[Wekalist] Mean absolute error in classification

Peter Reutemann fracpete at cs.waikato.ac.nz
Thu May 31 22:52:27 NZST 2007

- Previous message: [Wekalist] Mean absolute error in classification
- Next message: [Wekalist] SMO turn off normalization
- **Messages sorted by:** [date] [thread] [subject] [author]

```
> how is the mean absolute error calculated in classification? In numeric
> prediction it is: {|p1-a1|+....+|pn-an|}/n. But what is the difference of
> predicted values pi and actual value ai in classification?
> I am trying to figure out how weka came out with the following value of
> the Mean absolute error:
>
> Correctly Classified Instances        4520               47.494  %
> Incorrectly Classified Instances      4997               52.506  %
> Mean absolute error                      0.2717
> Total Number of Instances             9517
>
>     a    b    c    d    e   <-- classified as
>   566  589  226   31    1 |    a = '(-inf--0.0025]'
>   209  647  364   71    8 |    b = '(-0.0025--0.0015]'
>   181  681  708  390   60 |    c = '(-0.0015-0]'
>    31  134  367 1178  823 |    d = '(0-0.0015]'
>     1   17   66  747 1421 |    e = '(0.0015-inf)'
```

For each instance in the test set, Weka obtains a distribution (for each class label a value from 0 to 1, i.e., 0-100%). This distribution is matched against the expected distribution (the expected class label has 1 in that array, the others 0). For each class label the following is calculated:

```
  AbsErrPerLabel = abs(actual - predicted)/# of class labels
```

The absolute error per Instance is than the sum of these:

```
  AbsErrPerInstance = Sum(AbsPerLabel)
```

(Note: The instance weight is taken into account as well. But this is normally just 1.)

The mean absolute error is the sum over all the instances and their AbsErrPerInstance divided by the number of instances in the test set with an actual class label (that should normally be all of them).

```
  MeanAbsErr = Sum(AbsErrPerInstance) / # inst. with class label
```

See the following methods in the weka.classifiers.Evaluation class:

```
  updateStatsForClassifier(double[],Instance)
  updateNumericScores(double[],double[],weight)
```

Here are the distributions that Peter mentions in his answer:

>> For each instance in the test set, Weka obtains a distribution (for each
>> class label a value from 0 to 1, i.e., 0-100%). This distribution is
>> matched against the expected distribution (the expected class label has 1
>> in that array, the others 0). For each class label the following is
>> calculated:
>>   AbsErrPerLabel = abs(actual - predicted)/# of class labels

It took me too long to realize that Peter was referring to the values under the column distribution (and the others after that) of the classifications table shown above. For example, for the instance 1, the distribution given by Weka is:

```
   0  0.024390243902439025  0.975609756097561
```

(note that it adds up to 1; the order is the same as the order of the labels: fist = Setosa, second = Versicolor & third = Virginica)

I personally think that "distribution" it's a very vague name.
I would rather call them something like prediction scores maybe, as distribution can be many things in this context (for example, the actual distribution of clases in the dataset).

Anyway, in the case of this instance, the error is very simple to calculate.
First, the Expected distribution for the instance would be:

```
   0 0 1
```

Since it's an instance of Iris virginica. Then the error is:

```
   abs(0 - 0)/3 + abs(0.024390243902439025 - 0)/3 + abs(0.975609756097561 -
1)/3
   = 0.01626016
```

Repeating this for all the instances and summing up, I get 5.246992, which divided by 150 is 0.0349799, and that's the same answer I get with Weka.

Well, it took me some time to write all this, but at least it can be helpfull for someone else.

Cheers,
Juan Manuel

*Is not too hard to replicate this using the GUI: go to "More options..." in the Classfy tab and then configure the "Output predictions" to generate a CSV table. However, you will get a file with + and * signs in the middle of numeric columns, which are not a small anoyance.