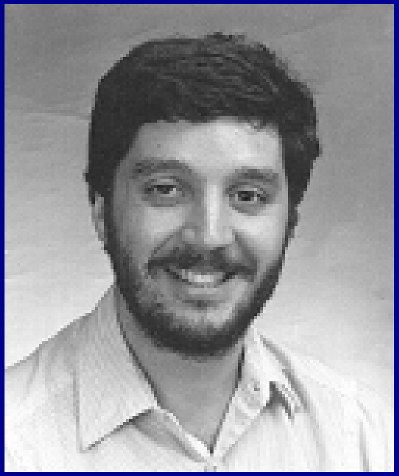
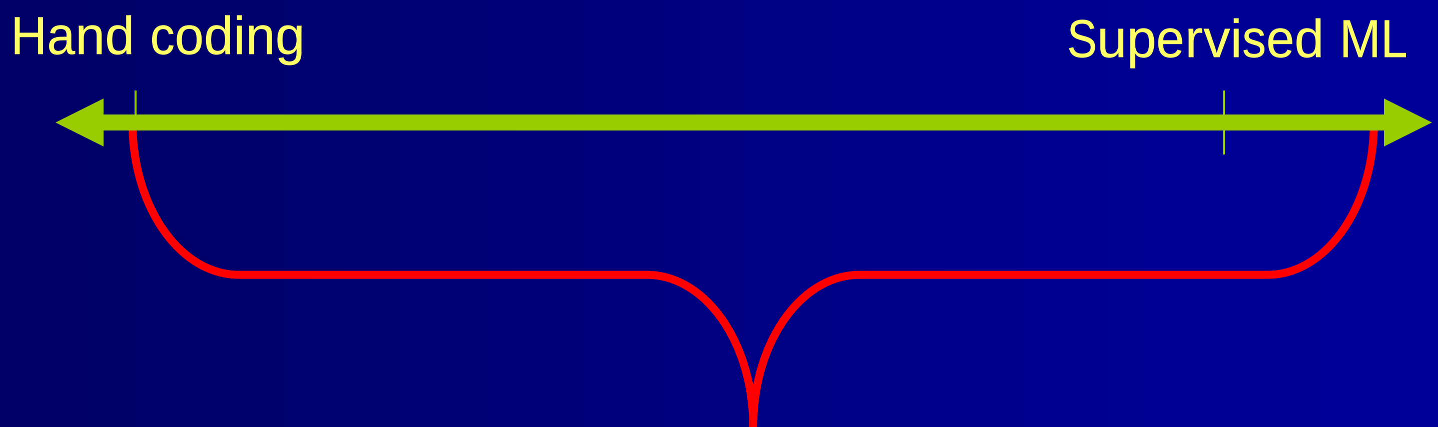
Twenty-Five Years of Combining Symbolic and Numeric Learning

Jude Shavlik
University of Wisconsin
USA



Key Question of AI:
How to Get Knowledge into Computers?



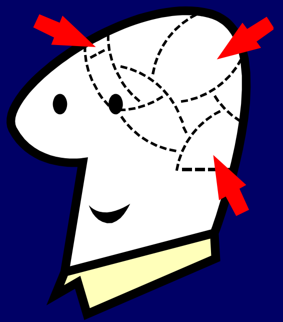
How can we mix these two extremes?

Neural-symbolic community (and others) have looked at ways to do so

Two Underexplored Questions in ML

- How can we go beyond teaching machines solely via I/O pairs?
  - ‘advice giving’
- How can we understand what an ML algorithm has discovered?
  - ‘rule extraction’

Outline



- Explanation-Based Learning (1980's)
- Knowledge-Based Neural Nets (1990's)
- Knowledge-Based SVMs (2000's)
- Markov Logic Networks (2010's)

Explanation-Based Learning (EBL) – My PhD Years, 1983-1987

- The EBL Hypothesis

  By understanding *why* an example is a member of a concept, can learn essential properties of the concept

- Trade-Off

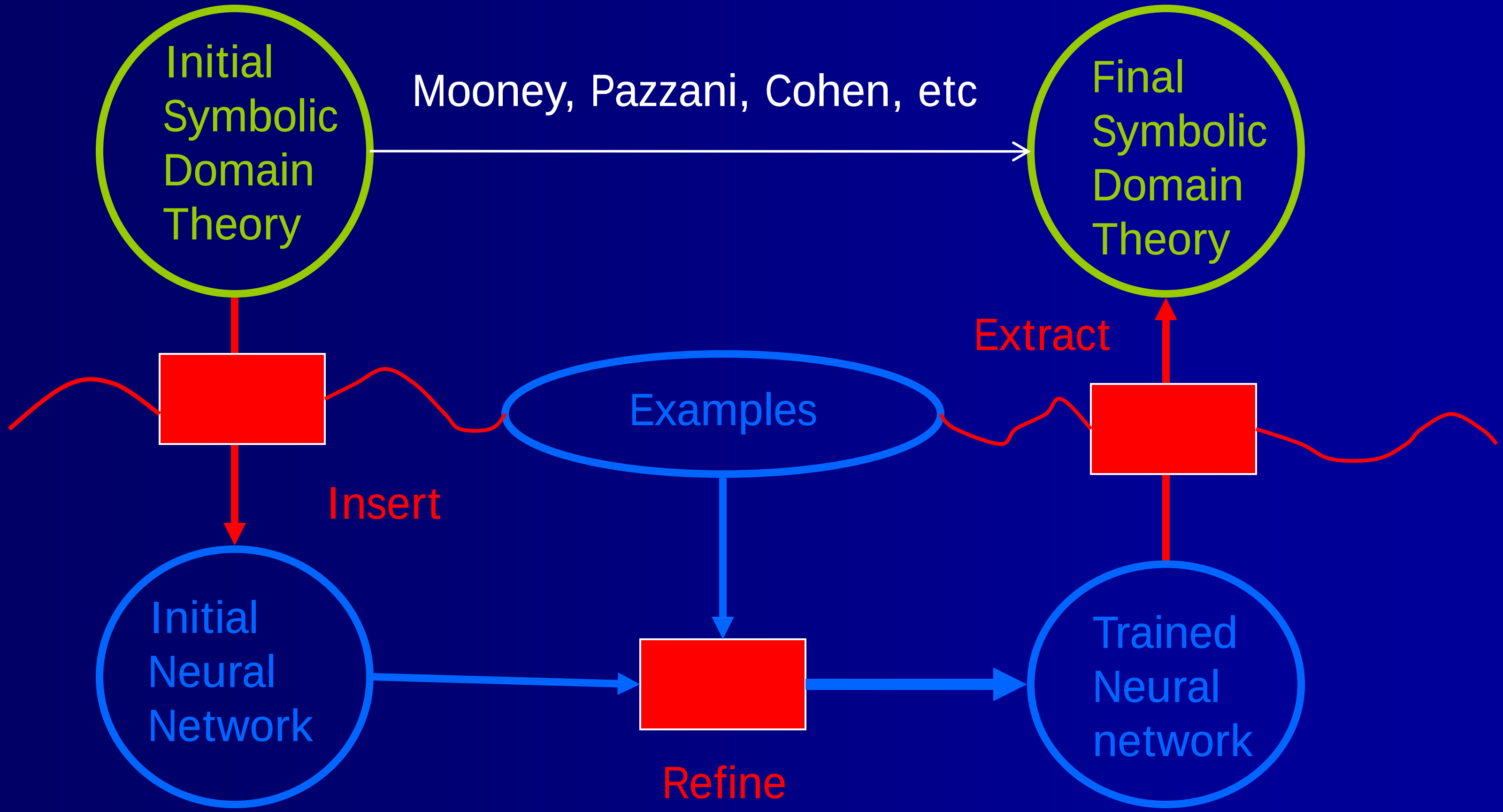
  The need to collect many examples

  for

  The ability to 'explain' single examples (a 'domain theory')

Ie, assume a smarter learner

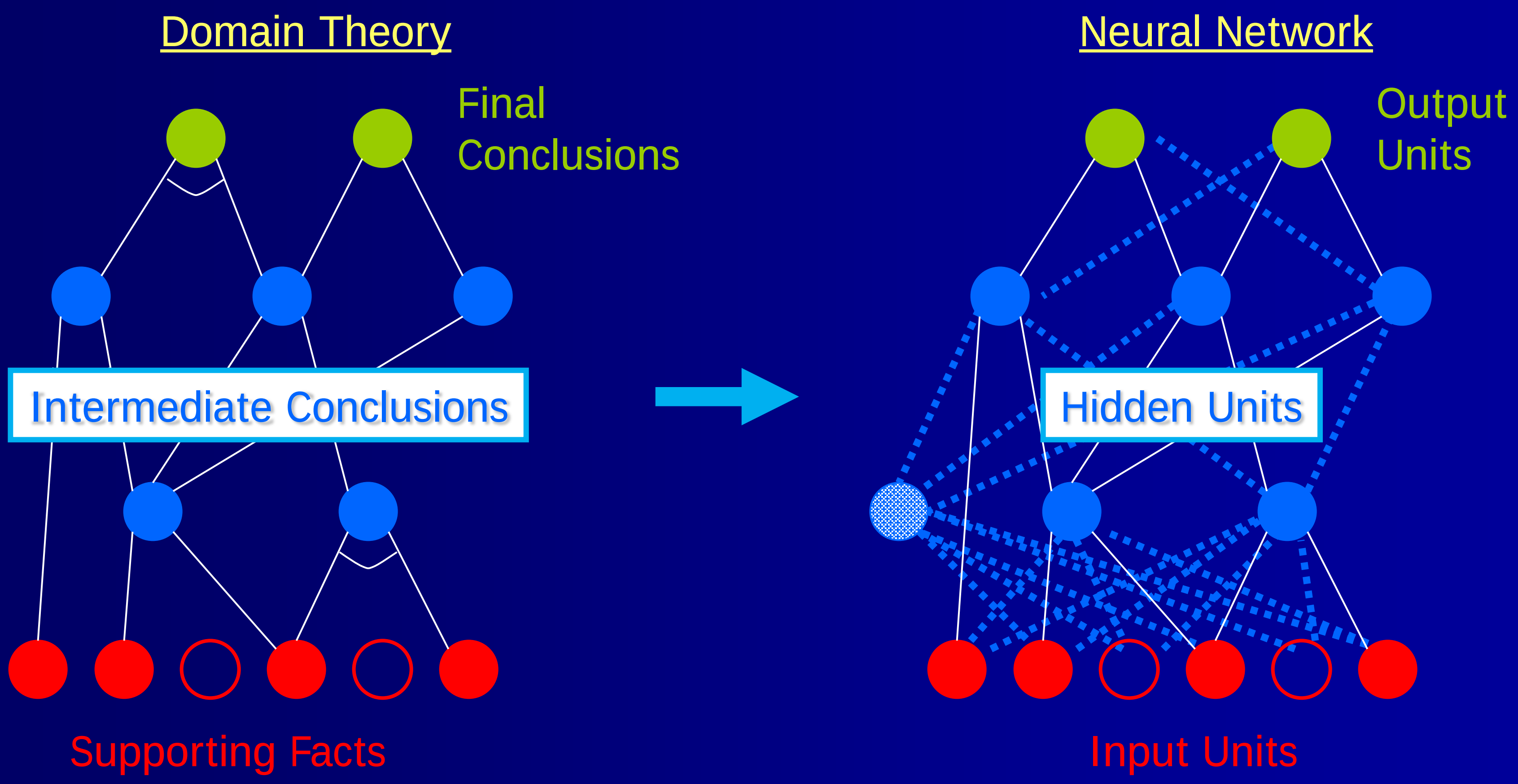
Knowledge-Based Artificial Neural Networks, KBANN (1988-2001)



What Inspired KBANN?

- Geoff Hinton was an invited speaker at ICML-88
- I recall him saying something like "one can backprop through <u>any</u> function"
- And I thought "what would it mean to backprop through a domain theory?"

Inserting Prior Knowledge into a Neural Network



Jumping Ahead a Bit

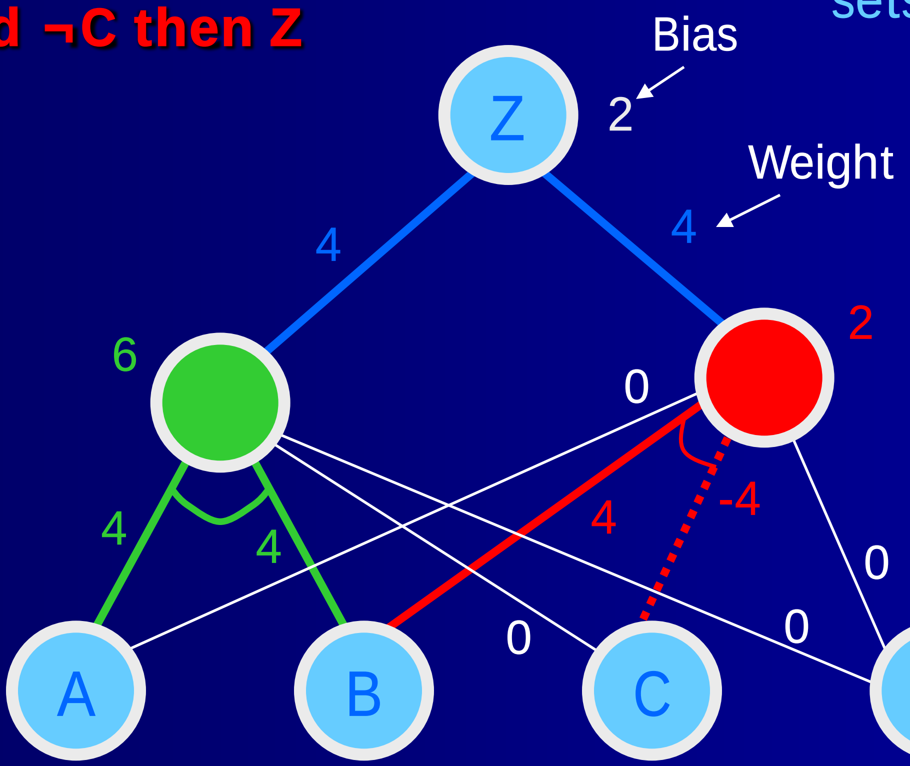
- Notice that symbolic knowledge <u>induces a graphical model</u>, which is then numerically optimized
- Similar perspective later followed in Markov Logic Networks (MLNs)
- However in MLNs, symbolic knowledge expressed in <u>first-order</u> logic

Mapping Rules to Nets

If A and B then Z

If B and ¬C then Z

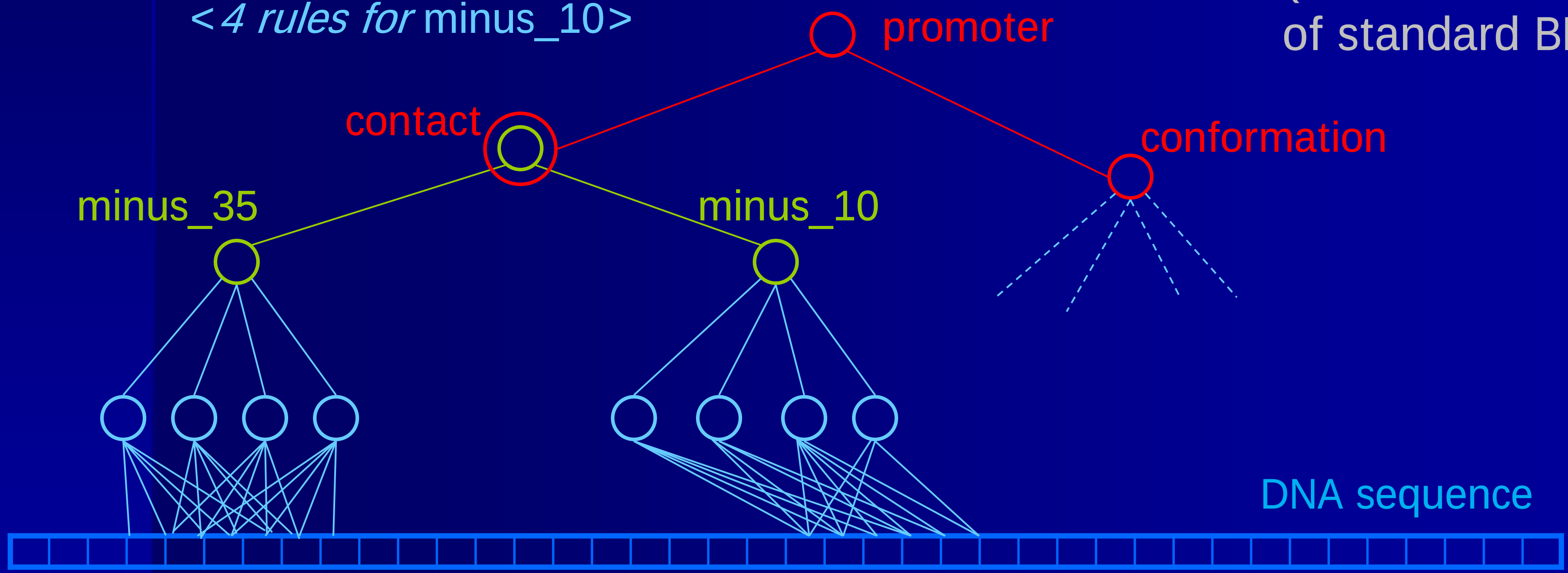
Maps propositional rule sets to neural networks



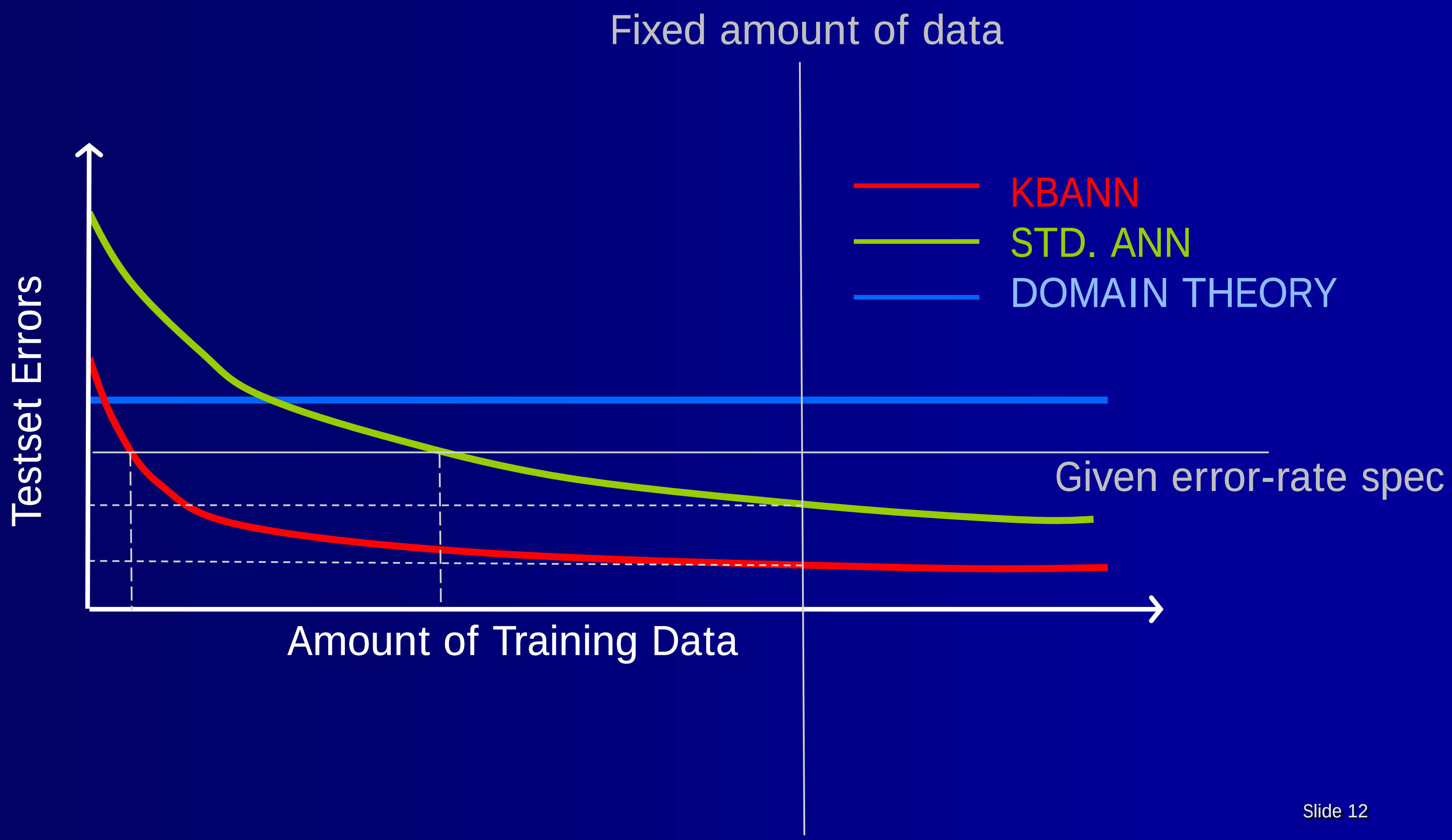
Case Study:
Learning to Recognize Genes
(Towell, Shavlik & Noordewier, AAAI-90)

promoter :- contact, conformation.
contact :- minus_35, minus_10.
<*4 rules for* conformation>
<*4 rules for* minus_35>
<*4 rules for* minus_10>

(Halved error rate of standard BP)



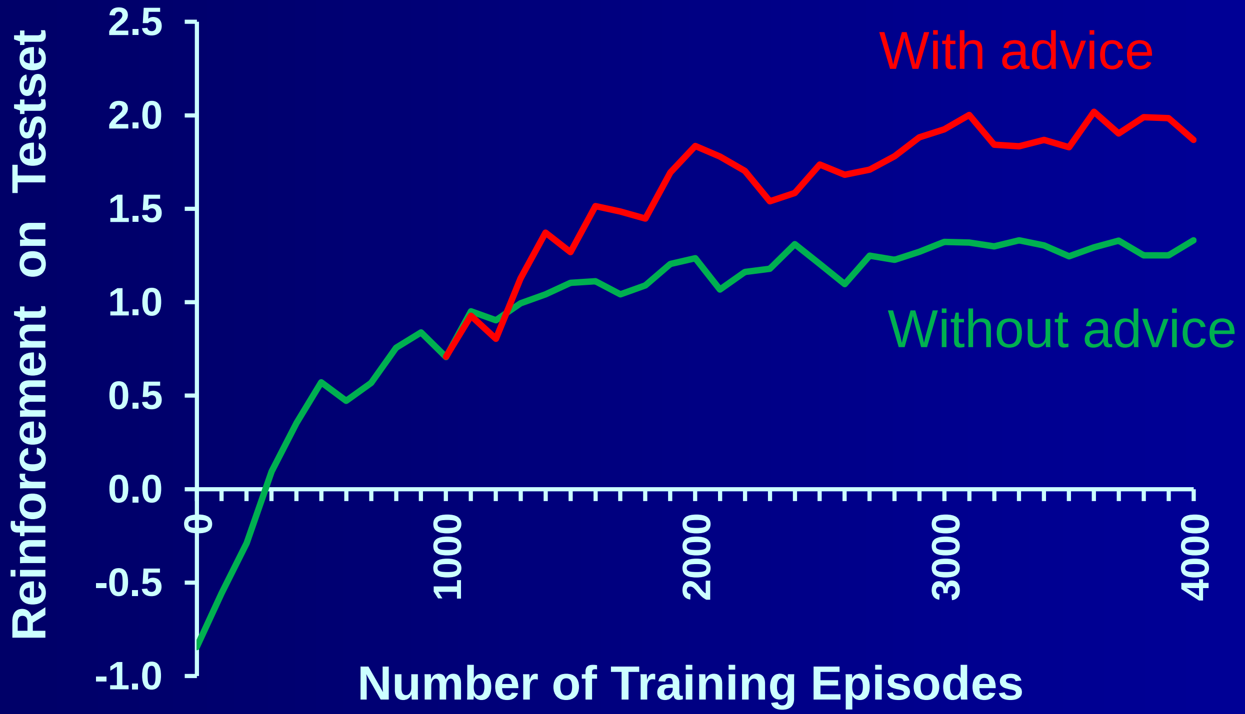
Learning Curves
(similar results on many tasks and with other advice-taking algo's)



From *Prior Knowledge* to *Advice* (Maclin PhD 1995)

- Originally ‘theory refinement’ community assumed domain knowledge was available <u>before</u> learning starts (<u>prior</u> knowledge)
- When applying KBANN to reinforcement learning, we began to realize
  - you should be able to provide domain knowledge to a machine learner <u>whenever</u> you think of something to say
- Changing the metaphor:
  - commanding vs. advising computers
- Continual (ie, lifelong) Human Teacher – Machine Learner Cooperation

Some Sample Results



Rule Extraction

- Initially Geoff Towell (PhD, 1991) viewed this as simplifying the trained neural network (M-of-N rules)
- Mark Craven (PhD, 1996) realized
  - This is simply another learning task!
  - Ie, <u>learn</u> what the neural network computes
    - Collect I/O pairs from trained neural network
    - Give them to decision-tree learner
  - Applies to SVMs, decision forests, etc

KBANN Recap

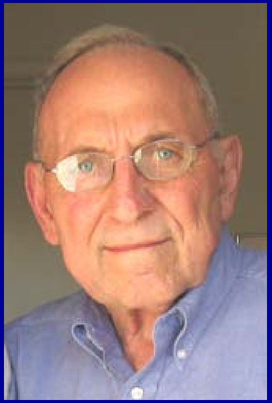
- Use symbolic knowledge to make an initial guess at the concept description

  Standard neural-net approaches make a *random* guess

- Use training examples to refine the initial guess ('early stopping' reduces overfitting)
- Nicely maps to incremental (aka online) learning
- Valuable to show user the learned model expressed in symbols rather than numbers

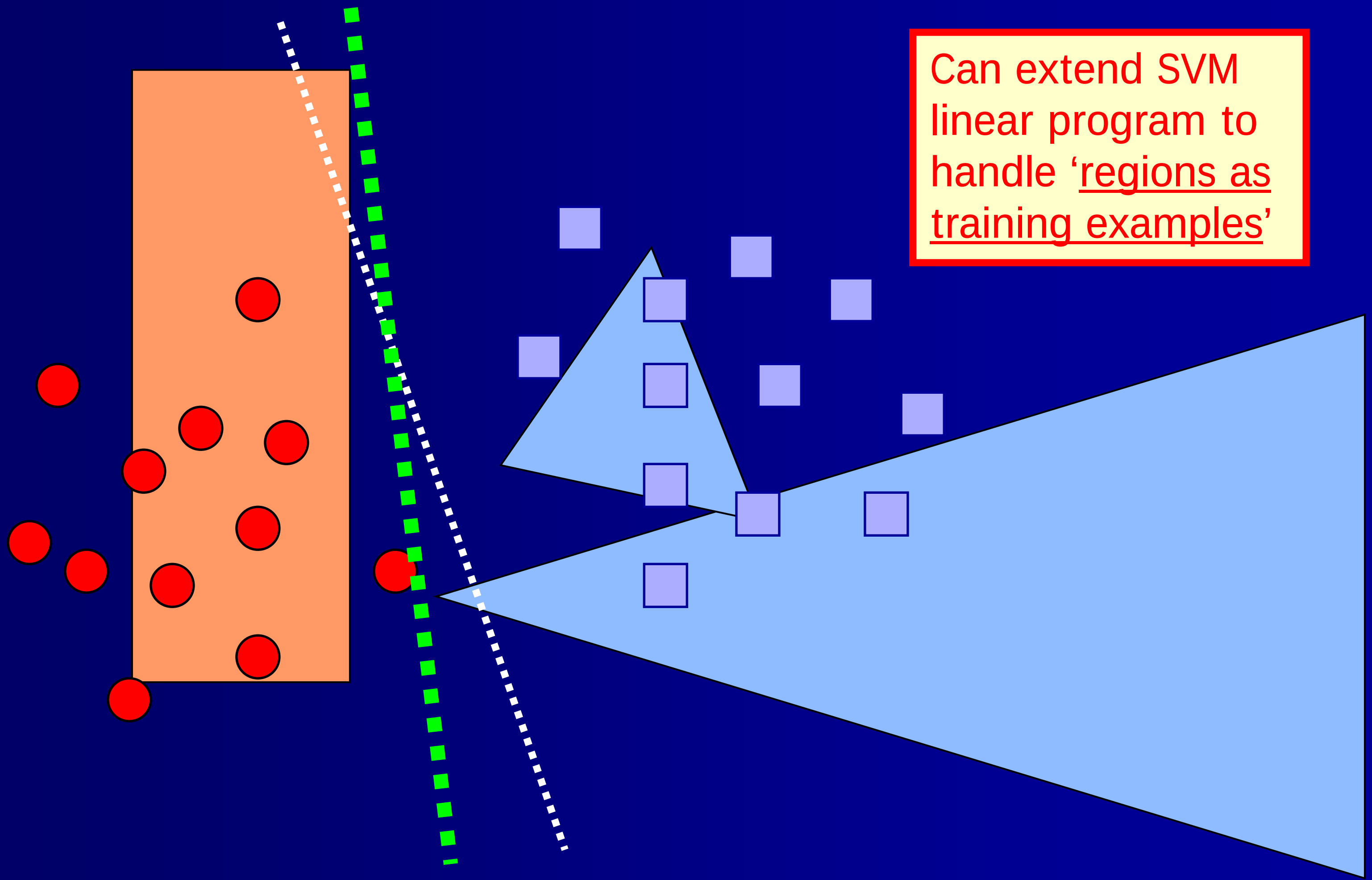
Knowledge-Based Support Vector Machines (2001-2011)

- Question arose during 2001 PhD defense of Tina Eliassi-Rad
  - How would you apply the KBANN idea using SVMs?



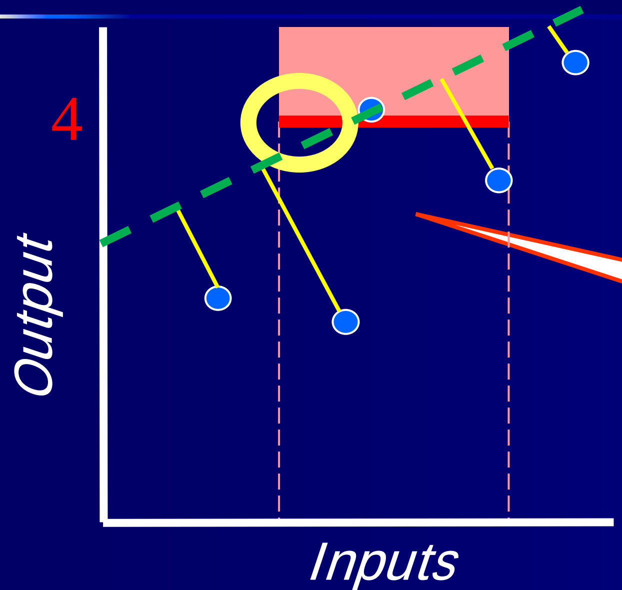
- Led to collaboration with Olvi Mangasarian (who has worked on SVMs for about 50 years!)

Generalizing the Idea of a Training Example for SVM's



Can extend SVM linear program to handle '<u>regions as training examples</u>'

Knowledge-Based Support Vector *Regression*



Add soft constraints to linear program (so need only follow advice *approximately*)

Advice: In this region, *y* should exceed 4

minimize	$\|\|w\|\|_1 + C\|\|s\|\|_1$
	+ *penalty for violating advice*
such that	$f(x) = y \pm s$
	constraints that represent advice

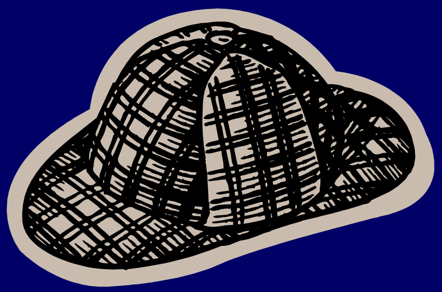
Automatically Creating Advice

Interesting approach to *transfer learning* (Lisa Torrey, PhD 2009)

So advice giving is done by MACHINE!

- Learn in task A
- Perform 'rule extraction'
- Give as advice for related task B
- Since advice not assumed 100% correct, differences between tasks A and B handled by training ex's for task B

KBSVM Recap



- Can view symbolic knowledge as a way to label <u>regions</u> of feature space (rather than solely labeling <u>points</u>)
- Maximize

  Model Simplicity

  \+ Fit to Advice

  \+ Fit to Training Examples

- Note: does <u>not</u> fit view of "guess initial model, then refine using training ex's"

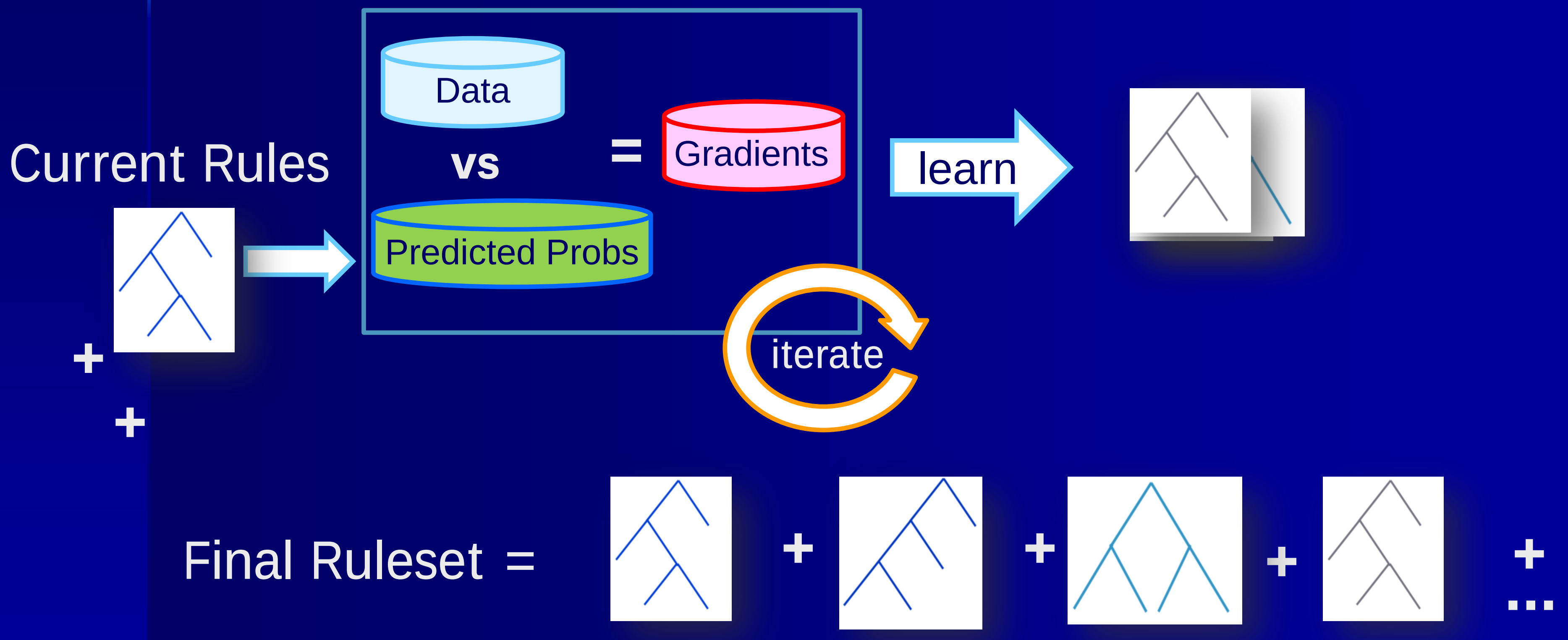
Markov Logic Networks, 2009+
(and statistical-relational learning in general)

- My current favorite for combining symbolic knowledge and numeric learning
- MLN = set of <u>weighted</u> FOPC sentences

  wgt=3.2 $\forall$x,y,z parent(x, y) $\wedge$ parent(z, y) $\rightarrow$ married(x, z)

- Have worked on speeding up MLN inference (via RDBMS) plus learning MLN rule sets

Learning a Set of First-Order Regression Trees (each path to a leaf is an MLN rule) – ICDM '11



Some Results

advisedBy	AUC-PR	CLL	Time
MLN-BT	0.94 ± 0.06	-0.52 ± 0.45	18.4 sec
MLN-BC	0.95 ± 0.05	-0.30 ± 0.06	33.3 sec
Alch-D	0.31 ± 0.10	-3.90 ± 0.41	7.1 hrs
Motif	0.43 ± 0.03	-3.23 ± 0.78	1.8 hrs
LHL	0.42 ± 0.10	-2.94 ± 0.31	37.2 sec

Differences from KBANN

- Rules involve logical variables
- During learning, we create new rules to correct errors in initial rules
- To do: also *refine* initial rules
  (note that KBSVMs also do NOT *refine* rules, though we had one AAAI paper on that)

Wrapping Up

- Symbolic knowledge refined/extended by
    - Neural networks
    - Support-vector machines
    - MLN rule and weight learning

Applications in genetics, cancer, machine reading, robot learning, etc

- Variety of views taken
    - Make initial guess at concept, then <u>refine</u> weights
    - Use advice to label a <u>region</u> in feature space
    - Make initial guess at concept, then <u>add</u> wgt’ed rules
- Seeing what was learned – rule extraction

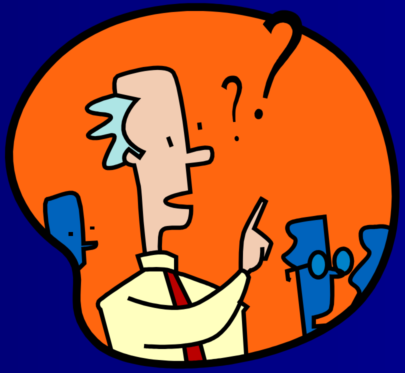
Some Suggestions for Neural-Sym Community

- Allow humans to continually observe learning and provide symbolic knowledge at any time
- Never assume symbolic knowledge is 100% correct
- Allow user to see what was learned in a symbolic representation to facilitate additional advice
- Replace ‘neural’ with ‘numeric’

- Put a graphic on every slide ☺

Thanks for Your Time

- Questions?



- Papers, Powerpoint presentations, and some software available on line

  pages.cs.wisc.edu/~shavlik/mlrg/publications.html

  hazy.cs.wisc.edu/hazy/tuffy/

  pages.cs.wisc.edu/~tushar/rdnboost/index.html