F20DL and F21DL: Part 2: Machine Learning Lecture 6.2: Linear Classifiers, ctd

Katya Komendantskaya

Course Feedback on Vision

- closes on Monday, 19th November
- please find 10 mins or so to give constructive feedback

Our progress

last time, we derived the Linear Regression by working with derivatives of the error function.

Our progress

last time, we derived the Linear Regression by working with derivatives of the error function.
Today:

- finishing regression
- starting Neural Nets

Customer transactions, converted to numeric

Trans.	Music on CD?	Music on MP3?	Board Games	On-line Games	Output
T1	0	1	0	1	1
T2	1	0	0	0	0
T3	1	0	0	1	1
T4	1	0	1	0	0
T5	0	1	0	0	0
T6	0	1	1	0	0
T7	0	0	0	1	1
T8	0	1	1	1	0
T9	1	1	0	0	0
T10	1	1	0	1	1

Running this data set in Weka

Functions ⇒ Linear Regression:
(Use "More Options" ⇒ "Output Predictions")
Gives output:

```
Linear Regression Model

Buys =

     -0.4     * gamesHard +
      0.72    * gamesOnline +
      0.16
```

Running this data set in Weka

Functions ⇒ Linear Regression:
(Use "More Options" ⇒ "Output Predictions")
Gives output:

```
Linear Regression Model

Buys =

     -0.4    * gamesHard +
      0.72   * gamesOnline +
      0.16
```

How do you interpret this, in "human" terms?

Accuracy?

```
=== Predictions on test data ===

    inst#     actual  predicted      error
        1      1          0.846     -0.154
        1      1          0.846     -0.154
        1      0          1          1
        1      0          0.222      0.222
        1      0          0.222      0.222
        1      1          0.846     -0.154
        1      1          0.846     -0.154
        1      0         -0.387     -0.387
        1      0          0.222      0.222
        1      0         -0.387     -0.387
```

Functions for Classification

In classification tasks, there are normally two values - 0 and 1, so linear function is not well suited.
For classification, one uses **squashed linear function** of the form

$$f(X_1, \ldots, X_n) = G(w_0 + w_1X_1 + \ldots + w_nX_n)$$

where G is **an activation function** from real numbers to $[0, 1]$.

Functions for Classification

In classification tasks, there are normally two values - 0 and 1, so linear function is not well suited.
For classification, one uses **squashed linear function** of the form

$$f(X_1, \ldots, X_n) = G(w_0 + w_1 X_1 + \ldots + w_n X_n)$$

where G is **an activation function** from real numbers to $[0, 1]$.

Example: A step function

$S(x) = 1 \text{ if } x \geq 0 \text{ and } S(x) = 0 \text{ if } x \leq 0$

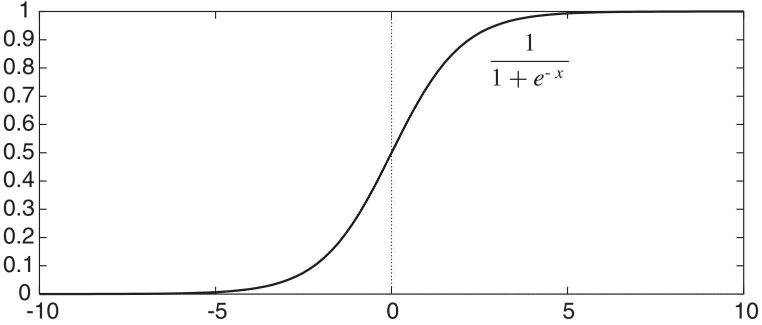
- ... was used in Perceptron [Rosenblatt, 1958] - one of the first methods for learning.
- Disadvantage: not differentiable.

Sigmoid (logistic) activation function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid (logistic) activation function

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



If the function is differentiable...

we can use gradient descent to update the weights:
For the sigmoid function σ, the derivative is
$\sigma'(x) = \sigma(x) \times (1 - \sigma(x))$
We use this to change line 15 in the algorithm *LinearLearner* to

$$w_i := w_i + \eta \times \delta \times pval^{\overline{\omega}}(e, Y) \times [1 - pval^{\overline{\omega}}(e, Y)] \times val(e, X_i).$$

where $pval^{\overline{\omega}}(e, Y) = \sigma(\sum_i w_i \times val(e, X_i))$, and δ as before.
– The resulting algorithm will be called "Logistic Regression"

If the function is differentiable...

we can use gradient descent to update the weights:
For the sigmoid function σ, the derivative is
$\sigma'(x) = \sigma(x) \times (1 - \sigma(x))$
We use this to change line 15 in the algorithm *LinearLearner* to

$$w_i := w_i + \eta \times \delta \times pval^{\overline{\omega}}(e, Y) \times [1 - pval^{\overline{\omega}}(e, Y)] \times val(e, X_i).$$

where $pval^{\overline{\omega}}(e, Y) = \sigma(\sum_i w_i \times val(e, X_i))$, and δ as before.
– The resulting algorithm will be called "Logistic Regression"
If your maths is good, try computing the derivative
$((val(e, Y) - pval(e, Y))^2)' =$
$2 \times \delta \times pval(e, Y) \times [1 - pval(e, Y)] \times val(e, X_i)$,
where $pval(e, Y) = \sigma(w_i \times val(e, X_i))$, for each weight w_i.
(very similar to our last lecture exercise)

Logistic regression

Algorithm LogisticLearner(X,Y,E,η)
Inputs:
    X: set of input features, $X = \{X_1, \ldots, X_n\}$
    Y: target feature
    E: set of examples from which to learn
    η: - learning rate
Output: parameters $w_0, \ldots, w_n$.
    Local $w_0, \ldots, w_n$ - real numbers
    $pval(e, Y) = \sigma(w_0 + w_1 \times val(e, X_1) + \ldots + w_n \times val(e, X_n))$
initialise $w_0, \ldots, w_n$ randomly
repeat
    for each example e in E **do**
        $\delta := val(e, Y) - pval(e, Y)$
        for each $i \in [0, n]$ **do**
            $w_i := w_i + \eta \times \delta \times pval(e, Y) \times [1 - pval(e, Y)] \times val(e, X_i)$
until termination
return $w_0, \ldots, w_n$

Example

Our old "Reading mail example" (from lecture on Decision trees) can be classified correctly by the following function:

$$Reads = \sigma(-8 + 7Short + 3New + 3Known),$$

where σ is the sigmoid function. A function similar to this can be found with about 3,000 iterations of gradient descent with a learning rate $\eta = 0{,}05$.

Example

Our old "Reading mail example" (from lecture on Decision trees) can be classified correctly by the following function:

$$Reads = \sigma(-8 + 7Short + 3New + 3Known),$$

where σ is the sigmoid function. A function similar to this can be found with about 3,000 iterations of gradient descent with a learning rate $\eta = 0{,}05$.
According to this function, *Reads* is true if and only if *Short* is true and either *New* or *Known* is true.

Running our Lecture1 exercise in Weka

```
Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                                   Class
Variable                            buys
====================================
musicCD                           0.3475
musicMP3                          6.6939
gamesHard                       -52.6445
gamesOnline                      51.9035
Intercept                       -38.6139
```

Running our Lecture1 exercise in Weka

```
Logistic Regression with ridge parameter of 1.0E-8
Coefficients...
                                    Class
Variable                             buys
=====================================
musicCD                            0.3475
musicMP3                           6.6939
gamesHard                        -52.6445
gamesOnline                       51.9035
Intercept                        -38.6139
```

How do you interpret this, in "human" terms?

Accuracy?

```
=== Predictions on test data ===

 inst#     actual  predicted error prediction
     1     1:buys     1:buys       1
     1     1:buys     1:buys       1
     1     1:buys     1:buys       1
     1     1:buys     1:buys       1
     1  2:cancels  2:cancels       1
     1  2:cancels     1:buys   +   1
     1  2:cancels  2:cancels       1
     1  2:cancels  2:cancels       1
     1  2:cancels  2:cancels       1
     1  2:cancels  2:cancels       1
```

Accuracy?

```
Correctly Classified Instances           9

=== Confusion Matrix ===

 a b   <-- classified as
 4 0 | a = buys
 1 5 | b = cancels
```

Bias in linear classifiers and decision trees

- It's easy for a logistic function to represent "at least two of $X_1, \ldots, X_k$ are true":

w_0	w_1	$\cdots$	w_k

Bias in linear classifiers and decision trees

- It's easy for a logistic function to represent "at least two of $X_1, \ldots, X_k$ are true":

w_0	w_1	$\cdots$	w_k
-15	10	$\cdots$	10

  This concept forms a large decision tree.

Bias in linear classifiers and decision trees

- It's easy for a logistic function to represent "at least two of $X_1, \ldots, X_k$ are true":

w_0	w_1	$\cdots$	w_k
-15	10	$\cdots$	10

  This concept forms a large decision tree.
- Consider representing a conditional: "If X_7 then X_2 else X_3":
  - Simple in a decision tree.
  - Complicated (possible?) for a linear separator

Conclusions

- We have learned about linear classifiers
- Next time, we will discuss their limitations and ways to overcome the limitations
- As always – check related Chapters in the Course textbook: §4.6, pp.124-129, §11.4, 459-469

Homework/Lab: Test 4

- Take the small emotion recognition data set (fer10.arff) again, available on Vision (attached to this lecture slides)
- It is not directly suitable for Linear Regression (Weka will not even allow to run it). Why?

Homework/Lab: Test 4

- Take the small emotion recognition data set (fer10.arff) again, available on Vision (attached to this lecture slides)
- It is not directly suitable for Linear Regression (Weka will not even allow to run it). Why?
- Convert your data set to numeric values, like this:

```
@attribute 'musicCD' numeric
@attribute 'musicMP3' numeric
@attribute 'gamesHard' numeric
@attribute 'gamesOnline' numeric
@attribute 'Buys' numeric
```

- Run, in Weka: Linear regression with the settings: weka.classifiers.functions.LinearRegression -S 2 -R 1.0E-8 -num-decimal-places 4

Homework/Lab: Test 4

- Take your small emotion recognition data set again
- Convert your data set to numeric values and NOMINAL class, like this:

```
@attribute 'musicCD' numeric
@attribute 'musicMP3' numeric
@attribute 'gamesHard' numeric
@attribute 'gamesOnline' numeric
@attribute 'Buys' {Buys, Cancels}
```

- Run, in Weka: Logistic Regression with the settings: weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4

Homework/Lab: Test 4

- We use the same test set as for Decision trees
- Compare performance of these two linear classifiers with your results for Decision trees, on the same test set.
- Be ready to answer test questions about your conclusions
- For both Linear regression and Logistic regression, note the weights w_0, w_1, w_2, w_3, w_4 computed by Weka.
- For both cases, write the *pval* formula in the general form (as we did in the lecture) and compare to what Weka gives.