Faster Treatment Learning

By Ryan Clark

Preface

- Preface
- What is Treatment Learning?
- How Can Treatment Learning be Improved?
- Tar4.0: Can Bayes Help Tar4?
- Tar4.1
- Experiments
- Future Work & Conclusion

What inspired this work?

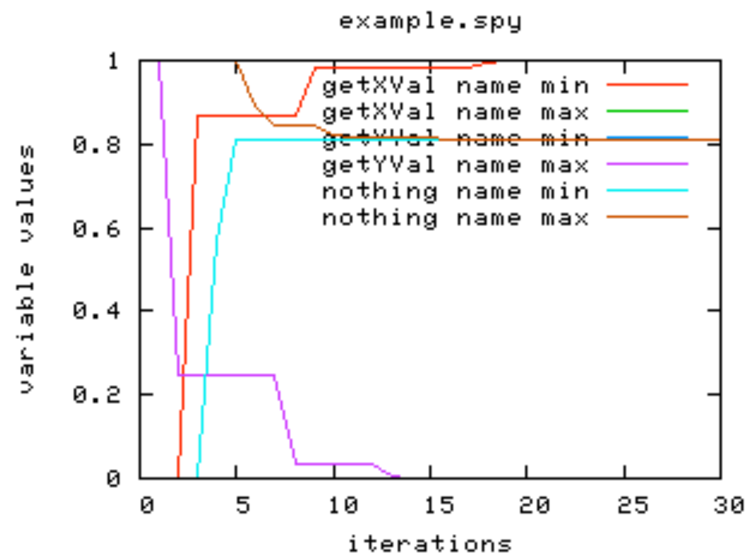
- This research was funded by NASA in order to find a better ways to evaluate procedural systems
  - Current methods, like model checkers, are limited by the state space explosion problem
  - Models used are very large
- Random sampling might prove useful

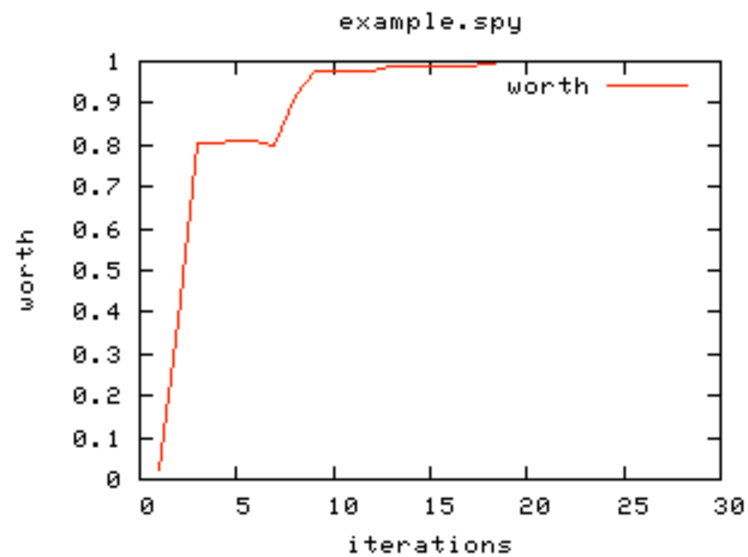
NASA

Rockwell Collins

Another Option

- Create set of conventions that allow procedural language to be:
  - Data Mined
  - Controlled
  - Altered
- This is SPY
- Current data mining techniques would not fit for SPY





Contributions and Goals

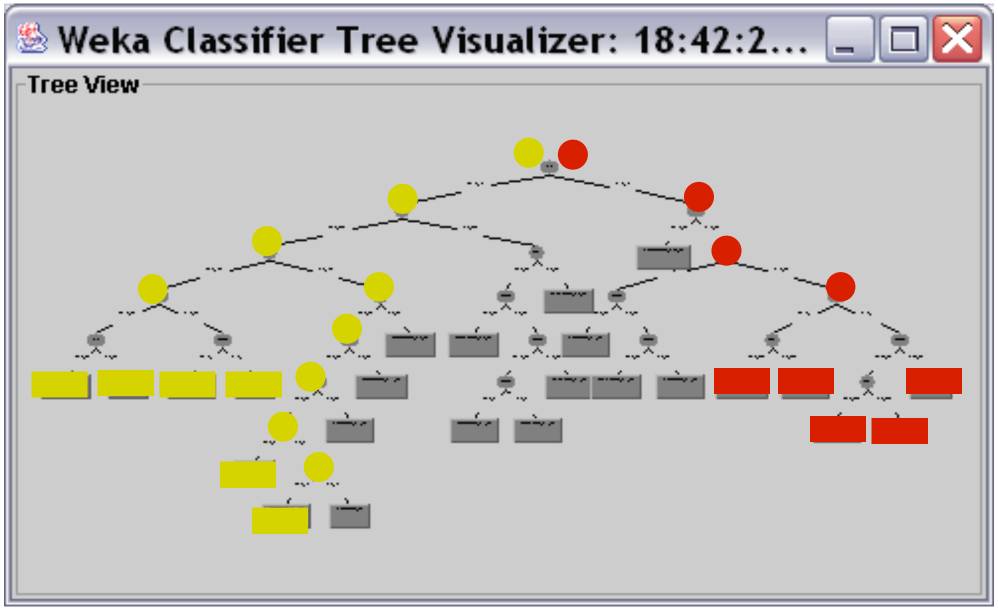
- To develop a set of conventions that allow procedural language to be data mined, controlled and altered.
- Result: a new treatment learner for this purpose that has a:
  - smaller memory footprint
  - Dramatically faster runtime

What is Treatment Learning?

- Preface
- What is Treatment Learning?
- How Can Treatment Learning be Improved?
- Tar4.0: Can Bayes Help Tar4?
- Tar4.1
- Experiments
- Future Work & Conclusion

The Explanation Problem

- Standard miners (e.g. even decision tree learners) can produce theories that are detailed yet incomprehensible to many readers.
- For Example, we are looking for good housing in Boston
- Minimum number of decisions that make the greatest difference in outcome

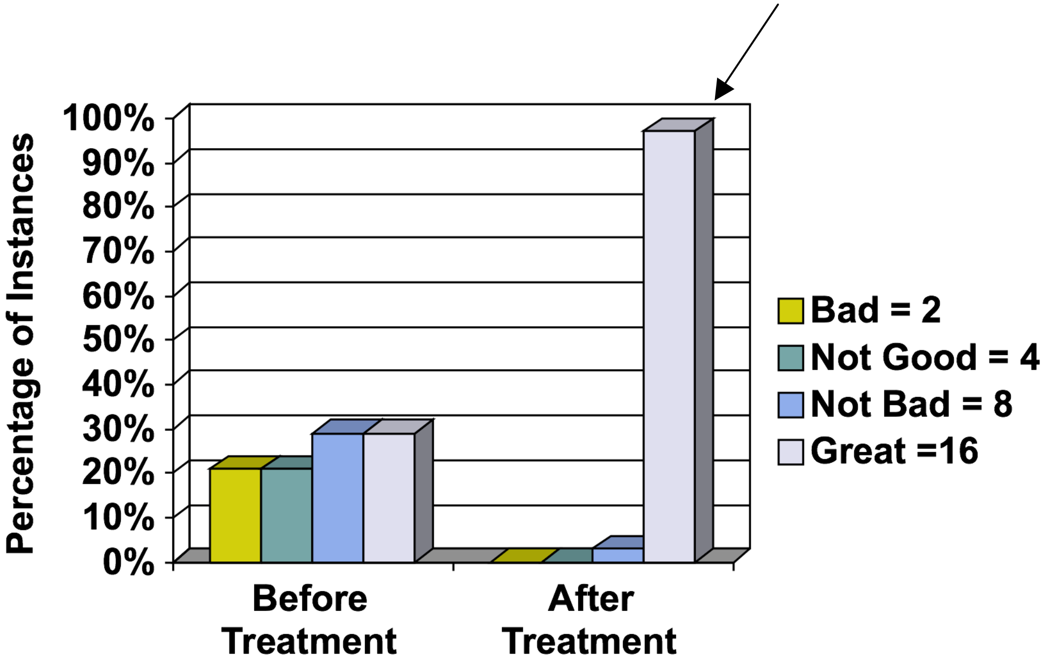


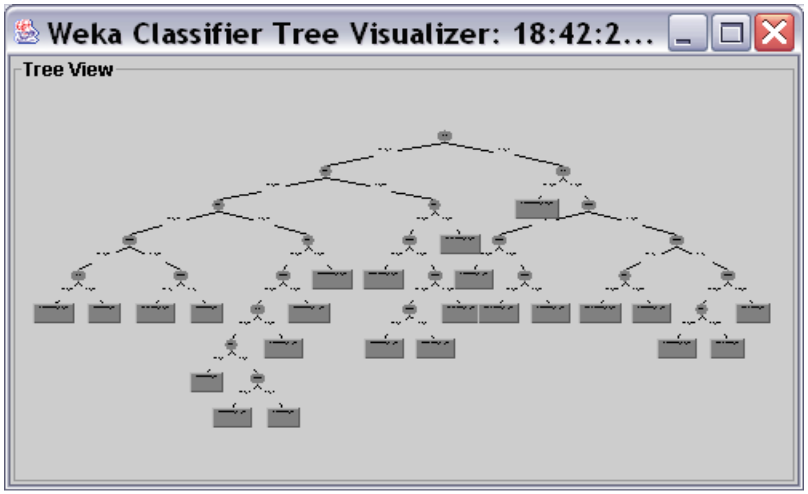
We want:

- Fewer details about the definition of each class.
- More about what actions..
  - avoid negative outcomes
  - and promote positive ones.
- More formally, Treatment Learning seeks:
  - a conjunction of attribute range-pairs
  - that identify a subpopulation in the larger population
    - with a high concentration of desired classes
    - a lower concentration of undesired classes
  - All based on a set of weighted classes
- Goal:
  - the mouse that frees the lion
  - I.e. the ***smallest*** treatment…
  - … provides the ***highest*** lift

Back to the example

- We are looking for good housing in Boston
- A treatment produced by a treatment learner is:
  - $(6.7 \le RM < 9.8) \wedge (12.6 \le PT < 15.9)$





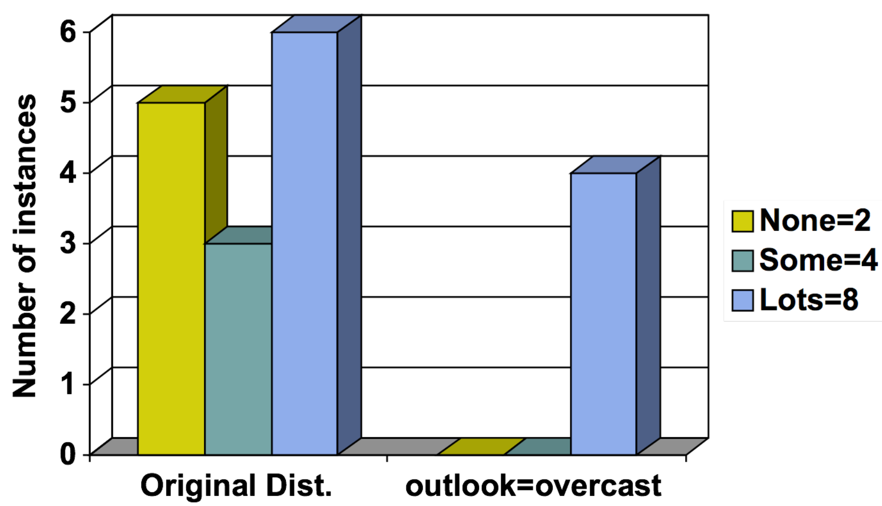
Four Concepts Define Treatment Learning

1. Lift (search bias)
2. Minimum Best Support (overfitting avoidance bias)
3. Small Treatment Effect (language bias)
4. Bias of weighted classes

1) Lift

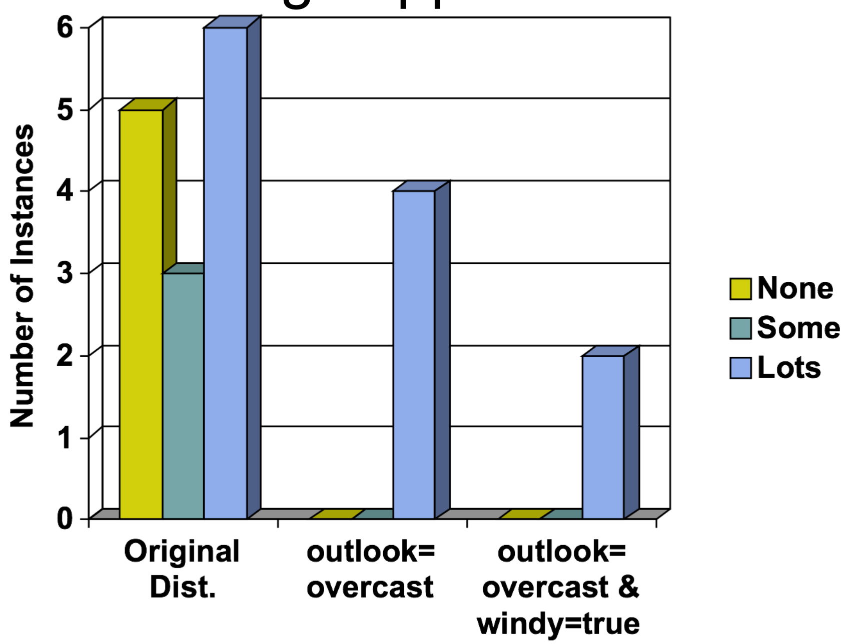
- Lift is the change in population ratio of the desired class over the undesired class compared to the original distribution
- Lift is a measure of effectiveness of a given

outlook	*temp(°F)*	*humidity*	*windy*	*class*	*outlook= overcast*
sunny	*85*	*86*	*false*	*none*	
sunny	*80*	*90*	*true*	*none*	
sunny	*72*	*95*	*false*	*none*	
rain	*65*	*70*	*true*	*none*	
rain	*71*	*96*	*true*	*none*	
rain	*70*	*96*	*false*	*some*	
rain	*68*	*80*	*false*	*some*	
rain	*75*	*80*	*false*	*some*	
sunny	*69*	*70*	*false*	*lots*	
sunny	*75*	*70*	*true*	*lots*	
overcast	*83*	*88*	*false*	*lots*	√
overcast	*64*	*65*	*true*	*lots*	√
overcast	*72*	*90*	*true*	*lots*	√
overcast	*81*	*75*	*false*	*lots*	√



2) Minimum Best Support

- A balance of purity and support for that treatment is desirable.
- An absolutely pure treatment with many attribute range pairs will not be useful if it is not well represented in the population.
- Lesson: Rules with strong support are better



3) Small Treatment Effect

- Empirically, most treatments very small.
  - four attribute-range pairs is often the max a treatment learner will produce.
- A side effect of minimum best support
- This is how treatment learners combat overfitting.

Where does Treatment Learning Fit into Data Mining

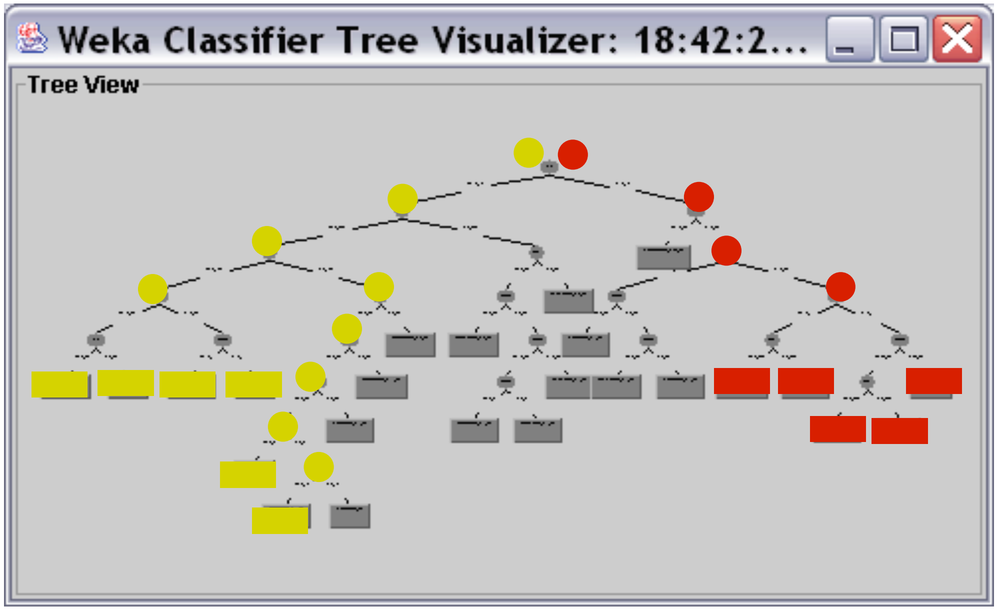
- Classification Learning
  - e.g.Decision Trees [Quinlan92] C4.5
- Association Rule Learning
  - e.g. Apriori [Zheng02]
- Contrast Set Learning
  - e.g STUCCO [Bay99]
  - Treatment Learners
    - Contrast set + minimal + weighted classes

How Can Treatment Learning be Improved?

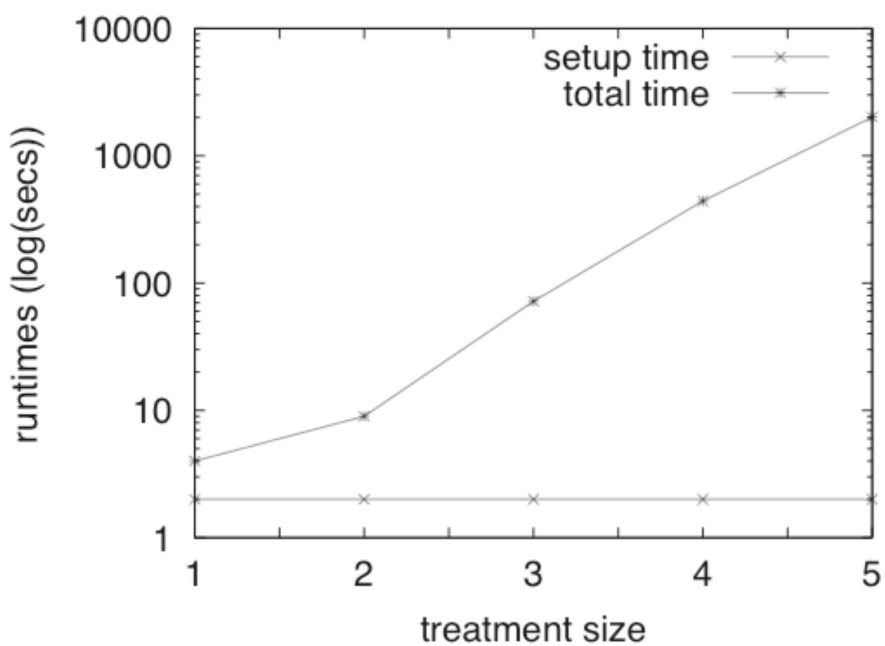
- Preface
- What is Treatment Learning?
- How Can Treatment Learning be Improved?
- Tar4.0: Can Bayes Help Tar4?
- Tar4.1
- Experiments
- Future Work & Conclusion

Tarzan

- A post-processor for for a decision tree
  - Traverse the tree looking for desired classes
  - Collapsing nodes that are unimportant
  - Minimum number of decisions that make the most difference in outcome



Tar2 [03tar2] Menzies and Hu 2003



- While useful in its test domain, it suffered from runtimes that grew exponentially with the size of the learned treatments
- Experiment of the process not the optimization

Tar3 [hu02] Menzies and Hu 2003

- stochastic search algorithm
- While the algorithm was incomplete, it was shown to produce almost identical treatments to Tar2's exhaustive enumeration of all possible treatments

```
function ONE(x = random(SIZE) )
   x timesDo
      treatment = treatment +  ANYTHING()
   return treatment

function ANYTHING()
   return a random range from CDF(lift1)

function SOME()
   REPEATS timesDo
      treatments = treatments +  ONE()
   sort treatments on lift
   return ENOUGH top items

function TAR3(lives = LIVES )
   for every range r do lift1[r]= lift(r)
   repeat
     before = size(temp)
     temp   = union(temp,  SOME())
     if (before==size(temp))
     then lives--
     else lives = LIVES
   until lives == 0
   sort temp on lift;
   return ENOUGH top items
```

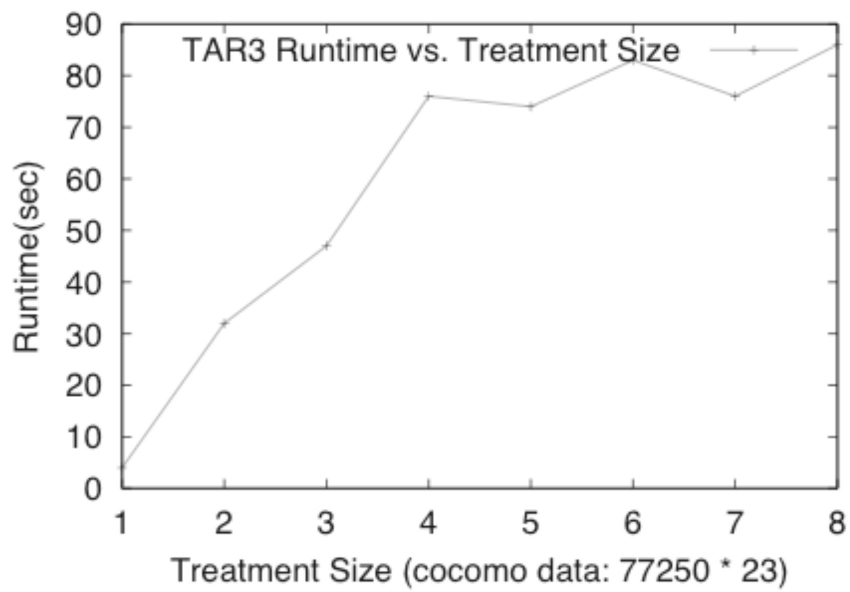
Tar3 is not a Data Miner

- According to [Bradley98] a data miner needs to:
  - Requires one scan, or less of the data
  - On-line, anytime algorithm
  - Suspend-able, stoppable, resume-able
  - Efficiently and incrementally add new data to existing models
  - Works within the available RAM

The Problem

Tar3 required multiple passes through the data in order to chronologically:

1. discretize the numerics;
2. collect statistics on the frequency of the discretize data;
3. test candidate treatments. (This step could require hundreds of passes through the data).



Tar4.0: Can Bayes Help Tar4?

- Preface
- What is Treatment Learning?
- How Can Treatment Learning be Improved?
- Tar4.0: Can Bayes Help Tar4?
- Tar4.1
- Experiments
- Future Work & Conclusion

How to Learn Treatments in a Single Pass of the Data

- This was initially accomplished by using concepts from a Bayes' Classifier
  - storing data in frequency tables
  - potential treatments were calculated using Bayes' Law
  - Various people have proposed that "**Bayes is enough**". (Domingos and Pazzani & Menzies and Orrego)
- Everything is stored in a two class system
  - If the dataset is continuous or contains more than two discrete classes then it is transferred to a two class system like so…

Two Class System

- There are two classes “apex” and “base”
  - Where apex is the most desired and base is the least desired.
- If a discrete class dataset is encountered with say 10 different classes and an instance that the third most desirable is encountered
  - The apex frequency counter for that instance would be 7/10 and 3/10 for the base
- If a continuous class is encountered and the max and min values are known
  - The apex frequency counter for a particular instance is (instance_value-min)/(max-min)
  - The base frequency counter for a particular instance would be 1-apex

Tar4.0

- The first attempt at a Bayesian treatment learner was find the *smallest* treatment T that *maximizes*:

$$\frac{L(apex \mid E)}{L(apex \mid E) + L(base \mid E)}$$

- didn't work: vastly out-performed by Tar3
- Why?
    - The infamous independence assumption.
- So is Bayes really enough?
    - Yes, but needs "support-based pruning"

```
function ONE (x = random(SIZE) )
   x timesDo
      treatment = treatment + ANYTHING()
   return treatment

function ANYTHING ( )
   return a random range from CDF(lift1)

function SOME ( )
   REPEATS timesDo
      treatments = treatments + ONE()
   sort treatments on lift
   return ENOUGH top items

function TAR3(lives = LIVES )
   for every range r do lift1[r]= lift(r)
   repeat
     before = size(temp)
     temp   = union(temp, SOME())
     if (before==size(temp))
     then lives--
     else lives = LIVES
   until lives == 0
   sort temp on lift;
   return ENOUGH top items
```

So what is the problem?

	E_1	E_2	E_3
$H = car$	*job*	*suburb*	*wealthy?*
ford	*tailor*	*NW*	*y*
ford	*tailor*	*SE*	*n*
ford	*tinker*	*SE*	*n*
bmw	*tinker*	*NW*	*y*
bmw	*tinker*	*NW*	*y*
bmw	*tailor*	*NW*	*y*

$$\overbrace{P(H|E)}^{future=} = \overbrace{\left(\prod_i P(E_i|H)\right)}^{now*} * \frac{\overbrace{P(H)}^{past}}{P(E)}$$

$P(H)$	$P(E_i\|H)$ job	suburb	wealthy?
ford:3=0.5	tinker:1=0.33 tailor:2=0.67	NW:1=0.33 SE:2=0.67	y:1=0.33 n:2=0.67
bmw:3=0.5	tinker:2=0.67 tailor:1=0.33	NW:3=1.00 SE:0=0.00	y:3=1.00 n:0=0.00

$$E = (job = tailor) \,\&\, (suburb = NW) \boxed{\,\&\, (wealthy = y)}$$

$$L(bmw \mid E) = \prod_i P(E \mid bmw) * P(bmw) = 0.33 * 1.00 * \boxed{1.00} * 0.5 = 0.16500$$

$$L(ford \mid E) = \prod_i P(E \mid ford) * P(ford) = 0.67 * 0.33 * \boxed{0.33} * 0.5 = 0.0364815$$

$$\Pr(bmw \mid E) = \frac{L(bmw \mid E)}{L(bmw \mid E) + L(ford \mid E)} = 81.9\%$$

Was 59.9%

$$\Pr(ford \mid E) = \frac{L(ford \mid E)}{L(bmw \mid E) + L(ford \mid E)} = 18.1\%$$

Was 40.1%

The Dependency Problem

- Works for Naïve Bayes.
  - The probability is inaccurate
  - But it doesn't matter because it just picks the largest of the classes
  - Domingos and Pazzani [1997]
- Destroyed Tar4.0
  - Tar4.0 doesn't just rank them
  - We need to use the probability calculation

Tar4.1

- Preface
- What is Treatment Learning?
- How Can Treatment Learning be Improved?
- Tar4.0: Can Bayes Help Tar4?
- Tar4.1
- Experiments
- Future Work & Conclusion

So what to do… Tar4.1

- Add support based pruning

$$0 \le likelihood \le 1$$

$$L(apex \mid E) = \Pr(E \mid apex) * \Pr(apex)$$

$$probability * likelihood = L(apex \mid E) \frac{L(apex \mid E)}{L(apex \mid E) + L(base \mid E)} = \frac{L(apex \mid E)^2}{L(apex \mid E) + L(base \mid E)}$$

- Intuition
  - By penalizing the treatment as its size grows there are less possibilities for dependencies.
  - Rich paths from our experience states not weak paths.

Evaluation Without Support Based Pruning - Tar4.0

- Without support based pruning the evaluation function would look like this:

$$a = L(apex \mid E)$$

$$b = L(base \mid E)$$

$$E = E_1 E_2 E_3 \ldots E_m$$

$$E' = E_1 E_2 \ldots E_{n-1} E_{n+1} \ldots E_m$$

$$a/x = L(apex \mid E')$$

$$b/y = L(base \mid E')$$

E_n is removed from the evidence.

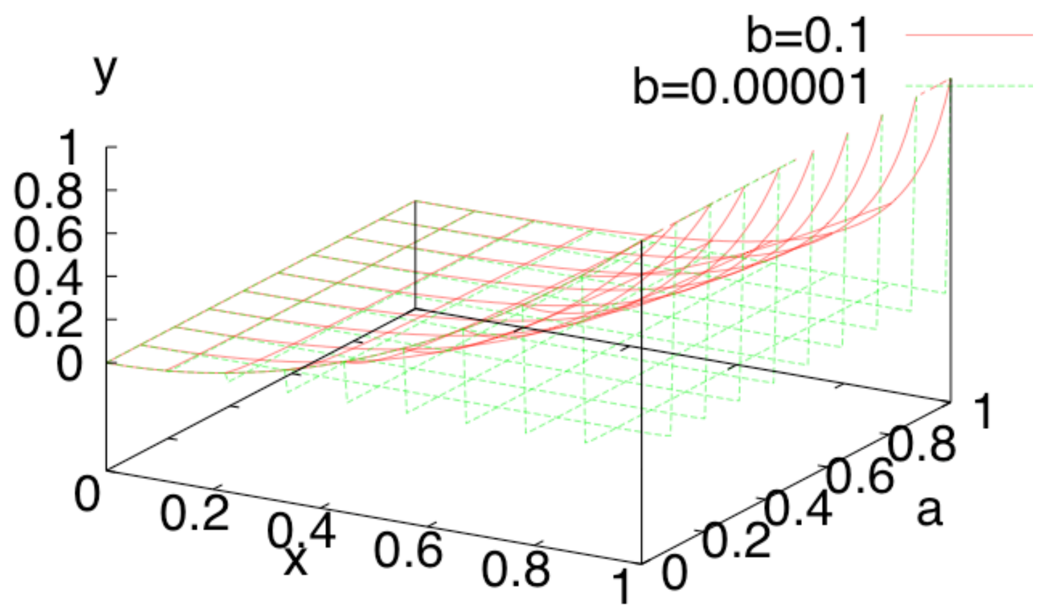
$$\left(\frac{(a/x)}{a/x + b/y} > \frac{a}{a+b} \right)$$

Tar4.0 would not be confused when the left term is greater than the right.

Evaluation With Support Based Pruning Tar4.1

- With support based pruning the evaluation function would look like this:

$$\left(\frac{(a/x)^2}{a/x+b/y} > \frac{a^2}{a+b} \right)$$

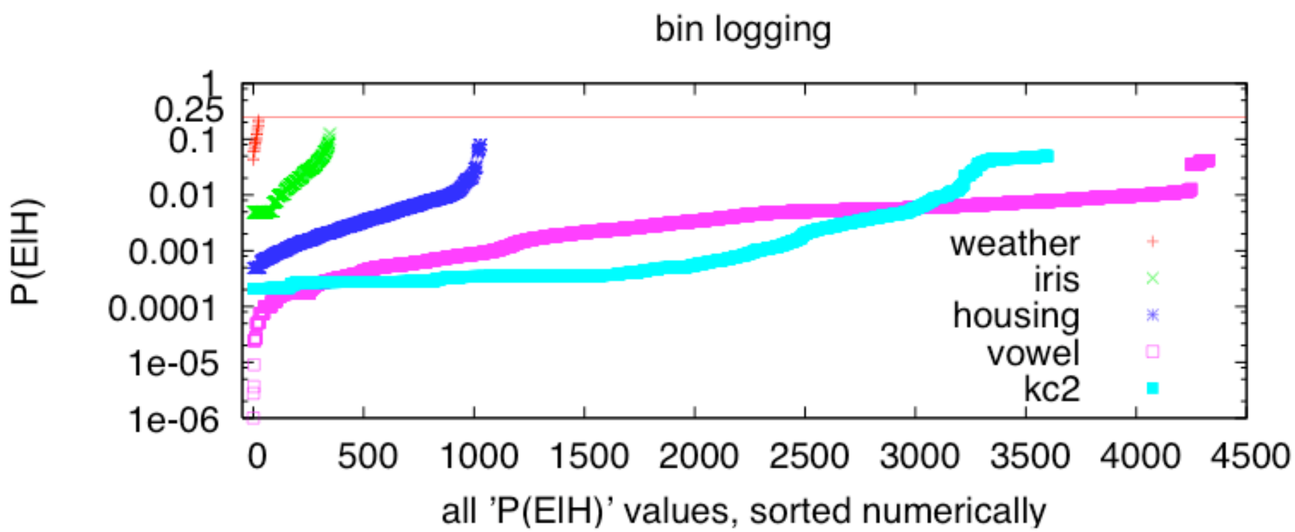


- Tar4.1 would not be confused when the left term is greater than the right.

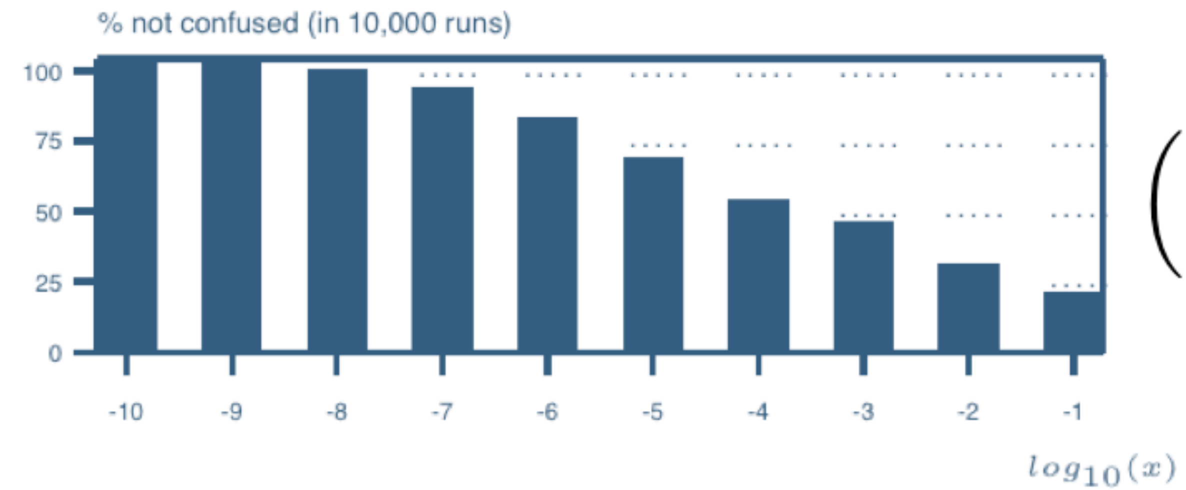
Using a Simulation

- It was run 10,000 with the following restrictions

$0 < i \leq 20$	*; treatment size*
$b < a$	; apex *is better than* base
$min < x \leq y \leq max$	*; see graphs*
$0 < a \leq x^i \leq x \leq 0.25$	; a *combines many* x*-like numbers*
$0 < b \leq y^i \leq y \leq 0.25$	; b *combines many* y*-like numbers*



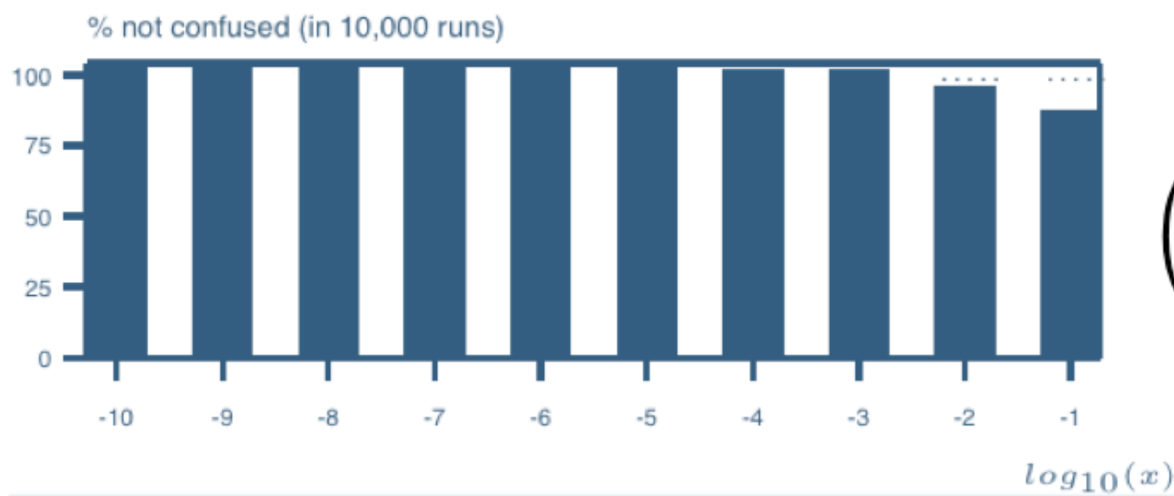
Results from Simulation



Tar4.0

$$\left(\frac{(a/x)}{a/x+b/y} > \frac{a}{a+b}\right)$$

Often confused.



Tar4.1

$$\left(\frac{(a/x)^2}{a/x+b/y} > \frac{a^2}{a+b}\right)$$

Rarely confused.

Experiments

- Preface
- What is Treatment Learning?
- How Can Treatment Learning be Improved?
- Tar4.0: Can Bayes Help Tar4?
- Tar4.1
- Experiments
- Future Work & Conclusion

Experiments

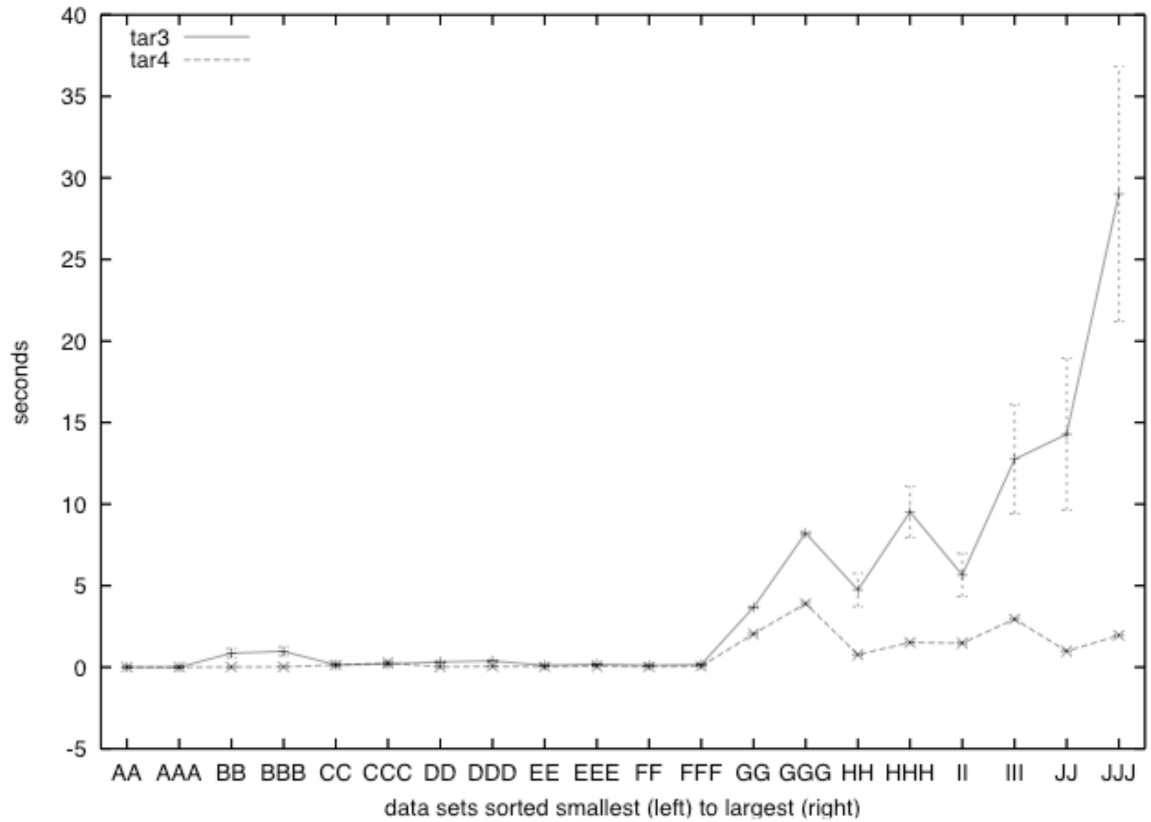
- Using the following data sets:

Dataset	Name	Number of Attributes	Number of Instances	Number of Classes
A	Contacts	4	24	3
B	Hepatitis	19	80	2
C	Sonar	60	208	2
D	Vote	16	232	2
E	Wisconsin Breast Cancer	9	699	2
F	Diabetes	8	768	2
G	Splice	60	3190	3
H	Kr-vs-Kp	36	3196	2
I	Waveform	40	5000	3
J	Mushroom	20	8124	2

Experiments for effectiveness, speed, and memory foot print were conducted.

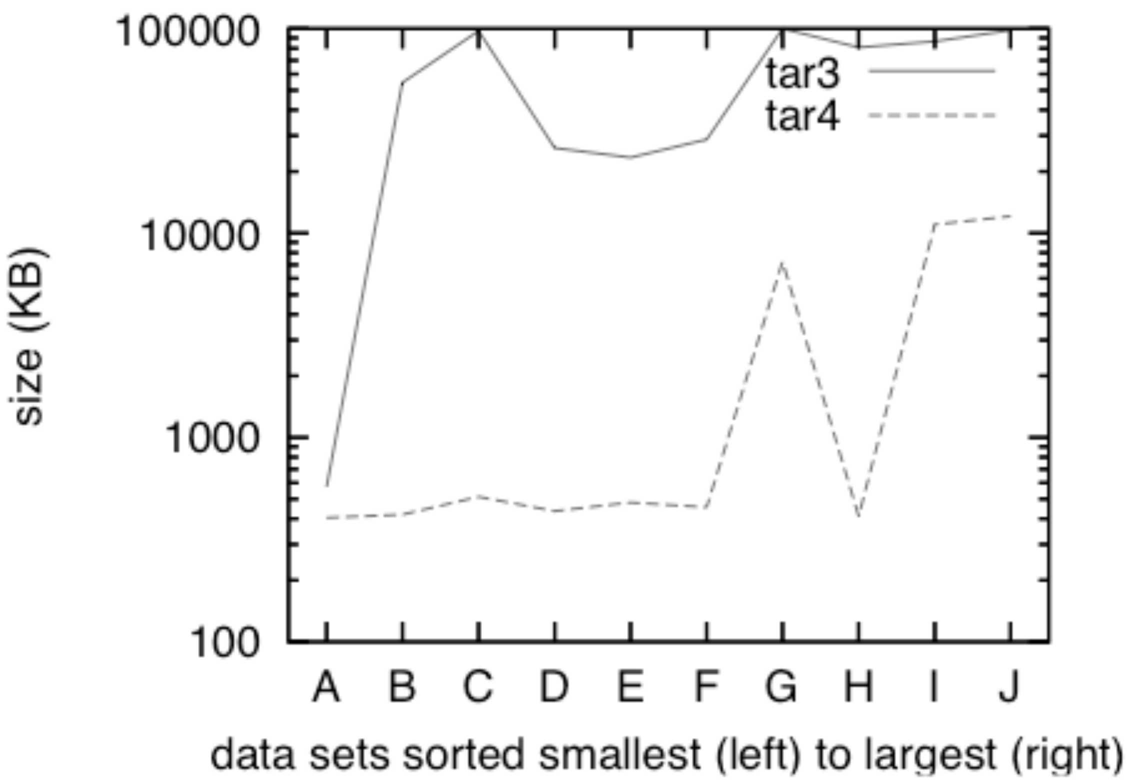
Runtime (Tar3 VS Tar4.1)

- Tar4.1 runs faster than Tar3, especially in large datasets
- Tar4.1 has far less variance in performance

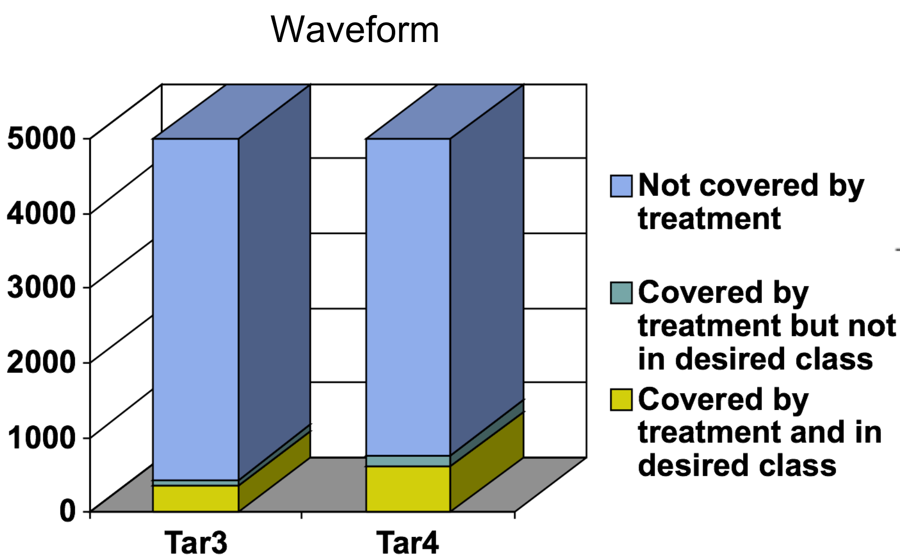


Memory Footprint (Tar3 VS Tar4)

- Low memory requirements. The memory footprint left by Tar4 is dramatically smaller than Tar3: often over 100 times smaller.



Effectiveness (Tar3 VS Tar4)



Dataset	Support for Tar3's best treatment	Support for Tar4's best treatment	Tar3's Percentage of support in desired class	Tar4's Percentage of support in desired class
Vote	108	113	95%	95%
Splice	442	442	95%	95%
Breast Cancer	69	69	100%	100%
Mushroom	2720	2160	95%	100%
Kr-vs-Kp	743	891	100%	76%
Waveform	435	770	83%	79%
Diabetes	123	46	66%	85%
Sonar	28	22	89%	91%
Hepatitis	42	47	100%	98%
Contacts	12	12	100%	100%

- $\frac{3}{10}$ Tar4 chose the same treatment
- $\frac{7}{10}$ Tar4's treatment had at least the same support
- $\frac{7}{10}$ Tar4's treatment had at least the same percentage in the desired class
- $\frac{1}{10}$ Tar4's treatment had the same percentage as Tar3 but with greater support.

Future Work & Conclusion

- Preface
- What is Treatment Learning?
- How Can Treatment Learning be Improved?
- Tar4.0: Can Bayes Help Tar4?
- Tar4.1
- Experiments
- Future Work & Conclusion

Future Work

- Run on more Rockwell-Collins Models
- Add a windowing policy
- Try Tar4 with incremental learning
  - The first step of this has been completed by adding the SPADE [orrego05]
- Infinite stream of data
- Eventually have numeric overflow.

Conclusion

- Treatment learning:
  - very useful for creating small, easy to explain, theories.
- Runtime monitors for large systems
  - must handle large data sets
- We need scalable learners:
  - Tar3 wont scale.
- Tar4.1 (Bayesian Treatment Learning + Support based pruning) does scale
  - The costs are low:
    - Low guesstimate errors
  - The benefits are high:
    - Fast runtimes
    - Low memory requirements

Questions or Comments?