CS6220 Data Mining Techniques

Assignemnt 2

Yaxin Huang

Feb 14th, 2015

Question 1

(I) The complete decision tree trained by using the training data is the one below. The bold nodes are the nodes selected to form the decision tree; The blued information gains are the larger information gains compared to the ones got from split using other features:

Top [6L 4H], Entropy = 0.9710
If split by Education, IG = 0.1246
(1) High School [4L 1H], Entropy = 0.7219
If split by Career, IG = 0.2628
(1.1) Management [2L 1H], Entropy = 0.9183
(1.2) Service [2L 0H], Entropy = 0
If split by Experience, IG = 0.3219
(1.3) Less than 3 [1L 0H], Entropy = 0
(1.4) 3 to 10 [2L 0H], Entropy = 0
(1.5) More than 10 [1L 1H], Entropy = 1
If split by Career, IG = 1
(1.5.1) Management [0L 1H], Entropy = 0
(1.5.2) Service [1L 0H], Entropy = 0
(2) College 2L 3H, Entropy = 0.9710
If split by Career, IG = 0.4200
(2.1) Management [0L 2H], Entropy = 0
(2.2) Service [2L 1H], Entropy = 0.9183
If split by Experience, IG = 0.9183
(2.2.1) Less than 3 [1L 0H], Entropy = 0
(2.2.2) 3 to 10 [0L 1H], Entropy = 0
(2.2.3) More than 10 [1L 0H], Entropy = 0
If split by Experience, IG = 0.1710
(2.3) Less than 3 [1L 1H], Entropy = 1
(2.4) 3 to 10 [0L 1H], Entropy = 0
(2.5) More than 10 [1L 1H], Entropy = 1

If split by Career, IG = 0.1246
(3) Management [2L 3H], Entropy = 0.9710
(4) Service [4L 1H], Entropy = 0.7219

If split by Experience, IG = 0.0200
(5) Less than 3 [2L 1H], Entropy = 0.9183
(6) 3 to 10 [2L 1H], Entropy = 0.9183
(7) More than 10 [2L 2H], Entropy = 1

(II) Prune the tree using the validation data. Every tree node will be examined. The following boxes contains the branches being considered to be pruned at each step. The underscored tree node is the node connecting these branches.

If split by Experience, IG = 0.3219
(1.3) Less than 3 [1L 0H], Entropy = 0
(1.4) 3 to 10 [2L 0H], Entropy = 0
(1.5) More than 10 [1L 1H], Entropy = 1
If split by Career, IG = 1
(1.5.1) Management [0L 1H], Entropy = 0
(1.5.2) Service [1L 0H], Entropy = 0

If we keep the branches under tree node (1.5) :

Node	Trained Label	True Labels	Errors
1.5.1	High	0 Low 2 High	0
1.5.2	Low	1 Low 0 High	0

If we remove the children of node (1.5) and treat it as leaf node:

Node	Trained Label	True Labels	Errors
1.5	High (1L 2H)	1 Low 2 High	1

Since treating node (1.5) as a leaf node increases the number of errors, so we don't prune it.

If split by Education, IG = 0.1246
(1) High School [4L 1H], Entropy = 0.7219
If split by Experience, IG = 0.3219
(1.3) Less than 3 [1L 0H], Entropy = 0
(1.4) 3 to 10 [2L 0H], Entropy = 0
(1.5) More than 10 [1L 1H], Entropy = 1
If split by Career, IG = 1
(1.5.1) Management [0L 1H], Entropy = 0
(1.5.2) Service [1L 0H], Entropy = 0

If we keep tree node (1):

Node	Trained Label	True Labels	Errors
1.3	Low	1 Low 0 High	0
1.4	Low	2 Low 0 High	0

1.5.1	High	0 Low 2 High	0
1.5.2	Low	1 Low 0 High	0

If we prune the branches as treat node (1) as a leaf node:

Node	Trained Label	True Labels	Errors
1	Low (4L 2H)	4 Low 2 High	2

Pruning the subtree will result in increase in errors, so we keep the node (1).

If split by Career, IG = 0.4200
(2.1) Management [0L 2H], Entropy = 0
(2.2) Service [2L 1H], Entropy = 0.9183
If split by Experience, IG = 0.9183
(2.2.1) Less than 3 [1L 0H], Entropy = 0
(2.2.2) 3 to 10 [0L 1H], Entropy = 0
(2.2.3) More than 10 [1L 0H], Entropy = 0

If we keep node (2.2):

Node	Trained Label	True Labels	Errors
2.2.1	Low	1 Low 0 High	0
2.2.2	High	1 Low 1 High	1
2.2.3	Low	1 Low 0 High	0

If we prune the subtree and tree node (2.2) as a leaf node:

Node	Trained Label	True Labels	Errors
2.2	Low (3L 1H)	3 Low 1 High	1

Since pruning the subtree does not increase the number of errors and pruning the branches will decrease the complexity of the tree, the subtree is pruned and the resulting one is:

If split by Career, IG = 0.4200
(2.1) Management [0L 2H], Entropy = 0
(2.2) Service [2L 1H], Entropy = 0.9183

(2) College 2L 3H, Entropy = 0.9710
If split by Career, IG = 0.4200
  (2.1) Management [0L 2H], Entropy = 0
  (2.2) Service [2**L** 1H], Entropy = 0.9183

If we keep node (2):

Node	Trained Label	True Labels	Errors
2.1	High	1 Low 2 High	1
2.2	Low	3 Low 1 High	1

If we prune the subtree and treat node (2) as a leaf node:

Node	Trained Label	True Labels	Errors
2	Low (4L 3H)	4 Low 3 High	3

Pruning the subtree increases the number of errors. So the node(2) and its subtree is kept.

All the tree nodes (except the root node) have been examined and the resulting decision tree is:

Top [6L 4H], Entropy = 0.9710
Split by Education, IG = 0.1246
  (1) High School [4L 1H], Entropy = 0.7219
  Split by Experience, IG = 0.3219
    (1.3) Less than 3 [1L 0H], Entropy = 0
    (1.4) 3 to 10 [2L 0H], Entropy = 0
    (1.5) More than 10 [1L 1H], Entropy = 1
    Split by Career, IG = 1
      (1.5.1) Management [0L 1H], Entropy = 0
      (1.5.2) Service [1L 0H], Entropy = 0
  (2) College 2L 3H, Entropy = 0.9710
  Split by Career, IG = 0.4200
    (2.1) Management [0L 2H], Entropy = 0
    (2.2) Service [2L 1H(Low)], Entropy = 0.9183

Question 2

The codes for Q2 are in the folder “q2_codes”. Please check the README file to see how to run the coes and get output result.

Question 3

1. PolyKernel with exponent = 1

Correctly Classified Instances	717	84.7518 %
Incorrectly Classified Instances	129	15.2482 %

2. PolyKernel with exponent = 2

Correctly Classified Instances	810	95.7447 %
Incorrectly Classified Instances	36	4.2553 %

3. PolyKernel with exponent = 4

Correctly Classified Instances	791	93.4988 %
Incorrectly Classified Instances	55	6.5012 %

4. RBFKernel with gamma = 0.01

Correctly Classified Instances	614	72.5768 %
Incorrectly Classified Instances	232	27.4232 %

5. RBFKernel with gamma = 1.0

Correctly Classified Instances	764	90.3073 %
Incorrectly Classified Instances	82	9.6927 %

For detailed outputs, you can check the log files in folder “q3_weka_outputs”.

From the above records we can see that the PolyKernel with exponent = 1 and RBF Kernel with gamma = 0.01 do not have as good performance as other configurations.

The reason that the PolyKernel with exponent = 1 works not so well might be the data itself is not separable. The PolyKernel with exponent = 1 keeps the original space so the unseparable property is kept.

The reason that the RBF Kernel with gamma = 0.01 works not so well might be the influence of a single example is too far, and this allows the outliers to impact the result a lot. (Reference: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html)

Question 4

The kernel can be wriiten as:

$$K(x,z)=x_1\cdot z_1+x_1\cdot e^{z_2}+z_1\cdot e^{x_2}+e^{x_2+z_2}$$

$$K(x,z)=x_1\cdot z_1+x_1\cdot e^{z_2}+z_1\cdot e^{x_2}+e^{x_2}\cdot e^{z_2}$$

$$K(x,z)=x_1\cdot z_1+x_1\cdot e^{z_2}+e^{x_2}(z_1+e^{z_2})$$

$$K(x,z)=x_1(z_1+e^{z_2})+e^{x_2}(z_1+e^{z_2})$$

$$K(x,z)=(x_1+e^{x_2})(z_1+e^{z_2})$$

$$K(x,z)=\Phi^T(x)\cdot\Phi(z)$$

So K(x, z) is a dot product of two functions. If you switch x and z, then K(z,x) = K(x,z). Therefore K(x,z) is symmetric and it is a valid kernel function.

Question 5

x1	**x2**	**x3**	**y**
1	0	1	1
1	1	1	-1
1	0	0	1
1	1	0	1

α = (−0.8, 1, 6.4, −1.9)

Using this formula:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

w = (-0.8) * 1 * <1,0,1> + 1 * (-1) * <1,1,1> + 6.4 * 1 * <1,0,0> + (-1.9) * 1 * <1,1,0>
= <-0.8,0,-0.8> + <-1,-1,-1> + <6.4,0,0> + <-1.9,-1.9,0>
= <2.7, -2.9, -1.8>

solve for b using any SV: $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$

We use the first datapoint:
b = 1/1 – transpose(w) * <1,0,1>
  = 0.9

Test item : (1, 0.8, 1)
Label = sign(transpose(w) * x + b)
  = sign(-1.42 + 0.9)
  = sign(-0.52)

So the test item should be classified as -1.