Next Generation “Treatment Learning”
(finding the diamonds in the dust)

Tim Menzies
UpLift Systems

timm@bart-massey.com
The strangest thing...

“In any field, find the strangest thing, and explore it” – John Wheeler

Q: How have dummies (like me) managed to gain (some) control over a (seemingly) complex world?

A: The world is simpler than we think.

− Models contain clumps
− A few collar variables decide which clumps to use.

TAR2, TAR3, TAR4:

− Data miners that assume clumps/collars
− Reports effects never seen before
− Finds solutions faster than other methods
− Returns tiniest theories
− Scales to infinite data streams (⇐ new result)
How Complex are our Models?

**COLLARS**
A small number few variables controls the rest:
- **DeKleer [1986]**: “Minimal environments” in the ATMS;
- **Menzies and Singh [2003]**: “Tiny minimal environments”;
- **Crawford and Baker [1994]**: “Master variables” in scheduling;
- **Williams et al. [2003]**: ‘Backdoors” in satisfiability.

**CLUMPS**
- **Druzdzel [1994]**. Commonly, a few states; very rarely, most states;
- **Petránek [2004]**. “Straight jackets” in formal models: state spaces usually sparse, small diameter, many diamonds.
Exploiting Simplicity

- If clumps
  - most of the action in a small number of states
  - effective search space = small

- If collars:
  - A few variables that switch you between states

- Treatment learning
  - If a few variables control the rest, then..
    - All paths \( \text{inputs} \rightarrow \text{outputs} \) use the collars (by definition).
  - So don’t search for the collars:
    - They’ll find you.
    - Just sample, and count frequencies \( F \).
  - Divide output \( \text{good} \) and \( \text{bad} \)
    - Focus on ranges \( R_i \) with large \( \frac{F(R_i | \text{good})}{F(R_i | \text{bad})} \)

- Great way to learn tiny theories.
Learns Smaller Theories

<table>
<thead>
<tr>
<th>find graphics on a page from 11 features</th>
<th>find good housing in Boston</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
</tbody>
</table>

34 \leq \text{height} < 86 \land \\
3.9 \leq \text{mean}_tr < 9.5

6.7 \leq RM < 9.8 \land \\
12.6 \leq PTRATION < 15.9
Why Learn Small Theories?

Reduce Uncertainty:
Linear regression: $\sigma^2 \propto |variables|$ (Miller [2002]);

“Pluralitas non est ponenda sine necessitate”:
MDL (Wallace and Boulton [1968]); FSS (Hall and Holmes [2003])

Explanation:
Smaller theories are easier to explain (or audit).

Performance:
The simpler the target concept, the faster the learning.

Construction cost:
Need fewer sensors and actuators.

Operations cost:
Less to do: important for manual procedures;
Less to watch: important for data-intensive tasks like security monitoring.

Pruning is good modeling:
Real world data often has noisy, irrelevant, redundant variables.
So What is Treatment Learning?

34 \leq \text{height} < 86 \land 3.9 \leq \text{mean}_{tr} < 9.5

- **E**: training data with examples of $R_i \rightarrow C$
  - $R_i$: attribute ranges
  - $C$: classes with utilities \( \{ U_1 < U_2 < \ldots < U_C \} \)
  - $F_1\%, F_2\%, \ldots, F_C\%$: frequencies of $C$ in $E$

- **T** treatment of size $X$: \( \{ R_1 \land R_2 \ldots \land R_X \} \);
  - $T \cap E \rightarrow e \subseteq E$ with frequencies $f_1\%, f_2\%, \ldots f_C\%$
  - seek smallest $T$ with largest \( \text{lift} = \frac{\sum_C U_C f_C}{\sum_C U_C F_C} \)

- This talk:
  - Implementation, examples, a new scale-up method
In practice...
The TAR3 Treatment Learner

- Assume clumps and collars
  - Just thrash around some.

- Build treatments
  \[ \{ R_1 \land R_2 \ldots \land R_X \} \] of size \( X \)
  - FIRST try \( X = 1 \)
  - THEN use the \( X = 1 \) results to guide the \( X > 1 \) search.

  - Discretization: equal frequency binning

- Empirically:
  - Run times linear on treatment SIZE, number of examples
  - Works as well as TAR2's complete search

---

function ONE \((x = \text{random}(\text{SIZE})\) )
\[
\begin{align*}
x \times \text{timesDo} \\
\text{treatment} = \text{treatment} + \ \text{ANYTHING()} \\
\text{return} \ \text{treatment}
\end{align*}
\]

function ANYTHING ()
\[
\text{return a random range from CDF(lift1)}
\]

function SOME ()
\[
\begin{align*}
\text{REPEATS timesDo} \\
\text{treatments} = \text{treatments} + \ \text{ONE()} \\
\text{sort treatments on lift} \\
\text{return ENOUGH top items}
\end{align*}
\]

function TAR3 \((lives = \text{LIVES})\)
\[
\begin{align*}
\text{for every range } r \text{ do lift1}[r] &= \text{lift}(r) \\
\text{repeat} \\
\text{before} = \text{size(temp)} \\
\text{temp} = \text{union(temp, SOME())} \\
\text{if} \ (\text{before}==\text{size(temp)}) \text{ then lives--} \\
\text{else lives} = \text{LIVES} \\
\text{until lives == 0} \\
\text{sort temp on lift;} \\
\text{return ENOUGH top items}
\end{align*}
\]

Useful defaults: \(<\text{SIZE}=10, \text{REPEATS}=100, \text{ENOUGH}=20, \text{LIVES}=5>\)
"Limits to Growth" :: [Meadows et al. 1972]
A second look at "Limits to Growth": [Geletko and Menzies 2003]
Vensim's World-3 (1991): 295 variables

Happily ever after if
- family size ≤ 2, menstruation onset > 18, industrial capital output = [3..5).
- This happy ending is not mentioned in [Meadows et al. 1972].
**Compared with More Complete Search**

- DDP requirements models from deep-space missions (from JPL).
- Iterative learning: \( simulation_i \rightarrow learn \rightarrow constrain \rightarrow simulation_{i+1} \)

\[
SA = \frac{\text{benefit}}{\text{maxBenefit}} + \left( 1 - \frac{\text{cost}}{\text{maxCost}} \right) \left( 2 \times \frac{\text{number of selected mitigations}}{2} + 1 \right)
\]

![Graph showing benefits vs. cost](image)

**TAR3:** 7*300 samples  
**SA:** 9*3000 samples
Learns Very Tiny Theories

- Compare with feature subset selection: [Hall and Holmes 2003]
- For each class $c \in C$
  - Give $c$ the largest utility $U_c$.
  - Find treatments for $c$
- $Selected = \text{all attributes in treatments for all } c \in C$.
- $Accuracy = selected's$ performance in some target learner.
- [Menzies et al. 2005]

### Table: #attributes selected (target learner = C4.5)

<table>
<thead>
<tr>
<th>domain</th>
<th>% of attributes ignored</th>
<th>accuracy improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anneal</td>
<td>81.6%</td>
<td>2.66%</td>
</tr>
<tr>
<td>credit-g</td>
<td>75.0%</td>
<td>2.17%</td>
</tr>
<tr>
<td>Soybean</td>
<td>54.3%</td>
<td>0.65%</td>
</tr>
<tr>
<td>vote</td>
<td>62.5%</td>
<td>0.21%</td>
</tr>
<tr>
<td>breast-c</td>
<td>77.8%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Segment</td>
<td>78.9%</td>
<td>-0.51%</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>94.1%</td>
<td>-0.90%</td>
</tr>
<tr>
<td>Diabetes</td>
<td>87.5%</td>
<td>-2.28%</td>
</tr>
<tr>
<td>lymph</td>
<td>83.3%</td>
<td>-2.75%</td>
</tr>
<tr>
<td>HorseColic</td>
<td>90.9%</td>
<td>-4.45%</td>
</tr>
<tr>
<td>average</td>
<td>78.6%</td>
<td>-1.13%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>domain</th>
<th>original</th>
<th>ig</th>
<th>cfs</th>
<th>cbs</th>
<th>rlf</th>
<th>wrp</th>
<th>pc</th>
<th>select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soybean</td>
<td>35</td>
<td>19</td>
<td>24</td>
<td>35</td>
<td>32</td>
<td>19</td>
<td>30</td>
<td>▶16</td>
</tr>
<tr>
<td>Anneal</td>
<td>38</td>
<td>17</td>
<td>21</td>
<td>15</td>
<td>20</td>
<td>18</td>
<td>36</td>
<td>▶7</td>
</tr>
<tr>
<td>vote</td>
<td>16</td>
<td>12</td>
<td>10</td>
<td>▶6</td>
<td>11</td>
<td>9</td>
<td>11</td>
<td>▶6</td>
</tr>
<tr>
<td>credit-g</td>
<td>20</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>▶4</td>
<td>5</td>
</tr>
<tr>
<td>Segment</td>
<td>19</td>
<td>16</td>
<td>12</td>
<td>9</td>
<td>13</td>
<td>9</td>
<td>16</td>
<td>▶4</td>
</tr>
<tr>
<td>lymph</td>
<td>18</td>
<td>6.8</td>
<td>5.3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>▶3</td>
</tr>
<tr>
<td>breast-c</td>
<td>9</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>7</td>
<td>4</td>
<td>4</td>
<td>▶2</td>
</tr>
<tr>
<td>Horse colic</td>
<td>22</td>
<td>4</td>
<td>4</td>
<td>▶2</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>▶2</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>34</td>
<td>12</td>
<td>7</td>
<td>9</td>
<td>9</td>
<td>7</td>
<td>10</td>
<td>▶2</td>
</tr>
<tr>
<td>Diabetes</td>
<td>8</td>
<td>33</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>▶1</td>
</tr>
</tbody>
</table>
Scaling Up

- TAR3 is not a Data Miner
- SAWTOOTH
- NaïveBayes classifiers
- CUBE & TAR4
- Why did TAR4.0 fail?
- TAR4.1
- Pre-condition
- Typical values
- TAR4.1 Works
- So What?
- But Why Big Treatments?

Related Work

And so...

Questions? Comments?
TAR3 is not a Data Miner

The data mining desiderata: Bradley et al. [1998]:

- Requires one scan, or less of the data
- On-line, anytime algorithm
- Suspend-able, stoppable, resumable
- Efficiently and incrementally add new data to existing models
- Works within the available RAM

TAR3 is not a data miner

- Stores all examples in RAM
- Requires at three scans
  1. discretization
  2. collect statistics, build treatments
  3. rank generated theories
SAWTOOTH is a data miner

SAWTOOTH = incremental NaïveBayes classifier [Menzies and Orrego, 2005]

- Exploits the “saturation effect”:
  - Learners performance improves and plateaus, after 100s of examples
  - Processes data in chunks (window = 250)
  - Disables learning while performance stable

- One-pass through the data
  - Incremental discretization of numeric data (SPADE)
  - Input each example, converted to frequency counts, then deletes

- Results
  - Small memory; scales.
  - Recognizes and reacts to concept drift

- Can we model treatment learning as a NaïveBayes classifier?
**NaïveBayes classifiers**

**NaïveBayes classifiers**

- $\text{Evidence } E$, hypothesis $H$

$$P(H|E) = \left( \prod_i P(E_i|H) \right) \times \frac{P(H)}{P(E)}$$

<table>
<thead>
<tr>
<th>$H = \text{car}$</th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ford</td>
<td>job</td>
<td>suburb</td>
<td>wealthy?</td>
</tr>
<tr>
<td>ford</td>
<td>tailor</td>
<td>NW</td>
<td>y</td>
</tr>
<tr>
<td>ford</td>
<td>tailor</td>
<td>SE</td>
<td>n</td>
</tr>
<tr>
<td>bmw</td>
<td>tinker</td>
<td>NW</td>
<td>y</td>
</tr>
<tr>
<td>bmw</td>
<td>tinker</td>
<td>NW</td>
<td>y</td>
</tr>
<tr>
<td>bmw</td>
<td>tinker</td>
<td>NW</td>
<td>y</td>
</tr>
</tbody>
</table>

- $E = \text{job}=\text{tailor} & \text{suburb}=\text{NW}$
- **Likelihood** $L(\text{bmw}|E) = \prod_i P(E|\text{bmw}) \times P(\text{bmw}) = 0.33 \times 1.00 \times 0.5 = 0.16500$
- $L(\text{ford}|E) = \prod_i P(E|\text{ford}) \times P(\text{ford}) = 0.67 \times 0.33 \times 0.5 = 0.11055$
- $$\text{Prob}(\text{bmw}|E) = \frac{L(\text{bmw}|E)}{L(\text{bmw}|E)+L(\text{ford}|E)} = 59.9\%$$
- $$\text{Prob}(\text{ford}|E) = \frac{L(\text{ford}|E)}{L(\text{bmw}|E)+L(\text{ford}|E)} = 40.1\%$$
- So our tailor drives a **bmw**
- Naïve: assumes independence; counts single attribute ranges (not combinations)
  - But optimal under the one-zero assumption \cite{Domingos and Pazzani [1997]}.  
  - Incremental simple, fast learning/classification speed, low storage space.
Introduction

In practice...

Scaling Up

- TAR3 is not a Data Miner
- SAWTOOTH
- Naïve Bayes classifiers
- CUBE & TAR4
- Why did TAR4.0 fail?
- TAR4.1
- Pre-condition
- Typical values
- TAR4.1 Works
- So What?
- But Why Big Treatments?

Related Work

And so...

Questions? Comments?

---

**CUBE & TAR4**

<table>
<thead>
<tr>
<th>outlook</th>
<th>$U_1$: minimize temperature</th>
<th>humidity</th>
<th>windy</th>
<th>$U_2$: maximize play</th>
<th>$up_i$</th>
<th>$down_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>overcast</td>
<td>64</td>
<td>65</td>
<td>TRUE</td>
<td>yes=1</td>
<td>1.00</td>
<td>0</td>
</tr>
<tr>
<td>rainy</td>
<td>68</td>
<td>80</td>
<td>FALSE</td>
<td>yes=1</td>
<td>0.87</td>
<td>0.13</td>
</tr>
<tr>
<td>sunny</td>
<td>80</td>
<td>90</td>
<td>TRUE</td>
<td>no=0</td>
<td>0.11</td>
<td>0.89</td>
</tr>
</tbody>
</table>

- Examples are placed in a $U$-dimensional hypercube (one dimension for each utility):
  - apex = best = $\{1,1,1,1,...\}$;
  - base = worst = $\{0,0,0,0,...\}$

- $example_i$ has distance $0 \leq D_i \leq 1$ from apex (normalized by $U^{0.5}$)

- Each range $R_j \in example_i$ adds $down_i = D_i$ and $up_i = 1 - D_i$ to $F(R_j|base)$ and $F(R_j|apex)$.

  
  \[
  P(apex) = \frac{\sum_i up_i}{\sum_i up_i + \sum_i down_i} \\
  P(base) = \frac{\sum_i down_i}{\sum_i up_i + \sum_i down_i} \\
  P(R_j|apex) = \frac{F(R_j|apex)}{\sum_i up_i} \\
  P(R_j|base) = \frac{F(R_j|base)}{\sum_i down_i} \\
  L(apex|R_k \land R_l \land ...) = \prod_x P(R_x|apex) \ast P(apex) \\
  L(base|R_k \land R_l \land ...) = \prod_x P(R_x|base) \ast P(base)
  \]

**TAR4.0**: Bayesian treatment learner = find the smallest treatment $T$ that maximizes:

\[
P(apex|T) = \frac{L(apex|T)}{L(apex|T) + L(base|T)}
\]

; didn’t work: out-performed by TAR3
Why did TAR4.0 fail?

- Hypothesis: muddled-up by dependent attributes;
- “Naïve” Bayes: assume independence, keeps singleton counts.

<table>
<thead>
<tr>
<th>$H = \text{car}$</th>
<th>$E_1$</th>
<th>$E_2$</th>
<th>$E_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ford</td>
<td>tailor</td>
<td>NW</td>
<td>y</td>
</tr>
<tr>
<td>ford</td>
<td>tailor</td>
<td>SE</td>
<td>n</td>
</tr>
<tr>
<td>ford</td>
<td>tinker</td>
<td>SE</td>
<td>n</td>
</tr>
<tr>
<td>bmw</td>
<td>tinker</td>
<td>NW</td>
<td>y</td>
</tr>
<tr>
<td>bmw</td>
<td>tinker</td>
<td>NW</td>
<td>y</td>
</tr>
<tr>
<td>bmw</td>
<td>tailor</td>
<td>NW</td>
<td>y</td>
</tr>
</tbody>
</table>

| $E$                  | $P(\text{bmw}|E)$ | $P(\text{ford}|E)$ |
|----------------------|-------------------|-------------------|
| $\text{job = tailor}$ & $\text{suburb = NW}$ | 59.9%             | 40.1%             |
| $\text{job = tailor}$ & $\text{suburb = NW}$ & $\text{wealthy = y}$ | 81%               | 19.0%             |

- Adding redundant information radically changes probabilities? Bad!
- Note: gets class probabilities WRONG, but RANKS classes correctly
  - Domingos and Pazzani [1997]
- We asked TAR4.0 to do what you must never do:
  - compare numeric of probabilities of the same class in NaïveBayes.
Introduction

In practice...

Scaling Up

- TAR3 is not a Data Miner
- SAWTOOTH
- NaïveBayes classifiers
- CUBE & TAR4
- Why did TAR4.0 fail?

- Pre-condition
- Typical values
- TAR4.1 Works
- So What?
- But Why Big Treatments?

Related Work

And so...

Questions? Comments?

TAR4.1

- Prune treatments with low support in the data.
- What does “support” mean?
  - Maximal when includes all examples from a class
  - $0 \leq \text{support} \leq 1$
  - $\text{support} = \text{likelihood} = \prod_x P(R_x | H) \ast P(H)$

- $\text{probability} \ast \text{support} = \frac{L(\text{apex} | E)^2}{L(\text{apex} | E) + L(\text{base} | E)}$

- Worked!
  - Much faster, less memory than TAR3:
    - No need for a second scan
    - No need to hold examples in RAM
  - Bayesian guess-timate for support of best class
    - (almost) the same as TAR3
  - No connection treatment size to guess-timate error.

- But why did it work so well?
When Won't Dependencies Confuse TAR4?

- $T' = T + t$ where $t$ is an attribute dependent on members of $T$;
- TAR4.1 not confused by $t$ when it ignores treatments that use it.

$$
\begin{align*}
    a &= L(\text{apex}|T') = P(t|\text{apex}) \times \prod_i P(T_i|\text{apex}) \times P(\text{apex}) \\
    b &= L(\text{base}|T') = P(t|\text{base}) \times \prod_i P(T_i|\text{base}) \times P(\text{base})
\end{align*}
$$

- Then when is $\text{support} \times \text{probability}$ increased by ignoring $x$ and $y$?

$$
\begin{align*}
    \left( \frac{(a/x)^2}{a/x + b/y} \right) &> \left( \frac{a^2}{a + b} \right) \\
    \Rightarrow y &> \frac{bx^2}{b + a - xa}
\end{align*}
$$

- And for TAR4.0: s pre-condition for no confusion: \( \frac{(a/x)}{a/x+b/y} > \frac{a}{a+b} \)
Typical Values and Constraints:\n\[ \frac{(a/x)^2}{a/x + b/y} \]

- \( 0 < i \leq 20 \); treatment size
- \( b < a \); \textit{apex} is better than \textit{base}
- \( 10^{-10} < x \leq y \leq 0.25 \); 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

![Graph](image)
TAR4.1 Works

- Pick \(\{a,b,x,y,i\}\) at random within typical values; reject those violate our constraints;
- Check pre-conditions; report rounded \(\log_{10}\) values;
- TAR4.0: not confused when \(\left( \frac{(a/x)}{a/x+b/y} > \frac{a}{a+b} \right)\)

% not confused (in 10,000 runs)

- TAR4.1 Works

% not confused (in 10,000 runs)

TAR4.1: not confused when \(\left( \frac{(a/x)^2}{a/x+b/y} > \frac{a^2}{a+b} \right)\)

Often confused.

Rarely confused.
So What?

- Mathematically, TAR4.0 will always fail (except for $x \ll 1$);
- TAR4.1 succeeds since pre-condition is usually satisfied
  - In 96.52% of our simulations
- So, theoretically and empirically:
  - Bayesian treatment learning with CUBE can guess effect of treatments using frequency counts,
  - Does not need a second scan of the data (providing you use $support \times probability$)
  - Now we have a data miner TAR4.1.
- By the way,
  - No need for Bayes nets in this domain
  - Why doesn’t this mean that treatments will never grow beyond size=1?
But Why Big Treatments?

- When are larger treatments acceptable; i.e. \( \left( \frac{(a/x)^2}{a/x+b/y} < \frac{a^2}{a+b} \right) \)?
- When is \( y < \frac{bx^2}{b+a-xx} \).

When \( x \) is large and \( y \) is much smaller than \( x \)
- i.e. when some attribute ranges has a high frequency in the apex and a much lower frequency in the base.
- If collars then such ranges are not common; i.e. dependencies unlikely.
Related Work
References

[SAWTOOTH]


[Treatment learning]

- R. Clark. Faster treatment learning, 2005
References (2)

- **Phase transition**

- **Contrast set learners**
References (3)

- **Collars and clumps**

- **Data mining**
  - P. Domingos and G. Hulten. Mining high-speed data streams. In *Knowledge Discovery and Data Mining*, pages 71–80, 2000. URL [citeseer.ist.psu.edu/domingos00mining.html](http://citeseer.ist.psu.edu/domingos00mining.html)
Feature subset selection


Why Does NaïveBayes Work?


References (5)

- **Machine learning**

- **Misc**

- **MDL & MML**
And so...
Success Despite Complexity

- Maybe....
  - The world is not as complex as we think
  - Real world models clump, have collars.
  - Possible to quickly search, find ways to select for preferred states.

- Ultimately, this is an empirical study.
  - Q: When does a clumping/collaring-inspired search engine succeed?
  - A: Often
    - Reports effects never seen before (limits to growth)
    - Finds solutions faster than other methods (JPL).
    - Returns tiniest theories (fss)
    - Scales to infinite data streams (TAR4.1)

- Many applications. May I try this on your problems?
A Final Word

Sometimes the world is complex:
- 2% optimizing air-flow over leading wing in trans-sonic range
- synthesis of optimized code for complex engineering problems

And sometimes it ain’t.
- Try the simple solution before the more complex.
- Benchmark the complex against the seemingly less sophisticated.
- Warning: your straw man may not burn
Questions? Comments?