# Learning from Snapshot Examples

Jacob Beal MIT CSAIL April, 2005









• Space is cluttered with objects



• Space is cluttered with objects











# Snapshot Learning Framework



- Bootstrapping feedback cycle
  - better model  $\rightarrow$  better examples  $\rightarrow$  better model

# Snapshot Learning Framework



- What are the targets?
- How can it choose good examples?

### Targets

"Lemon" would be best, settle for its components

- Each percept is a target
- Learn each target independently

This means we'll learn each association several times



# Examples from Samples

Input is DT sampling of evolving perceptual state

- Incrementally select examples from samples
- Can only learn about things coextensive in time *Solvable by buffering w. short term memory*

### Relevance of a Sample

- Create a relevance measure for each channel
  - High-relevance should indicate useful content



#### Sparseness Assumptions

At the right level of abstraction, the world is sparse

• Percepts are sparse across time *most of life doesn't involve lemons* 



• Percepts are sparse at each sample *most of life doesn't appear when the lemon does* 

# Sparseness→ Irrelevant periods



#### Lots of irrelevant periods $\rightarrow$ lots of relevant periods

# Be choosy!



Time

- Many chances  $\rightarrow$  take only the best
  - a few good >> many iffy
  - avoid overfitting from closely correlated examples

Relevance peaks?

### Are peaks a good idea?

Consider the relevance measures as signals:



Projecting to a single measure loses a lot of info...

# Top-Cliff Heuristic

- Generalizing "peak" to multiple dimensions
  - Some channel's relevance is falling
  - No channel's relevance is rising
  - All *relevant channels* have risen since their last drop



# **Top-Cliff Examples**



# Experiment: Learning from Examples



- Sequence of randomly generated examples
- Transition between examples in random order



- Sequence of randomly generated examples
- Transition between examples in random order



- Sequence of randomly generated examples
- Transition between examples in random order



- Sequence of randomly generated examples
- Transition between examples in random order



- Sequence of randomly generated examples
- Transition between examples in random order



- Sequence of randomly generated examples
- Transition between examples in random order

# Applying Snapshot Learning

- Target Model: {possible associate, confidence}
- Modified Hebbian Learning
- Relevance = # of possible associates present
- Extra virtual channel for target percept
  - Relevance 1 if present, 0 if absent
  - Determines if example is positive or negative

# Modified Hebbian Learning

- Initial set: percepts from first relevant period
  Late entry is possible but difficult
- Examples adjust confidence levels
  - Positive Example: +1 if present, -1 if absent
  - Negative Example: -1 if present, 0 if absent
  - Confidence  $\langle P \rightarrow prune \text{ out associate}$ !
    - Same channel as target are harder to prune
    - If no associates, restart

### **Experimental Parameters**

- 50 features
- 2 channels
- 1 percept/feature/channel = 100 targets
- Randomly generated examples, 2-6 features/exa
- Random transition between examples

#### Top-Cliff vs. Controls



• 10 trials of 1000 examples each

#### Predictable Variation w. Parameters



#### Resilient to Adverse Conditions



#### ...much more than the controls...



# Experiment: Learning w/o a Teacher

What if there's no teacher providing examples?

- A teacher guarantees there are associations...
- ... but *world* has lots of structure!
- Without a teacher, the system will still find targets and examples.

Will they teach it anything?

# 4-Way Intersection Model

- 5 locations (*N*,*S*,*E*,*W*,*Center*)
- 11 types of vehicle (Sedan, SUV, etc.)
  - Cars arrive randomly, with random exit goals.
  - Arrive moving, but queue up if blocked.
  - Moving or starting moving takes 1 second.
  - Left turns only when clear.
- 6 lights (NS-red, EW-green, etc.)
  - 60 second cycle: 27 green, 3 yellow, 30 red
  - Go on green, maybe yellow, right on red when clear.

#### Intersection Percepts

- 6 channels: N, S, E, W, Center, Light
  - Cardinal directions: type of 1<sup>st</sup> in queue, exiting cars
  - Center: types of cars there
  - Light: two active lights
- Distinguishable copy of previous percepts
- Random transitions, as before

(L NS\_GREEN EW\_RED PREV\_NS\_GREEN PREV\_EW\_RED)
(N) (S PREV\_CONVERTIBLE) (C CONVERTIBLE)
(E SEDAN PREV\_SEDAN) (W COMPACT PREV\_COMPACT)

#### What does it learn?

- After 16 light cycles:
  - Lights don't depend on cars
  - Stoplight state transitions (97% perfect)

**EW\_GREEN** = PREV\_NS\_RED, PREV\_EW\_GREEN, PREV\_NS\_YELLOW, NS\_RED **EW\_YELLOW** = PREV\_EW\_YELLOW, NS\_RED, PREV\_EW\_GREEN, PREV\_NS\_RED **EW\_RED** = NS\_YELLOW, PREV\_EW\_RED, PREV\_NS\_GREEN, NS\_GREEN **NS\_GREEN** = PREV\_EW\_RED, PREV\_NS\_GREEN, EW\_RED, PREV\_EW\_YELLOW **NS\_YELLOW** = PREV\_NS\_YELLOW, EW\_RED, PREV\_NS\_GREEN, PREV\_EW\_RED **NS\_RED** = PREV\_NS\_RED, PREV\_EW\_GREEN, EW\_GREEN, PREV\_NS\_YELLOW

PREV\_EW\_GREEN = PREV\_NS\_RED, NS\_RED, EW\_GREEN
PREV\_EW\_YELLOW = PREV\_NS\_GREEN, PREV\_NS\_RED, NS\_GREEN EW\_RED
PREV\_EW\_RED = PREV\_NS\_YELLOW, NS\_YELLOW, EW\_RED, NS\_GREEN, PREV\_NS\_GREEN
PREV\_NS\_GREEN = PREV\_NS\_YELLOW, NS\_YELLOW, PREV\_EW\_RED, EW\_RED, NS\_GREEN
PREV\_NS\_YELLOW = EW\_GREEN, NS\_RED, PREV\_EW\_RED, NS\_YELLOW
PREV\_NS\_RED = PREV\_EW\_RED, EW\_RED, PREV\_EW\_YELLOW, NS\_GREEN

#### Reconstructed FSM



# Summary

- Snapshot learning simplifies a hard problem
  - Top-Cliff finds sparse examples incrementally
  - Feedback improves quality of examples over time
  - It's easier to find good examples for single targets
- Snapshot learning works for sequences of examples or a predictably evolving state
- Pretending there's a teacher helps learn!