Model creation

The first understanding:

1. Calculation of coefficients

The second understanding

1. Calculation of coefficients
2. Selection of threshold

For the first case threshold is something “universal” for database.

For second case we should find threshold for each model!

For 10 folds

TotalErr = 0

For it = 1 to nRep

Error = 0

For i=1 to 10

Read testSetIndex from text file

TrainingSet = data(~testSetIndex)

TestSet = data(testSetIndex)

Coeff=FitModelTo(TrainingSet)

Score = model.predict(TrainingSet)

Thres = selectThreshold(Score, Labels(~testSetIndex))

# Calculation of test set error

Score = model.predict(TestSet)

Outcome = np.where(Score > Thres, 1, 0)

Err = sum(Outcome <> Labels(testSetIndex))

Error = Error + Err

Next i

TotalErr = TotalErr + Error / N

Next it

TotalErr = TotalErr / nRep

Dishonest approach

Coeff=FitModelTo(data)

Score = model.predict(data)

Thres = selectThreshold(Score, Labels)

TotalErr = 0

For it = 1 to nRep

Error = 0

For i=1 to 10

Read testSetIndex from text file

TrainingSet = data(~testSetIndex)

TestSet = data(testSetIndex)

Coeff=FitModelTo(TrainingSet)

# Calculation of test set error

Score = model.predict(TestSet)

Outcome = np.where(Score > Thres, 1, 0)

Err = sum(Outcome <> Labels(testSetIndex))

Error = Error + Err

Next i

TotalErr = TotalErr + Error / N

Next it

TotalErr = TotalErr / nRep

We have – function to minimise.

We have weights .

Your procedure

x = random weights

it = 0

While (it < maxit)

G = gradient(f(x));

# Training

F1 = f(x)

x = x – G\*h

it = it + 1

F2 = f(x)

If F2 < F1

h = h \* 2

else

h = h/2

end

endW

h = 0.01

While (h<0.000001) # (F1-F2)/F1 < 0.001

G = gradient(f(x));

# Training

F1 = f(x)

x = x – G\*h

it = it + 1

F2 = f(x)

If F2 < F1

While F2 <= F1

x = x – G\*h

h = h \* 2

F1 = F2

F2 = f(x)

endW

h = h/2

x = x + G\*h

else

While F2 > F1

h = h / 2

x = x + G\*h

F2 = f(x)

endW

end

endW