In the Name of God
Machine Learning (Spring 2020)
Assignment #1_Bonus
Feating

Due date: 31st of Tir

In this assignment, you are about to implement a Feating ensemble. Feating (Feature Subspace Aggregating) subdivides the feature-space into non-overlapping local regions in a single subdivision; and ensures that different subdivisions provide the distinct local neighborhoods for each point in the feature space. There are many ways a feature-space can be subdivided. Instead of using heuristics, Feating subdivides the feature-space exhaustively based on a user-specified number of features to control the level of localisation. The set of exhaustive feature-space subdivisions forms the basis to develop a new ensemble method which aggregates all local models or a random subset of these local models.

Let X denote the input n-dimensional feature space, A_d denote the d-th feature, Y denote the set of classes representing the underlying concept, m denote training set size, n denote number of features, and $D := \{(x_a, y_a) \mid a \in \{1, 2, . . . , m\}\}$ denote the training set.

An example of Feating in binary domains

In a domain of n binary attributes, the feature space can be subdivided into two half-spaces n ways, where each such subdivision uses one of the n attributes. Extending to subdivision using h attributes, the feature-space can be subdivided into $\frac{1}{2^h}$ -spaces in C_n^h ways, where h is the number of binary attributes used to do the subdivision. In general, an h-subdivision is one of the possible non-overlapping subdivisions of the feature-space using h attributes. The exhaustive set has C_n^h h-subdivisions for a given feature-space defined by n attributes.

For example, if the feature space is defined by four binary attributes A_1, A_2, A_3 and A_4 and

$$x = \langle A_1 = 0, A_2 = 1, A_3 = 1, A_4 = 0\rangle$$

for *h* = 2, six sets of quarter-spaces r_i of two dimensions are shown below, where each column indicates the attributes used (with an implied value for each attribute) in each quarter-space:

$$\begin{array}{lllllll} A_1 & A_1 & A_1 & A_2 & A_2 & A_3 & Level\,1 \\ A_2 & A_3 & A_4 & A_3 & A_4 & A_4 & Level\,2 \end{array}$$

which define all six local neighborhoods for *x*. This example shows the enumeration method we employed to generate C_n^h h-subdivisions in a Feating ensemble, which can be done without attribute selection. This is implemented as function ‘attributeList’ shown in the pseudo code This requires a ranking of attributes to be done beforehand. This ranking is implemented using information gain in the function ‘rankAttribute’.

The data structure which implements the Feature-Subspace Aggregating mentioned above, is called a *Level Tree. Level Tree* is a restricted form of decision tree where each node at a single level of the tree must use the same attribute. Figure 1 shows an example of a *Level Tree* defined by three attributes: A_1, A_2 and A_3. A local model is trained from data in each leaf of the tree and it is attached to the leaf. To classify an example x, one first traverses the tree from the root to a leaf following the branches that x satisfies. Then the model at that leaf is applied to x to produce a classification. A *Level Tree* with local models is thus equivalent to a single global model, ready to predict when a test instance is presented. In the *Level Tree* of Figure 1*,* h is equal to 3.

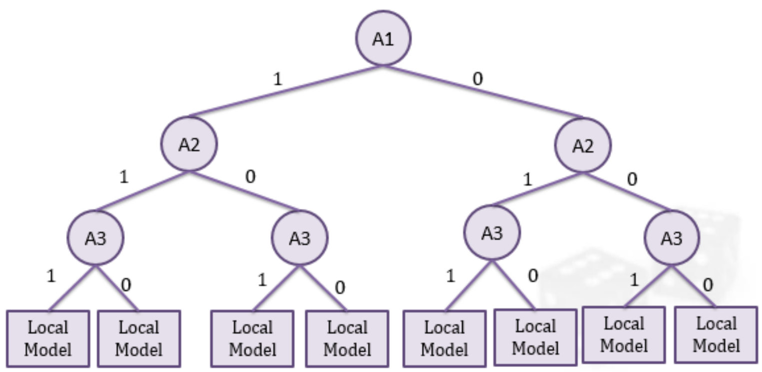

Figure 1: An example of Level Tree with three levels of localisation using attributes: A1 at level 1, A2 at level 2, and A3 at level 3, and a local model attached to each leaf. Each Level Tree with local models forms a single global model

Though each attribute can appear at one level only in a *Level Tree*, the cut-points used for a numeric attribute on different nodes of the same level can be different. The training data is filtered through the branches in the *Level Tree* as each node is created. If the training data is less than some minimum number (n_{min}) before a branch reaches the required level h or all instances belong to the same class, a leaf is formed. The default setting, n_{min} = 4, should be used .

The aggregation of all possible *Level Trees* of h-subdivision forms an ensemble. To make the final prediction, the predictions from all *Level Trees* in an ensemble are aggregated using a simply majority vote, like in bagging. The value for h can be $\{1,2,3\}$ in your experiments.

Feating employs three different base learning algorithms, J48 (decision tree), IBk (k-nearest neighbour where k = 5) and SVM in the leaves.

See the table below for an example of h and C_n^h values to better understand the number of used *Level Trees* for each h.

Data Sets	Data Size	Feating's Ensemble Size		
		$h=1$	$h=2$	$h=3$
coding	20,000	15	105	455
nursery	12,960	8	28	56
dna	3,186	60	1,770	34,220
wave40	5,000	40	780	9,880
satimage	6,435	36	630	7,140
wave21	5,000	21	210	1,330
segment	2,310	19	171	969
anneal	898	38	703	8,436

C_2^{15}

C_3^{15}

Table 1: Ensemble sizes for different levels of Feating for some data sets. The ensemble size for $h = 1$ is also the number of attributes in each data set

The pseudo codes for constructing the *Level Tree* and Feating ensemble are presented in the following figures.

Algorithm 1: Feating(D, A, h)
– Build a set of Level Trees based on Feating

INPUT D—Training set, A—the set of given attributes, h—the maximum level of Level Tree

OUTPUT E—a collection of Level Trees

$E \leftarrow \emptyset$
$n \leftarrow |A|$
$N \leftarrow C_n^h$
$P \leftarrow \text{rankAttribute}(A)$
for $i = 1$ to N **do**
  /* Construct an attribute list from P based on index i */
  $L \leftarrow \text{attributeList}(P, i)$
  $E \leftarrow E \cup \text{BuildLevelTree}(D, L, 0)$
end for
Return E

Algorithm 2: BuildLevelTree(D, L, j)
– Build a Level Tree recursively

INPUT D—Training set, L—Attribute set, j—Current tree level
OUTPUT *node*—Level Tree node

Global variables:
/* B is the base learning algorithm */
/* h is the maximum level of the Level Tree */
if $j = h$ **then**
  node.localModel $\leftarrow$ BuildLocalModel(D, B)
  Return *node*
end if
/* n_{min}—#instances required before a split is allowed. */
if *D belongs to the same class or* $|D| < n_{min}$ **then**
  Return a leaf labelled with the majority class
end if
/* Retrieve next attribute from L based on current level */
$v \leftarrow$ nextAttribute(L, j)
Construct a *node* with attribute v
if v is a numeric attribute **then**
  node.splitpoint $\leftarrow$ findSplitPoint(v, D)
  $D_1 \leftarrow$ filter(D, $v >$ *node.splitpoint*)
  $D_2 \leftarrow$ filter(D, $v \leq$ *node.splitpoint*)
  node.branch(1) $\leftarrow$ BuildLevelTree(D_1, L, $j+1$)
  node.branch(2) $\leftarrow$ BuildLevelTree(D_2, L, $j+1$)
  $m \leftarrow 2$
else
  /* Split based on nominal attributes */
  let $\{v_1, \ldots, v_m\}$ be the possible values of v
  for $i = 1$ to m **do**
    $D_i \leftarrow$ filter(D, $v = v_i$)
    node.branch(i) $\leftarrow$ BuildLevelTree(D_i, L, $j+1$)
  end for
end if
if $\sum_{i=1}^{m} |D_i| < |D|$ **then**
  /* Collect all instances with missing v values to form a new branch */
  $D_{m+1} \leftarrow$ filterMissingValue(D, v)
  node.branch($m+1$) $\leftarrow$ BuildLevelTree(D_{m+1}, L, $j+1$)
end if
Return *node*

Use three data sets of the Table 1 for experiments. Split the data set into train and test parts. Use 70% of the data for the training phase and the remaining 30% for the testing phase. Run your codes for 10 individual runs and report the mean and standard deviation of 10 runs for each performance metric.

In the Name of God
Machine Learning (Spring 2020)
Assignment #1_Bonus
Feating

For more information regarding the Feating algorithm, please see the attached paper.

Questions:

What does “Feating is an aggregation of local models” mean?

Why is localization important for us in classification?

How does Level Tree perform feature space division?

Important Notes:

- Pay extra attention to the due date. It is fixed and **will not be extended**.
- Be advised that no submissions after the deadline would **be graded**.
- Be sure to comment your code. Also, include instructions on how to run your code (if necessary).
- Prepare a complete report for your assignment and answer all the questions.
- The name of the uploading file should be your **Lastname_Firstname.**

Good Luck