Chapter 6: The BACON and Related Programs

After five chapters of Computational Scientific Discovery theory we will look at our first set of true CSD algorithms in this chapter. The BACON programs are meant to rediscover simple scientific equations given experimental data, information on the experimental apparatus, and hunches about the equation's structure.

The BACON series of programs were influential in the history of CSD. It started interest in a sequence of ever more clever equation finding algorithms, and ultimately into scientific process and scientific structural knowledge.

6.1 BACON 1

BACON 1 was the first equation-finder created by Pat Langley, Herbert Simon, Gary Bradshaw and Jan Zytkow. It finds equations by comparing attributes of objects, and by creating new attributes until it invents one that does not vary.

BACON is given data on a set of objects of interest, planets for example. For each object it is given the numeric value of a set of attributes. For planets the user could specify $D$, the distance between the planet and the Sun, and $P$, the length of the planet's orbital period (its “year”).

BACON’s goal is to find an attribute with invariant values across objects but as both $D$ and $P$ vary it will create a new attribute. It has three operators for creating a new attribute, all of which compare two existing attributes.

1. If two existing attributes $A_1$ and $A_2$ increase linearly relative to each other it will create two new attributes $S$ and $I$ which are the slope and y-intercept of the line. These new attributes are constant across all objects. When this consistency is detected during the next round the equation-finding process will stop.

2. If two attributes $A_1$ and $A_2$ increase together, but not linearly then BACON will create a new attribute $A_r$, which is the ratio between $A_1$ and $A_2$.

3. If attribute $A_1$ increases while $A_2$ decreases then BACON will create a new attribute $A_p$, which is the product of $A_1$ and $A_2$.

For example, say BACON was given the following data:

<table>
<thead>
<tr>
<th>Planet</th>
<th>$D$</th>
<th>$P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>B</td>
<td>4.0</td>
<td>8.0</td>
</tr>
<tr>
<td>C</td>
<td>9.0</td>
<td>27.0</td>
</tr>
</tbody>
</table>

This is invented data, but it would work on the actual data if it were noise-free. Further, both $D$ and $P$ have been scaled so that their smallest value is 1.0. The scaling is just done for convenience: it allows us mere humans easier understand the values.

BACON would see that neither attribute is constant so it would try to create a new attribute. It would see that they increase together, but not linearly. Therefore it would create a new attribute that divides one by the other:
This new attribute is not constant either. It compares $D/P$ with $D$ and with $P$ and sees that it is linear with neither. The new attribute does, however, vary inversely with both $D$ and $P$, so it can propose multiplying it by either $D$ or $P$. Multiplying $D/P$ by $P$ just creates $D$, but multiplying it by $P$ yields $D^2/P$:

<table>
<thead>
<tr>
<th>Planet</th>
<th>$D$</th>
<th>$P$</th>
<th>$D/P$</th>
<th>$D^2/P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>B</td>
<td>4.0</td>
<td>8.0</td>
<td>0.5</td>
<td>2.0</td>
</tr>
<tr>
<td>C</td>
<td>9.0</td>
<td>27.0</td>
<td>0.333</td>
<td>3.0</td>
</tr>
</tbody>
</table>

The attribute $D^2/P$ is not constant, nor does vary linearly with $D$, $P$, or $D/P$. It does increase with $D$ and $P$, and increase as $D/P$ decreases. Dividing $D^2/P$ by $D$ would just yield $D/P$ again. Dividing it by $P$ would create $D^2/P^2$, which would make progress. Let us, however, multiply it by $D/P$ to create $D^3/P^2$:

<table>
<thead>
<tr>
<th>Planet</th>
<th>$D$</th>
<th>$P$</th>
<th>$D/P$</th>
<th>$D^2/P$</th>
<th>$D^3/P^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>B</td>
<td>4.0</td>
<td>8.0</td>
<td>0.5</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>C</td>
<td>9.0</td>
<td>27.0</td>
<td>0.333</td>
<td>3.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

This last attribute is constant. Thus our equation is:

$$\frac{D^3}{P^2} = c$$

which is a rediscovery of **Kepler’s 3rd Law** (the mean distance of a planet from the Sun is proportional to the square of its period around the Sun).

**Besides Kepler’s 3rd Law Bacon 1 also rediscovered Boyle’s Law**

$$PV = c$$

where $P$ is pressure, $V$ is volume, and $c$ is a constant, and **Galileo’s Law of Uniform Acceleration**

$$X = -gt^2$$

where $X$ is vertical position, $t$ is time and $g$ is a constant.

**6.2 BACON 3**

A problem with BACON 1 was that it is purely observational; it has no concept of active experimentation. To alleviate this the authors developed BACON 3.
BACON 3 was given BACON 1’s heuristics and then more to partition its attributes into independent attributes and dependent attributes. When dependent attributes exist the system tries to fixed all independent ones but one. It then tries to elucidate the relationship between the remaining independent attribute and a dependent one by varying it systematically and seeing the dependent variable’s change.

For example, let attributes $A_1$ through $A_N$ be independent attributes and let $D$ be a dependent attribute. Let the domain of $A_1$ be $\{V_{A1,1}, V_{A1,2}, V_{A1,3}, \ldots\}$, the domain of $A_2$ be $\{V_{A2,1}, V_{A2,2}, V_{A2,3}, \ldots\}$, etc. BACON 3 would fix $A_1$ to $V_{A1,1}$, then fix $A_2$ to $V_{A2,1}$, etc., until it fixes $A_{N-1}$ to $V_{N-1,1}$. This defines $\text{sizeof}(\text{domain}(A_1)) \times \text{sizeof}(\text{domain}(A_2)) \times \ldots \times \text{sizeof}(\text{domain}(A_{N-1}))$ different sets of attribute-to-value mappings where all but $A_{N-1}$ is fixed.

Within each attribute-to-value mapping set it tries to sees how $A_N$ varies with $D$. If it finds a pattern consistent across all of the $\text{sizeof}(\text{domain}(A_1)) \times \text{sizeof}(\text{domain}(A_2)) \times \ldots \times \text{sizeof}(\text{domain}(A_{N-1}))$ attribute-to-value mapping sets then BACON 3 creates an attribute relates $A_N$ with $D$ to make a constant. It then removes $A_N$ from consideration and allows $A_{N-1}$ to vary. If it gets down to $A_1$ and finds a constant relationship then it has found an equation that relates the dependent attribute with all independent ones.

For example say we have data on gas samples where we systematically varied the quantity ($n$ with domain $\{1, 2, 3\}$), temperature ($T$ with domain $\{10, 20, 30\}$) and pressure ($V$ with domain $\{1000, 2000, 3000\}$) of the gas to observe the dependent pressure ($P$). Our independent variables are $n$, $T$ and $V$, and our dependent one is $P$. BACON 3 defines 9 attribute-to-value mapping sets:

\[
\begin{align*}
(n = 1, \ T = 10) & \quad (n = 1, \ T = 20) & \quad (n = 1, \ T = 30) \\
(n = 2, \ T = 10) & \quad (n = 2, \ T = 20) & \quad (n = 2, \ T = 30) \\
(n = 3, \ T = 10) & \quad (n = 3, \ T = 20) & \quad (n = 3, \ T = 30)
\end{align*}
\]

Within each it varies $P$ systematically and find that $V$ varies inversely. It creates the new attribute $A$ as the product of $P$ and $V$.

Next it varies $T$ systematically to find a linear relationship between $A$ and $T$. This is given as $A = bT + c$ for new constants $b$ and $c$. New constants $b$ and $c$ are constant within each of the three attribute-to-value mapping sets (one for each value of $n$), but vary as $n$ varies.

Finally it varies $n$ and finds inverse relationships between it and both $b$ and $c$. This is expressible with the equations $d = b/n$ and $e = c/n$ where both $d$ and $e$ are constant across all of the data.

BACON 3 found the equation $PV = DnT + en$, which is more conventionally written as

\[PV = 8.32n(T-273)\]

and is the Ideal Gas Law. In addition to this equation BACON 3 also found Coulomb’s Law

\[\frac{F D^2}{Q_1 Q_2} = \text{const}\]

where $F$ is the force between two particles of charges $Q_1$ and $Q_2$ separated by a distance $D$. 

6.3 BACON 4

A problem with BACON 3 is that it can only handle numeric attributes. Symbolic attribute values may have other attributes that are numeric. For example, the symbolic attribute substance may have values wood, copper, and plastic, and these values may have numeric attributes like density, conductivity and specific_heat. Such attributes describe “intrinsic” properties of physical objects made from those materials.

BACON 4 was built by extending BACON 3 with an operator to define a new numeric attribute \( A_{\text{new}} \) when given the pair \((A_S, A_N)\). Here, \( A_S \) is an existing symbolic attribute and \( A_N \) is an existing numeric attribute. New attribute \( A_{\text{new}} \) will be defined to be equal to existing attribute \( A_N \), but it represents a brand new attribute. Its equivalence with \( A_N \) is just a matter of scaling.

For example, if BACON 4 was given information on nine objects giving us attribute object with domain \{A, B, C, D, E, F, G, H, I\}. These nine objects have different compositions (composition) and weights (\( w \)). Further, in this experiment all were submerged three different levels of water. Both the original volume of water (\( V \)) and the volume after submerging (\( C \)) were recorded.

<table>
<thead>
<tr>
<th>object</th>
<th>composition</th>
<th>( W )</th>
<th>( V )</th>
<th>( C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>silver</td>
<td>55.923</td>
<td>100,200,300</td>
<td>105.326, 205.326, 305.326</td>
</tr>
<tr>
<td>B</td>
<td>silver</td>
<td>74.708</td>
<td>100,200,300</td>
<td>107.115, 207.115, 307.115</td>
</tr>
<tr>
<td>C</td>
<td>silver</td>
<td>99.561</td>
<td>100,200,300</td>
<td>109.482, 209.482, 309.482</td>
</tr>
<tr>
<td>D</td>
<td>gold</td>
<td>121.841</td>
<td>100,200,300</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>gold</td>
<td>91.135</td>
<td>100,200,300</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>gold</td>
<td>170.168</td>
<td>100,200,300</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>lead</td>
<td>57.182</td>
<td>100,200,300</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>lead</td>
<td>39.820</td>
<td>100,200,300</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>lead</td>
<td>77.828</td>
<td>100,200,300</td>
<td></td>
</tr>
</tbody>
</table>

In the table above the three values for independent attribute \( V \) and dependent attribute \( C \) are given in one row for conciseness.

BACON 4’s behavior deviates from that of the previous versions in that it is able to posit new numeric attributes. Among the first thing it discovers is the linear relationship between independent variable \( V \) and dependent variable \( C \). It creates two new attributes \( S_{c,v} \) and \( I_{c,v} \), that are the slope and y-intercept respectively in the linear equation \( V = S_{c,v} \cdot C + I_{c,v} \). The values for \( S_{c,v} \) and \( I_{c,v} \) are constant for the three values of \( V \) and \( C \), but vary by object.

Among the next things that BACON 4 does is to use the attribute pair (composition,\( I_{c,v} \)) to define a new “intrinsic” numeric attribute \( o \). The values of \( o \) are the same as \( I_{c,v} \); they also are constant for the three values of \( V \) and \( C \), but vary by object. We could also have used object as the symbolic attribute of an attribute pair and created a new numeric attribute from it but because there are so many more values for object than for composition we would have less information on each specific value of object.

With new intrinsic numeric attribute \( o \) that has a unique constant for each value of object BACON 4 can see that it varies linearly with \( I_{c,v} \). This, however, is tautological because it was defined using \( I_{c,v} \).

Attribute \( o \), however, does increase with \( w \). It creates a new attribute \( x \) that results from
dividing W by o. This attribute is constant for each value of composition, and corresponds to our notion of density.

Besides density BACON 4 also rediscovered Black’s Law of Specific Heat:

\[ T_f = \frac{(M_1T_1+M_2T_2)}{(M_1+M_2)} \]

where \( M_1 \) and \( M_2 \) are the masses of two physical objects of the same composition and physical phase, \( T_1 \) and \( T_2 \) are their two initial temperatures, and \( T_f \) is their final temperature.

### 6.4 BACON 5

As the BACON programs are given more operators and more attributes their search space grows exponentially. Empirically, however, not all equations are equally likely.

Scientists use this property in their research. For example, the discovery of Coulomb’s Law, which is more conventionally given as:

\[ F = \frac{1}{(\mathbf{r} \cdot \mathbf{e})} \times \left( \frac{q_1 \cdot q_r}{r^2} \right) \]

was probably highly influenced by the structurally similar Newton’s Law of gravitation:

\[ F = G \times \left( \frac{m_1 \cdot m_r}{r^2} \right) \]

So rather than search the complete space of equations it makes sense to bias our search for equations that have a form that we are more likely to expect.

BACON 5 built of BACON 4 but search for structurally symmetric equations before non-symmetric ones if it is about the structure of the experimental apparatus, and it finds that the apparatus itself has a symmetric structure.

Using this heuristic BACON 5 found Snell’s Law of Refraction faster than BACON 4 did:

\[ \sin(\theta_1)/\sin(\theta_2) = n_1/n_2 \]

where a light beam is traveling from one substance \( s_1 \) to another \( s_2 \). Angle \( \theta_1 \) is the angle between the boundary of \( s_1 \) and \( s_2 \), and the light beam as it travels through \( s_1 \) to \( s_2 \), angle \( \theta_2 \) is the angle between the boundary of \( s_1 \) and \( s_2 \), and the light beam as it travels through \( s_2 \) away from the boundary, \( n_1 \) is the density of \( s_1 \), and \( n_2 \) is the density of \( s_2 \).

### 6.5 GLAUBER

A

### 6.6 Analysis and Summary
The BACON series, and its allied programs were limited in their discovery ability. They were not equipped to handle noise or missing values.

This limitation was a serious one in the sense that they always rediscovered rather than discovered any brand-new scientific equation. Further, they had to use data. It would have been impossible for them to posit Newton's Law of Gravity ($F = \frac{Gm_1m_2}{r^2}$) or Einstein's iconic $E = mc^2$ given the same backgrounds as Newton or Einstein because both of these equations arose from purely theoretical considerations, not empirical ones.

Yet, despite these limitations the BACONs and allies established a powerful precedent by combining empirical and theoretical heuristics. BACON 1 established a few simple empirical operators. BACON 3 extended it by differentiating between independent attributes and dependent ones. BACON 4 extended that by allowing for the discovery of intrinsic numeric attributes. BACON 5 extended that by reorganizing its behavior to search for symmetric equations first if it thinks there is reason to suspect the relationship is symmetric. In this series of programs we see the powerful techniques of constructive induction in BACON 3 and BACON 4, and attuned heuristic search in BACON 5.

Indeed the BACON series of programs started research on a series of ever more powerful equation-finding programs. . .

References: