Calculating the VC-Dimension of Decision Trees

Özlem Aslan$^1$ Olcay Taner Yıldız$^2$ Ethem Alpaydın$^1$

$^1$Department of Computer Engineering
Boğaziçi University

$^2$Department of Computer Engineering
Işık University

24th International Symposium on Computer and Information Sciences, 2009
Outline

1. Introduction
   - Model Complexity
   - VC Dimension

2. Proposed Method
   - Exhaustive Search Algorithm
   - Estimating VC-Dimension By Regression
   - Complexity Control Using VC-Dimension Estimates

3. Conclusion
1. Introduction
   - Model Complexity
   - VC Dimension

2. Proposed Method
   - Exhaustive Search Algorithm
   - Estimating VC-Dimension By Regression
   - Complexity Control Using VC-Dimension Estimates

3. Conclusion
1. Introduction
   - Model Complexity
   - VC Dimension

2. Proposed Method
   - Exhaustive Search Algorithm
   - Estimating VC-Dimension By Regression
   - Complexity Control Using VC-Dimension Estimates

3. Conclusion
Outline

1. Introduction
   - Model Complexity
   - VC Dimension

2. Proposed Method
   - Exhaustive Search Algorithm
   - Estimating VC-Dimension By Regression
   - Complexity Control Using VC-Dimension Estimates

3. Conclusion
Underfit vs Overfit

Calculating the VC-Dimension of Decision Trees
Calculating the VC-Dimension of Decision Trees

Best Model

- \( x_1 < \alpha_3 \)
- \( x_2 < \beta_4 \)

\[ \beta_4 \]

\[ x_2 \]

\[ \alpha_3 \]

\[ x_1 \]
Structural Risk Minimization

\[ E_g = E_t + \frac{\epsilon}{2} \left( 1 + \sqrt{1 + \frac{4E_t}{\epsilon}} \right) \]  

(1)

\[ \epsilon = a_1 \frac{V[\log(a_2 N/V) + 1]}{N} - \log(\nu) \]  

(2)

(Vapnik95)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_t )</td>
<td>training error</td>
</tr>
<tr>
<td>( V )</td>
<td>VC dimension of the model</td>
</tr>
<tr>
<td>( \nu )</td>
<td>confidence level</td>
</tr>
<tr>
<td>( a_1 ) and ( a_2 )</td>
<td>empirically fitted constants</td>
</tr>
<tr>
<td>( N )</td>
<td>sample size</td>
</tr>
</tbody>
</table>
Calculating the VC-Dimension of Decision Trees
Outline

1. Introduction
   - Model Complexity
   - VC Dimension

2. Proposed Method
   - Exhaustive Search Algorithm
   - Estimating VC-Dimension By Regression
   - Complexity Control Using VC-Dimension Estimates

3. Conclusion
Procedure

- An exhaustive search algorithm to calculate the exact VC-dimensions.
- Fit a regressor so that we can estimate the VC-dimension of any tree.
- VC-dimension estimates in pruning to validate that they are indeed good estimates.
Illustration

Calculating the VC-Dimension of Decision Trees
Computational Complexity

\[ \sum_{N=1}^{\infty} \binom{2d}{N} 2^N |H| \]

- The full tree with depth 4 and for 4 input features requires 2 days to complete on a quad-core computer.
- Depth 5 and for 5 input features will require approximately $10^{13}$ days.
- We can run the exhaustive search algorithm only on few $H$ and on cases with small $d$ and $|H|$. 
Experimental Setup

- We thoroughly searched decision trees with depth up to four.
- We use the fact that two isomorphic trees have the same VC dimension.
Regression Model

154 training instances

\[ V = 0.7152 + 0.6775 V_I + 0.6775 V_r - 0.6600 \log d + 1.2135 \log M \]

\( R^2 \) is 0.9487.
Experimental Setup

- CVprune
- SRMprune
- NOprune
Experimental Setup

Functions:

\[
F_1 = x_0x_1 + x_0x_2 + x_1x_2 \\
F_2 = x_0x_1 + x_0x_2 + x_0x_3 + x_1x_2 + x_1x_3 + x_2x_3 \\
F_3 = x_0x_1' + x_0'x_1
\]

- The number of input features \( d = 8 \) and \( d = 12 \)
- Five different noise levels \( \rho = 0.0, 0.01, 0.05, 0.1, \) and \( 0.2 \).
- Four different sample size percentage \( S = 10, 25, 50, 100 \).
$d = 12$, $\rho = 0.0$, and $S = 100$

<table>
<thead>
<tr>
<th>Function</th>
<th>Error Rate</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NOprune</td>
<td>CVprune</td>
</tr>
<tr>
<td></td>
<td>NOprune</td>
<td>CVprune</td>
</tr>
<tr>
<td>$F_1$</td>
<td>0.0± 0.0</td>
<td>0.0± 0.0</td>
</tr>
<tr>
<td>$F_2$</td>
<td>0.0± 0.0</td>
<td>0.0± 0.0</td>
</tr>
<tr>
<td>$F_3$</td>
<td>3.9± 2.8</td>
<td>8.5± 7.0</td>
</tr>
</tbody>
</table>
Complexity Control Results

$\rho = 0.2$, $S = 100$, and $F = F_2$

<table>
<thead>
<tr>
<th>$d$</th>
<th>Error Rate</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NO prune</td>
<td>CV prune</td>
</tr>
<tr>
<td>8</td>
<td>38.1 ± 4.1</td>
<td>37.8 ± 5.3</td>
</tr>
<tr>
<td>12</td>
<td>35.5 ± 1.2</td>
<td>28.2 ± 3.0</td>
</tr>
</tbody>
</table>
\( \rho = 0.2, \ S = 100, \) and \( F = F_2 \)

<table>
<thead>
<tr>
<th>( d )</th>
<th>( d )</th>
<th>Error Rate</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d )</td>
<td>( NO ) prune</td>
<td>( CV ) prune</td>
<td>( SRM ) prune</td>
</tr>
<tr>
<td>8</td>
<td>38.1± 4.1</td>
<td>37.8± 5.3</td>
<td>35.3± 2.7</td>
</tr>
<tr>
<td>12</td>
<td>35.5± 1.2</td>
<td>28.2± 3.0</td>
<td>21.0± 0.6</td>
</tr>
</tbody>
</table>
$d = 12$, $S = 50$, and $F = F_1$

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>Error Rate</th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NO prune</td>
<td>CV prune</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0± 0.0</td>
<td>0.0± 0.0</td>
</tr>
<tr>
<td>0.01</td>
<td>3.6± 0.5</td>
<td>1.5± 0.3</td>
</tr>
<tr>
<td>0.05</td>
<td>12.2± 0.8</td>
<td>5.0± 0.5</td>
</tr>
<tr>
<td>0.1</td>
<td>21.7± 0.9</td>
<td>12.8± 4.7</td>
</tr>
<tr>
<td>0.2</td>
<td>35.7± 1.4</td>
<td>29.3± 5.4</td>
</tr>
</tbody>
</table>
$d = 8$, $\rho = 0.05$, and $F = F_3$

<table>
<thead>
<tr>
<th>S</th>
<th>Error Rate</th>
<th></th>
<th>Number of Nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NO prune</td>
<td>CV prune</td>
<td>SRM prune</td>
</tr>
<tr>
<td>100</td>
<td>19.0±5.9</td>
<td>25.3±14.9</td>
<td>15.8±8.6</td>
</tr>
<tr>
<td>50</td>
<td>23.7±14.7</td>
<td>28.9±17.2</td>
<td>23.4±14.6</td>
</tr>
<tr>
<td>25</td>
<td>27.0±11.7</td>
<td>37.4±15.7</td>
<td>27.0±11.7</td>
</tr>
<tr>
<td>10</td>
<td>41.7±17.1</td>
<td>45.0±17.2</td>
<td>41.7±17.1</td>
</tr>
</tbody>
</table>
Outline

1 Introduction
   - Model Complexity
   - VC Dimension

2 Proposed Method
   - Exhaustive Search Algorithm
   - Estimating VC-Dimension By Regression
   - Complexity Control Using VC-Dimension Estimates

3 Conclusion
Conclusion

- VC-dimension calculation by exhaustive search
- Estimation of VC-dimension via regression
- VC-dimension used in SRM based model selection
- Find trees that are as accurate as in CV pruning
Future Work

- The approach can easily be extended to univariate decision trees with discrete and/or continuous features.
- Extension to $K$-class
Extension

Discrete features with 3 values:

\[ V = -3.0014 + 0.5838 V_1 + 0.5838 V_2 + 0.5838 V_3 + 2.5312 \log d + 1.9064 \log M \]

\( R^2 \) is 0.91.

4 values:

\[ V = -1.6294 + 0.5560 V_1 + 0.5560 V_2 + 0.5560 V_3 + 0.5560 V_4 + 3.9830 \log d - 0.4073 \log M \]

\( R^2 \) is 0.861.
Extension

Discrete features with 5 values:

\[ V = 14.4549 + 0.3924V_1 + 0.3924V_2 + 0.3924V_3 + 0.3924V_4 + 0.3924V_5 - 4.7687 \log d - 1.3857 \log M \]

\( R^2 \) is 0.782.