Attribute Selection and Parameter Selection

I211: Information infrastructure II
Cross-validation is used to estimate predictor's performance. The question is if we are using KNN, what is the best $K$ we should use for each data set $D$? Or if we are performing attribute selection, how should we find the optimal combination of attributes?
4-Fold CV

Randomly and evenly split into 4 non-overlapping partitions

\( D \)

20 data points

Partition 1.
Data points: 1, 3, 5, 15, 16

Partition 2.
Data points: 6, 10, 11, 14, 17

Partition 3.
Data points: 4, 9, 12, 19, 20

Partition 4.
Data points: 2, 7, 8, 13, 17
Step 1: Use partition 1 as test and partitions 2-4 as training – just like before.

We want to do attribute selection on the training set and keep only the best attributes for the KNN classifier.

Here we will analyze filtering approaches to attribute selection.

Attribute selection must be done on the training set!
Attribute selection

Training data

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>0.4</td>
<td>1</td>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>3.2</td>
<td>0</td>
</tr>
</tbody>
</table>

How many possible attribute subsets?

This data contains 3 attributes. Possible situations are

- attribute 1 only
- attribute 2 only
- attribute 3 only
- attributes 1 and 2
- attributes 1 and 3
- attributes 2 and 3
- all three attributes (1, 2, 3)

How many subsets are possible for 3 features?

7 possible subsets ($2^3 - 1$). Empty set is not counted.
Attribute selection

How many possible attribute subsets if we have $k$ attributes?

$2^k - 1$

This is a hard computational problem and generally cannot be used.

For example, if $k = 50$, there are $2^{50} - 1$ non-empty subsets. This is about $1.1 \cdot 10^{15}$.

If each evaluation takes 1 second, this would take about 35 million years on a single computer.

The problems where computing time depends on a parameter that is in the exponent (e.g. $2^k$) are called exponential problems and cannot be tackled directly, that is, using brute-force algorithms!
Filtering Approach to Attribute Selection

To avoid exponential time, we will check attributes individually. This approach is called filtering because the method of checking attribute’s quality does not depend on classification algorithm.

**Approach (for binary classification or regression):**

1. Calculate correlation coefficient $\rho$ between each attribute and class.

2. Select top 5 (or 10, or some other number of attributes) or pick a threshold and then keep all attributes whose absolute value of the correlation coefficient with the class variable is greater than some threshold.

There are many ways to select top attributes. Correlation coefficient is only one of them!

Training data

$$
\begin{bmatrix}
0 & 0 & 6 & 0 \\
1 & 1 & 4 & 1 \\
0.4 & 1 & -2 & 1 \\
1 & 2 & 3 & 0 \\
0 & 0 & -1 & 0 \\
0 & 1 & 3.2 & 0
\end{bmatrix}
$$

$p(\text{attribute 1, class}) = 0.47$

$p(\text{attribute 2, class}) = 0.17$

$p(\text{attribute 3, class}) = -0.30$
Attribute Selection

Say, we decide to go with top 5 features. We can then evaluate this algorithm.

1) We select top 5 attributes using training set
2) We keep only these 5 attributes in the test set
3) Only selected attributes are used to calculate distances for KNN.
4) Accuracy is reported using N-fold cross-validation.
Selecting the number of attributes to retain

If our goal is to automatically select the best number of attributes to retain, it is generally not acceptable to find accuracy using top 5, top 10, top 15 etc. and then after seeing accuracy on the test set decide how many attributes to retain.

The reason is: you can’t use the test set to select attributes on the training set (this is called information leak). If this is done, it usually leads to overestimates of performance.
Selecting the number of attributes to retain

Why does it lead to overestimates?

Because, just by chance, one of the many possibilities will give you good performance and you will think that this is indeed the performance of your algorithm.

One should be careful with testing too many parameters.
Solution - Validation Set

IDEA: Further split Training set into Training and Validation sets. Use training to select top n features, then use validation set to pick the top one. Then evaluate accuracy on the Test set. Similarly proceed with other N-1 folds.
Solution - Validation Set

Once number $n$ is chosen (using validation set), top $n$ attributes are selected on the entire training set and when distances between data points from the test set and training set are calculated, the entire training set can be used.
Validation set is a general approach to selecting parameters. For example, what should be $K$ in KNN algorithm? We can select it using validation set and then estimate accuracy of the algorithm using the best $K$. 