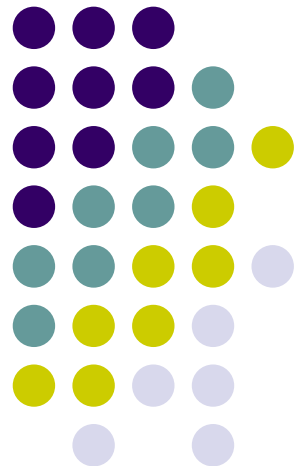
Hierarchical Learning

Oren B. Yeshua (oby1)

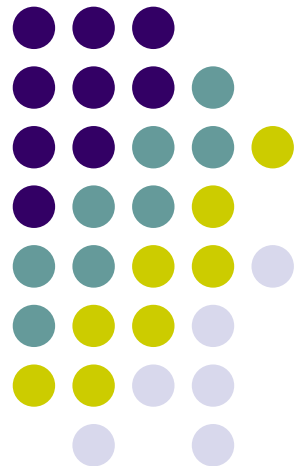
CS 6253 – Spring 2007



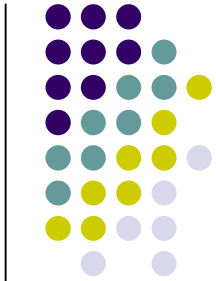
Hierarchical Learning

Oren B. Yeshua (oby1)

CS 6253 – Spring 2007

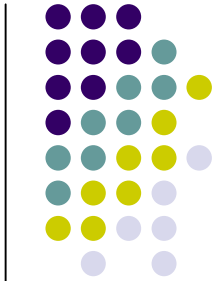


Agenda



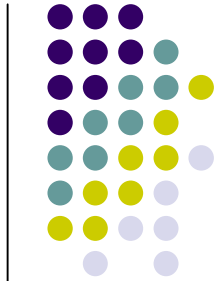
- Hierarchical Learning – what/why?
- A general model
- Constraining the model
  - Two concerns
    - Learning structure
    - Learning from induced concepts
      - Small subconcepts – CSL algorithm
      - Larger subconcepts – hard
  - Learner/Teacher tradeoff
- Open Questions
- Practical Considerations

Motivation



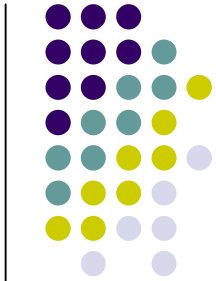
- General intelligence / cognitive computing
- Use learning as we know it as a building block
  - To learn intractable concept classes
  - To learn continuously from the environment
- ML/Cognitive Computing community often makes assumptions along these lines
  - Examples
    - Utgoff / Stracuzzi
      - "[Labels for every subconcept with every example] may be more information than is strictly necessary…a matter of communication efficiency…not a major concern of ours…provide [all truth values] avoiding all the problems related to discourse."

Motivation (continued)



- Valiant – Neuroidal Model
  - "If the system consists of a chain of circuit units trained in sequence in the above manner, then the errors in one circuit *need not propagate* to the next. Each circuit will aim to be accurate in the PAC sense as a function of the external inputs–the fact that intermediate levels of gates only approximate the functions that the trainer intended is not necessarily harmful as long as each layer relates to the approximations at the previous layer in a robustly learnable manner. At each internal level, these internal feature sets may nevertheless permit accurate PAC learning at that next stage. That this is indeed possible for natural data sets remains to be proved. Some analysis of this issue of hierarchical learning has been attempted [cites Rivest and Sloan 1994]"
  - For more: http://halcyon.googlepages.com/CLT

A Model for Hier. Learning



- CLT hardness results indicate more information is necessary to learn hard concept classes
  - Provide more labels
- Un-learnable target concept $c^* \in C$ broken into polynomially many learnable subconcepts $y_1,\ldots,y_s$
- Example oracles $EX^D, EX_1, EX_2, EX_3, \ldots, EX_s$ provided to the learner
  - EX^D: draws $x \in X$ at random from distribution D
  - EX_i: can be "chained" to EX^D to compute $y_i(x)$
- Learner must output $h \in H$ that predicts c^*

Constraining the Model

- Learn in sequence of L *lessons,* $1 \leq L \leq s$
  - Lesson 1: $EX_1,\ldots,EX_{d1}$ is provided
  - Lesson 2: $EX_{d1+1},\ldots,EX_{d2}$ is provided
  - …
  - Lesson L: $Ex_{d(L-1)+1},\ldots,EX_{dL}$ is provided
    - Can further constrain setting L=s
      - Learn one concept per lesson
  - Learner decides when to advance to next lesson
  - EX oracles from previous lesson are recalled
    - Learner can request example from past lesson (penalty?)
- L may or may not be provided to the learner
- Constrain size/type of subconcept classes
- PAC – h must be an ε,δ-approximation of c*

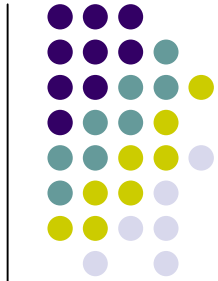
Two Concerns

- Learning structure
  - How do we organize the concept hierarchy? What should we learn from what?
  - Nature/environment/inputs/learnability impose structure
    - Simple strategy: learn whatever you can from current input and existing knowledge
      - Hypothesis testing
    - STL – Stream to Layers algorithm
      - Ugtoff & Stracuzzi
      - Tested w/ 1-hidden layer NN subconcept learners
      - Toy concept classes
        - Card stackability
        - Two-Clumps
- Learning from induced concepts
  - Theoretically challenging
    - No simple strategy
  - Required for learning structure (but not vice versa)
  - Crucial for cognitive systems
  - CSL
    - Rivest & Sloan
  - Big open question

The ~~Problem~~ Challenge

- Lessons above the first have attributes that may not match true value
  - Some attributes are approximations to true attributes computed by previously learned subconcepts
  - Attributes appear “noisy” with noise rate = error rate of subconcept hypothesis
  - Noise accumulates as number of learned subconcepts increases

A Small Step



- When subconcept size is small (polynomial) hierarchical learning is possible/easy/robust to noise (Rivest & Sloan ‘94)
  - Strong result / constrained setting
- Constraints:
  - Subconcept classes H_i, $|H_i| \leq K = \text{poly}(n)$
  - Limit to one subconcept per lesson
  - EX oracle recalled after lesson
- Key insight
  - Maintain version-space of hypotheses F_i for each y_i
    - Enables accurate examples at each lesson via “filtering”
    - Enables *reliable* and *probably useful* learning (stronger than PAC)

The CSL Algorithm

```
CSL(ε,δ,s)
    ε′← ε/sK
    δ′← δ/2s
    m ← ln(K/δ′)/ε′
    for i ← 1 to s
        get EX_i              # advance to next lesson
        for j ← 1 to 2m # create filtered sample
            (x,y) ← Push-Button(EX_i) # draw exmple
            if F_1,…,F_i-1 agree on x
                  add (x,y) to sample
            if size(sample) = m
                  break
        if size(sample) < m
            return FAIL   # not enough examples
        else
            F_i ← all hyp consistent w/ sample
    return F_1,…,F_s
```

CSL Intuition

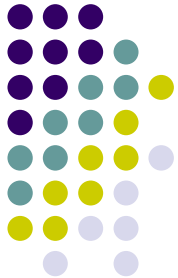
- Since target concept must be in F_1 , common output must be correct for given example when all $f \in F_1$ agree on x

F_1	f(x)
f_1	0
f_2	0
f_5	0
f_9	0

F_1 *agrees* on x

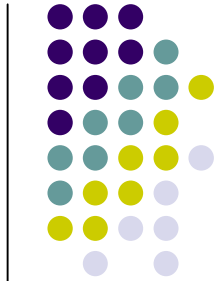
$y_1(x) = F_1(x) = 0$

CSL Intuition (continued)



- If $F_1,\ldots,F_{i-1}$ agree on x, then
  - $F_1(x),\ldots,F_{i-1}(x) = y_1(x),\ldots,y_{i-1}(x)$
- Can learn F_i from x, $y_1(x),\ldots,y_{i-1}(x)$
- $h(x,F_1,\ldots,F_s)$ is reliable
  - If CSL returns FAIL then abstain
  - If $F_1,\ldots,F_s$ agree on x, output $F_s(x) = y_s(x) = c^*(x)$
  - Else abstain
- h is probably useful: w/ probability $1-\delta$,
  - CSL doesn't FAIL
  - $Pr_{x \in D}[F_1,\ldots,F_s$ agree on x] $> 1-\varepsilon$

Taking a Bigger Bite

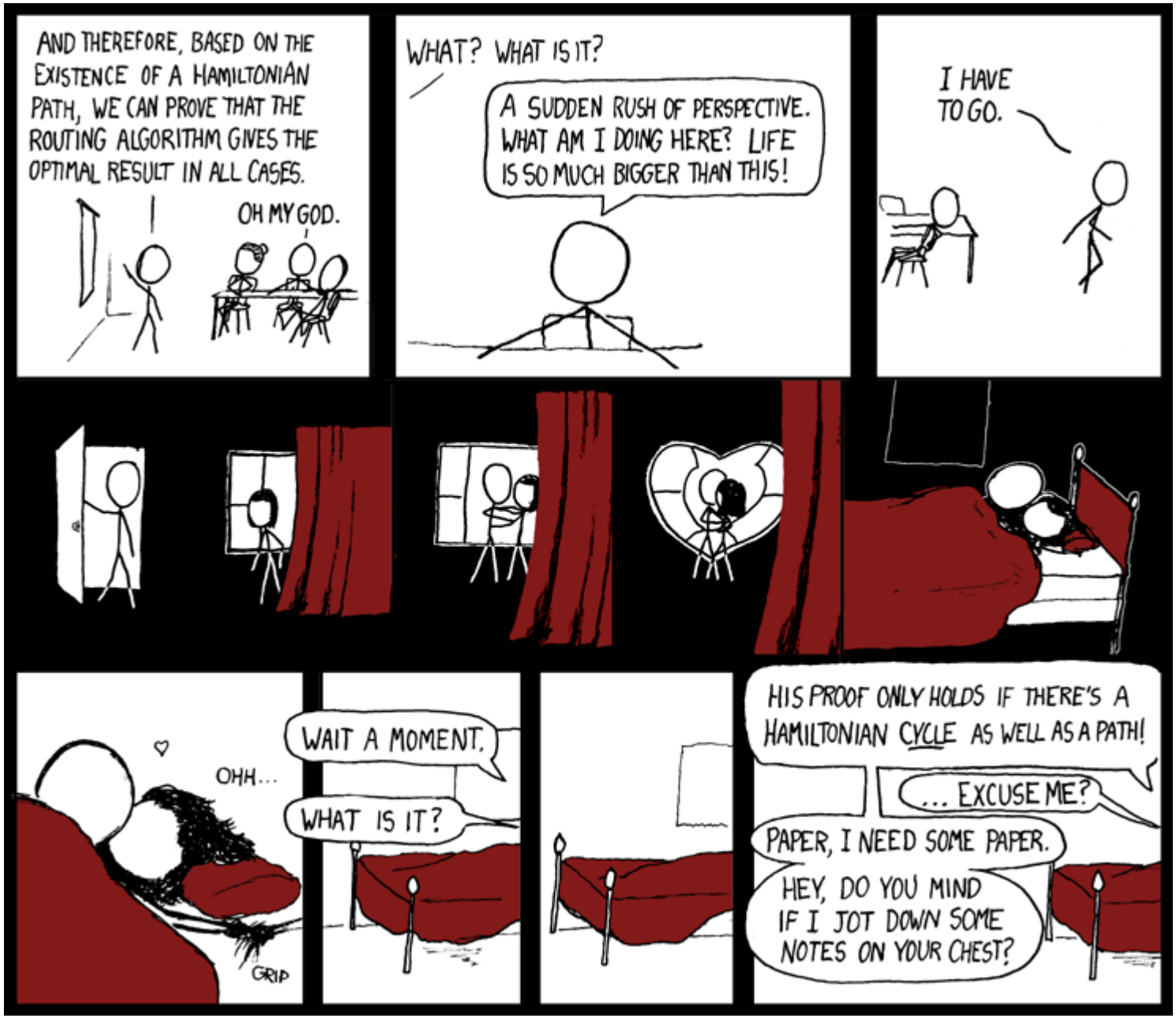


- CSL analysis suggests investigating attribute noise
- Sloan (’95) seems to have gone down this path
  - Results
  - Implications
    - Unlikely

	Monomials	k-DNF
E_{MAL}^{poly}	$\geq \varepsilon/n \ln(n/\varepsilon)$	$\geq \varepsilon/n^k \ln(n^k/\varepsilon)$
E_{MAL}	$< \varepsilon/(1+\varepsilon)$	$< \varepsilon/(1+\varepsilon)$
E_{URA}^{poly}	$\geq ½-\omega$, $\omega>0$	open
E_{URA}	$< 1/2$	$< 1/2$
E_{PRA}	$< 2\varepsilon$	$< 2\varepsilon$

Open Questions

- What is the largest subconcept class learnable in a hierarchical setting?
  - How?
- Can the *right* teacher/learner (model constraints) tradeoff enable learning of larger subconcepts?
  - Message passing between nodes
    - Same depth
    - Different depth
      - Feedback
- Practical Considerations
  - Promising results have been reported in practice
  - Are the theoretical models too focused on “worst-case” analysis?
  - Accuracy on attribute noise is measured with respect to a noise-free test set. In practice, all data is noisy – should measure with respect to noisy input data
    - New noise models?



Courtesy of:

xkcd

"The problem with perspective is that it's bi-directional."