = red **then** true **else** false

$1 : Size = small \longrightarrow true$
$2 : Size = medium \longrightarrow false$

VFDR MC: Illustrative Example

STAGGER SET:

Size = {small,medium,large}, Color = {red,green,blue}, Shape = {circle, square, triangle}

If Size = small and Color = red **then** true **else** false

$1 : Size = small \longrightarrow true$ **expands for each class:**

$$and(Size = small; Color = red;) \longrightarrow true$$
$$and(Size = small; Color = green;) \longrightarrow false$$

$2 : Size = medium \longrightarrow false$

VFDR MC: Illustrative Example

STAGGER SET:

Size = {small,medium,large}, Color = {red,green,blue}, Shape = {circle, square, triangle}

If Size = small and Color = red **then** true **else** false

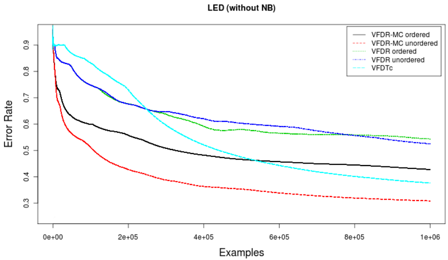
$$1 : and(Size = small; Color = red;) \longrightarrow true$$
$$2 : Size = medium \longrightarrow false$$
$$3 : and(Size = small; Color = green;) \longrightarrow false$$

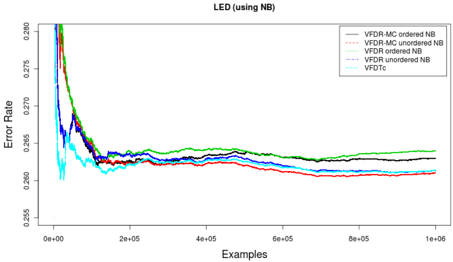
Datasets

Table: Datasets

Set	type	attributes	noise (%)	training set size	test
SEA	artificial	numerical	10	100,000	100,000
LED	artificial	nominal	10	1,000,000	100,000
RT	artificial	nominal	0	1,000,000	100,000
Hyperplane	artificial	numerical	5	1,000,000	100,000
Covertype	real	mix	?	464,810	116,202
KDDCup99	real	mix	?	4,898,431	311,029

Results - prequential error





LIADO
LABORATÓRIO DE INTELIGÊNCIA ARTIFICIAL E APOIO À DECISÃO
INESCPORTO LA
U.PORTO

Results - Train and Test

	Error rate % (variance)				
	VFDR$^{o}_{NB}$	VFDR-MC$^{o}_{NB}$	VFDR$^{u}_{NB}$	VFDR-MC$^{u}_{NB}$	VFDTc
LED	26.16 (0.04)	26.1 (0.05)	26.49 (0.59)	26.0(0)	26.0 (0.01)
RT(2,4,2,4)	0	0	0	0	0
RT(4,5,2,5)	0	0	15.59	0	0
RT(4,15,5,4)	26.14	20.69	42.26	11.1	0
SEA	13.24 (0.16)	12.86 (0.77)	14.71 (1.45)	**10.56 (0.11)**	11.12 (0.16)
Hyperplane	24.99 (11.53)	25.44 (12.14)	23.66 (9.82)	23.51 (15.09)	**23.12 (15.42)**
KDDCup	10.09	9.4	9.91	8.87	**8.3**
Covtype	58.71 (13.28)	49.49 (42.97)	60.65 (17.88)	**36.46 (7.38)**	38.15 (0.61)

Results - Train and Test

	Size				
	VFDR^o	VFDR-MC^o	VFDR^u	VFDR-MC^u	VFDTc
LED	22	21	47	1052	47
RT(2,4,2,4)	7	3	10	9	9
RT(4,5,2,5	21	18	33	128	23
RT(4,15,5,4)	259	263	85	5790	557
SEA	18	12	26	58	30
Hyperplane	136	55	186	700	208
KDDCup	23	24	33	212	616
Covtype	92	44	108	415	217

Results

- VFDR-MC algorithms correctly learn simpler nominal tasks
- VFDR-MC has improved accuracy for numerical and mixed sets
  - For multi-class problems due to one vs. all approach
  - Also for two-class problems due to the fact it can learn the concept faster by inducing more rules
- Using Naive Bayes within the rules facilitates faster learning curve and better any-time prediction capabilities
- Disadvantage is that VFDR-MC^u may produce large rule sets

Summary

- We introduced ordered and unordered `VFDR-MC` rule classifier for data streams
- Uses `FOIL` gain that distinguishes one class from all the others
- Hoeffding Bound guarantees with given confidence that new literal is the best one
- Achieves promising results on data with stationary distribution
- Has high potential for extensions to handle non-stationary data