Slide 1

Incremental Rule Learning with Partial Instance Memory for Changing Concepts

Mark Maloof

Department of Computer Science
Georgetown University
Washington, DC 20057-1232

maloof@cs.georgetown.edu
http://www.cs.georgetown.edu/~maloof

International Joint Conference on Neural Networks
Portland, OR

23 July 2003

Slide 2

Talk Overview

- Basics of rule learning
- On-line learning
- Concept memory and instance memory
- Partial instance memory and selecting examples
- Widmer and Kubat's Window Adjustment Heuristic
- Evaluation on the STAGGER Concepts

Slide 3

The Basics of Rule Learning

The AQ algorithm (Michalski, 1969)

1. Start with positive and negative training examples
2. Pick one positive training example
3. Form a rule by generalizing it as much as possible without "covering" a negative example
4. Remove the positive examples covered by the rule
5. Until covering all positive examples, goto Step 2
6. Repeat for the negative class, goto Step 2

Slide 4

Learning the Concept of "Who can vote"

- Attributes:
  - gender $\in$ { M, F }
  - age $\in$ { 1, . . . , 120 }
- Training examples:

gender	age	vote?
M	54	yes
F	42	yes
M	22	yes
F	32	yes
F	11	no
M	14	no
M	8	no
F	16	no

Slide 5

Learning the Concept of "Who can vote"

- Pick a positive example: $\langle$ M, 22, yes $\rangle$
- Rule: vote $\leftarrow$ [gender = M] & [age = 22]
- Generalize gender: vote $\leftarrow$ [gender = M $\vee$ F] & [age = 22]
- Cover any negative examples? No!
- Generalize age: vote $\leftarrow$ [gender = M $\vee$ F] & [age >= 22]
- Cover any negative examples? No!
- Generalize age: vote $\leftarrow$ [gender = M $\vee$ F] & [age > 16]
- Cover any negative examples? No!
- Final rule: vote $\leftarrow$ [age > 16]

Slide 6

On-line Learning

- Training examples distributed over time
- But system must always be able to perform
- Temporal-Batch Learning
  1. Learn rules from examples
  2. Store rules, store examples
  3. Use rules to predict, navigate, etc.
  4. When new examples arrive, add to current examples
  5. Goto step 1
- Incremental Learning
  1. Learn rules from examples
  2. Store rules, discard examples
  3. Use rules to predict, navigate, etc.
  4. When new examples arrive, learn new rules using old rules and new instances
  5. Goto step 2

Slide 7

Concept Memory

- Full: Learner stores concept descriptions, changing them only when new examples arrive (e.g., WINNOW)
- No: Learner stores no concept descriptions that generalize training examples (e.g., IB2)
- Partial: Learner stores concept descriptions and modifies them but not necessarily in response to the arrival of new training examples, like weight decay (e.g., FAVORIT)

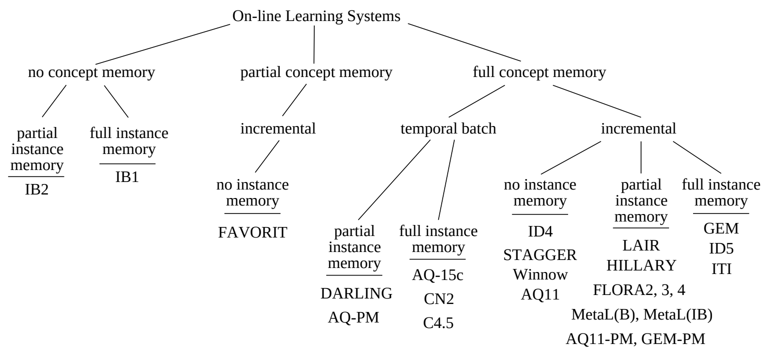
Slide 8

Instance Memory

- Full: Learner stores all examples from the input stream (e.g., ID5, GEM)
- No: Learner stores no examples (e.g., ID4, AQ11)
- Partial: Learner stores *some* examples (e.g., LAIR, HILLARY, FLORA, DARLING, METAL(B), METAL(IB), AQ-PM, AQ11-PM)

Slide 9

Classification of Learning Systems



Slide 10

Algorithm for Learning with Partial Instance Memory

1. Learn rules from training examples
2. Select a portion of the examples
3. Store rules, store examples
4. Use rules to predict, navigate, etc.
5. When new examples arrive
   - if incremental learning, then
     - learn new rules using old rules, new instances, and examples held in partial memory
   - if temporal-batch learning, then
     - learn new rules using new instances and examples held in partial memory
6. Combine new instances with those in partial memory
7. Goto step 2

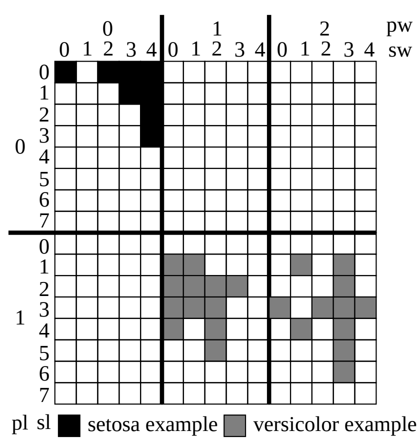
Slide 11

Selecting Examples for Partial Memory

- LAIR: the first positive example only
- HILLARY: only the negative examples
- DARLING: examples near the centers of clusters
- IB2: misclassified examples
- METAL(B), METAL(IB): sequence over a fixed window of time
- FLORA: sequence over a changing window, set adaptively
- AQ-PM, AQ11-PM, GEM-PM, AQ11-PM-WAH: examples on the boundaries of rules (i.e., *extreme examples*), possibly over a fixed window of time
  - for the rule: vote $\leftarrow$ [age $>$ 16]
  - extreme examples: $\langle$ F, 16, No $\rangle$ and $\langle$ M, 22, Yes $\rangle$
  - mark the boundary between the concepts 'Can Vote' and 'Cannot Vote'

Slide 12

Visualization of Training Examples: Discrete Version of the Iris Data Set



Slide 13

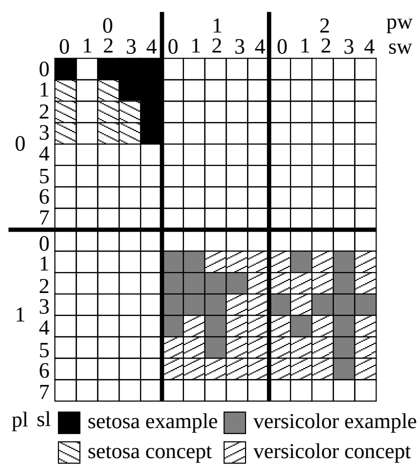
Induced Characteristic Rules

```
    setosa ← [pl = 0] & [pw = 0] &
             [sl = 0..3] & [sw = 0, 2..4]

versicolor ← [pl = 1] & [pw = 1..2] &
             [sl = 1..6] & [sw = 0..4]
```

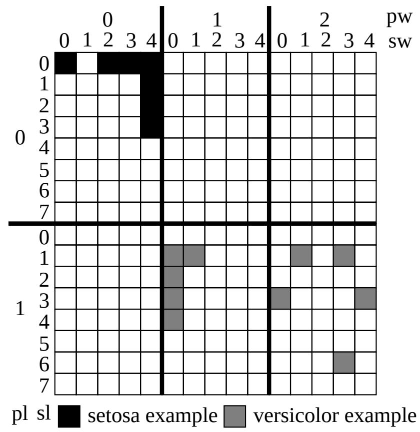
Slide 14

Visualization of Induced Rules



Slide 15

Visualization of Extreme Examples



Slide 16

Evaluation of Learning Systems

- AQ11: Incrementally learns disjunctive rules. No instance memory. A lesioned version of AQ11-PM.
- AQ11-PM: Incrementally learns disjunctive rules. Selects examples on the boundaries of these descriptions over a fixed window of time.
- AQ11-PM-WAH: AQ11-PM with a dynamically sized window of time using Widmer and Kubat's Window Adjustment Heuristic (WAH).
- AQ-PM: Temporal-batch learner. Disjunctive rules. Selects examples on the boundaries of these descriptions over a fixed window of time.

Slide 17

Widmer and Kubat's WAH

Window-Adjustment-Heuristic ()

lc: threshold for low coverage, user-defined
hc: threshold for high coverage, user-defined
p: threshold for acceptable accuracy, user-defined
N: examples covered by the positive concept description
S: number of conditions in the positive description
Acc: accuracy of current concept descriptions
w: window size

if $(N/S < lc \vee (Acc < p \wedge decreasing(Acc)))$
    $\Delta w = -0.2w;$
else if $(N/S > 2.0 \times hc \wedge Acc > p)$
    $\Delta w = -1.0;$
else if $(N/S > hc \wedge Acc > p)$
    $\Delta w = 0.0;$
else
    $\Delta w = 1.0;$
$w = w + \Delta w;$
end.

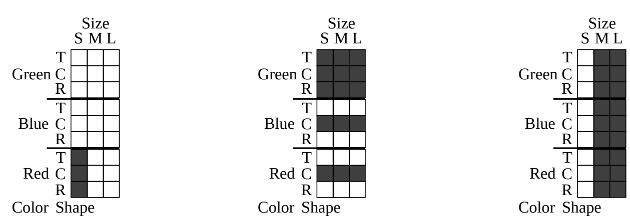
Slide 18

The STAGGER Concepts

`(size = small)` & `(color = red)`

`(shape = circle)` $\vee$ `(color = green)`

`(size = medium, large)`



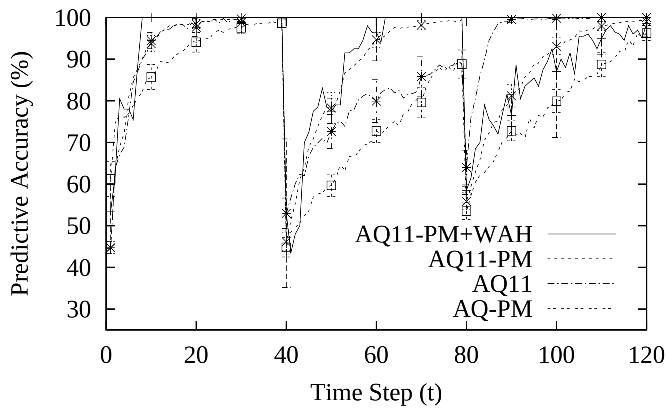
a. Target concept for time steps 1–39.

b. Target concept for time steps 40–79.

c. Target concept for time steps 80–120.

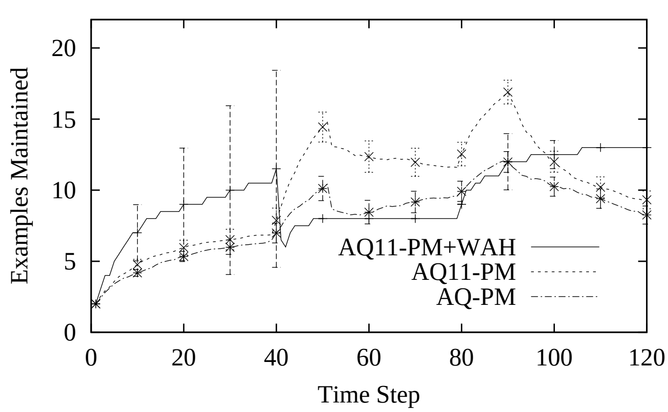
Slide 19

The STAGGER Concepts: Predictive Accuracy



Slide 20

The STAGGER Concepts: Memory Requirements



(AQ11 stores no examples.)

Slide 21

Observations

- Learners with partial instance memory (e.g., AQ11-PM-WAH, FLORA2) tend to outperform learners with no instance memory (e.g., STAGGER, AQ11)
- If the partial-memory learner can store the "right" examples, then when concepts change, it will have more examples for learning the new concept than will a learner that does not store examples
- Naturally, the trick is identifying the right examples
- Adding Widmer and Kubat's WAH to AQ11-PM slightly improved accuracy
- But it did improve AQ11-PM's generality since there's no fixed period for storing examples
- On the other hand, instead of one parameter for the length of the fixed window, we now have three parameters for the WAH

Slide 22

Current and Future Work

- Better characterization of performance using synthetic data sets: CNF, DNF, m-of-n, class noise, concept overlap
- Scale to larger data sets: Just acquired 10 GB of audit data
- Track changing concepts in real data sets
- Evaluate effect of skewed data
- Prove bounds for predictive accuracy and examples maintained
- Reinforcement learning to determine WAH parameters
- Ensemble methods for concept drift (Kolter & Maloof, 2003)

Slide 23

Incremental Rule Learning with Partial Instance Memory for Changing Concepts

Mark Maloof

Department of Computer Science
Georgetown University
Washington, DC

maloof@cs.georgetown.edu
http://www.cs.georgetown.edu/~maloof

International Joint Conference on Neural Networks
Portland, OR

23 July 2003

References

Aha, D., Kibler, D., & Albert, M. (1991). Instance-based learning algorithms. *Machine Learning*, *6*, 37–66.

Elio, R., & Watanabe, L. (1991). An incremental deductive strategy for controlling constructive induction in learning from examples. *Machine Learning*, *7*, 7–44.

Iba, W., Woogulis, J., & Langley, P. (1988). Trading simplicity and coverage in incremental concept learning. In *Proceedings of the Fifth International Conference on Machine Learning* (pp. 73–79). San Francisco, CA: Morgan Kaufmann.

Kolter, J., & Maloof, M. (2003). *Dynamic weighted majority: A new ensemble method for tracking concept drift* (Technical Report No. CSTR-20030610-3). Washington, DC: Department of Computer Science, Georgetown University. (http://daruma.georgetown.edu/techreports/techrep_data/cstr-20030610-3.pdf)

Kubat, M., & Krizakova, I. (1992). Forgetting and aging of knowledge in concept formation. *Applied Artificial Intelligence*, *6*, 195–206.

Littlestone, N. (1991). Redundant noisy attributes, attribute errors, and linear-threshold learning using Winnow. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory* (pp. 147–156). San Francisco, CA: Morgan Kaufmann.

Maloof, M. (1996). *Progressive partial memory learning.* Doctoral dissertation, School of Information Technology and Engineering, George Mason University, Fairfax, VA.

Maloof, M. (to appear). Incremental rule learning with partial instance memory for changing concepts. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN '03).* Los Alamitos, CA: IEEE Press.

Maloof, M., & Michalski, R. (2000). Selecting examples for partial memory learning. *Machine Learning*, *41*, 27–52.

Maloof, M., & Michalski, R. (2002). Incremental learning with partial instance memory. In *Foundations of intelligent systems* (Vol. 2366, pp. 16–27). Berlin: Springer-Verlag. (Proceedings of the Thirteenth International Symposium on Methodologies for Intelligent Systems, Lyon, France, June 27–29)

Maloof, M., & Michalski, R. (to appear). Incremental learning with partial instance memory. *Artificial Intelligence*.

Michalski, R. (1969). On the quasi-minimal solution of the general covering problem. In *Proceedings of the Fifth International Symposium on Information Processing* (Vol. A3, pp. 125–128).

Michalski, R., & Larson, J. (1983). *Incremental generation of VL_1 hypotheses: The underlying methodology and the description of program AQ11* (Technical Report No. UIUCDCS-F-83-905). Department of Computer Science, University of Illinois, Urbana.

Reinke, R., & Michalski, R. (1988). Incremental learning of concept descriptions: A method and experimental results. In J. Hayes, D. Michie, & J. Richards (Eds.), *Machine intelligence 11* (pp. 263–288). Oxford: Clarendon Press.

Salganicoff, M. (1993). Density-adaptive learning and forgetting. In *Proceedings of the Tenth International Conference on Machine Learning* (pp. 276–283). San Francisco, CA: Morgan Kaufmann.

Schlimmer, J., & Fisher, D. (1986). A case study of incremental concept induction. In *Proceedings of the Fifth National Conference on Artificial Intelligence* (pp. 496–501). Menlo Park, CA: AAAI Press.

Utgoff, P. (1988). ID5: An incremental ID3. In *Proceedings of the Fifth International Conference on Machine Learning* (pp. 107–120). San Francisco, CA: Morgan Kaufmann.

Widmer, G. (1997). Tracking context changes through meta-learning. *Machine Learning*, *27*, 259–286.

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, *23*, 69–101.