IINC classifier for MS Excel. The principle, method and Program

Jiřina, Marcel
2014

Institute of Computer Science

Academy of Sciences of the Czech Republic

IINC classifier for MS Excel
The principle, method and Program

Marcel Jiřina

Technical Report No. V-1199

October 2014

Abstract

In this report we describe the IINC from function point of view without any theory. The principle of the IINC classifier is illustrated. The classifier is built as a macro of the MS Excel that processes data at the separate user's sheet according to parameters written by user at the control sheet. Then there is described how macro in the MS Excel works and how the classifier in MS Excel is used. User can choose among three metrics, L1, L2, and Hassanat's metric. There is also discussed the problem of old and new (till 2003, and after 2003) Excel.

Keywords:

Multivariate data, classification, Excel, Hassanat's metric.

IINC classifier for MS Excel
The principle, method and Program

Marcel Jiřina

Contents

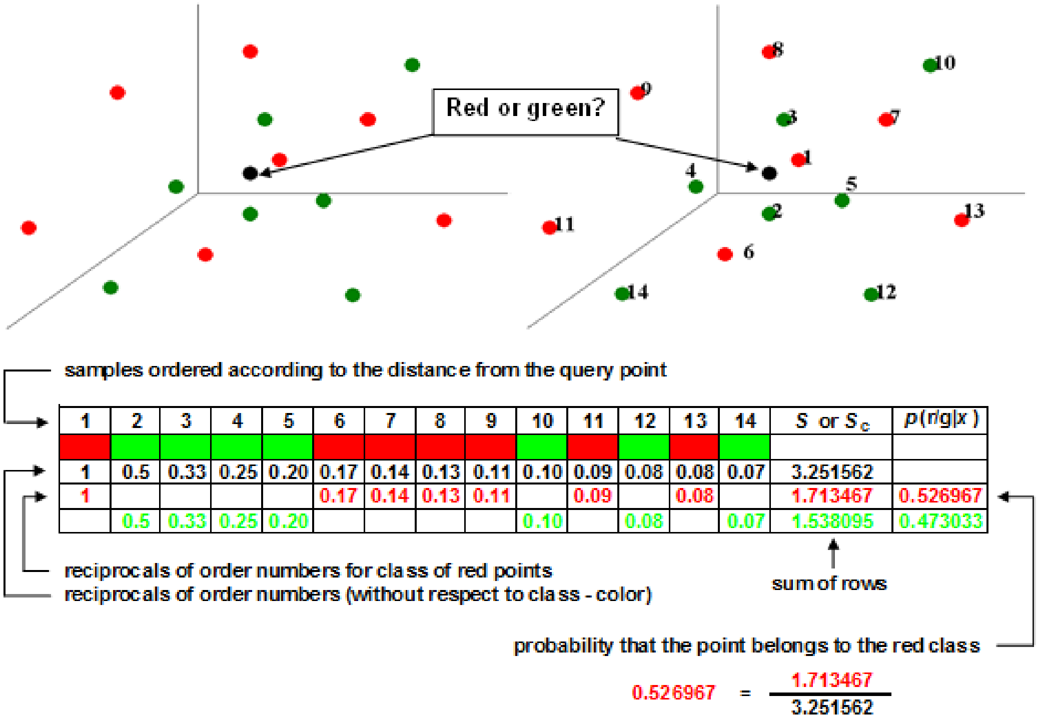
Introduction

The IINC (Inverted Indexes of Neighbors Classifier) is easy to use and also easy to program classifier. This work is motivated by the fact, that ready-to-use device would be useful in spite of simplicity mentioned.

In this report we describe the IINC from function point of view without any theory. The theoretical background is given in papers and reports, see references. Then we describe how macro in the MS Excel works and how the classifier in MS Excel is used. In this part also problem of old and new (till 2003, and after 2003) Excel is discussed.

IINC classifier

The classification procedure is depicted in Fig. 1. The problem is: What is color of given point *x* depicted in black at the left upper part of picture? First we rank points of the learning set according to their distances from point *x* as shown at the right upper part of picture. There are 14 points here, 7 red, 7 green as shown in the upper lines in the table below pictures. Reciprocals of rank numbers are in the third line. In the fourth and fifth line there are reciprocals of ranks of points x_i from sets $U_{c=red}$ and $U_{c=green}$. In the rightmost two columns of table are corresponding sums and estimated probabilities that point *x* is red (0.526967) or green (0.473033). Setting threshold $\theta = 0.5$ we can state that point *x* is red.



samples ordered according to the distance from the query point

1	2	3	4	5	6	7	8	9	10	11	12	13	14	S or S_c	$p(r/g\|x)$
🟥	🟩	🟩	🟩	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟩	🟥	🟩		
1	0.5	0.33	0.25	0.20	0.17	0.14	0.13	0.11	0.10	0.09	0.08	0.08	0.07	3.251562	
1					0.17	0.14	0.13	0.11		0.09		0.08		1.713467	0.526967
	0.5	0.33	0.25	0.20					0.10		0.08		0.07	1.538095	0.473033

reciprocals of order numbers for class of red points

reciprocals of order numbers (without respect to class - color)

sum of rows

probability that the point belongs to the red class

$$0.526967 = \frac{1.713467}{3.251562}$$

Figure 1. Illustration of classification procedure for the simplest case of two classes and of the same number of samples of both classes.

For *C* classes and different numbers of samples N_c in individual classes, $c = 1, 2, \dots C$ there is

$$\hat{p}(c \mid x) = \frac{\frac{1}{N_c} \sum_{x_i \in U_c} 1/i}{\sum_{k=1}^{C} \frac{1}{N_k} \sum_{x_i \in U_k} 1/i} ,$$

where $x_i \in U_c$ denotes that the sum goes over indexes *i* for which the corresponding samples of the learning set are of class *c*,

Macro in the MS Excel and its use

Here we describe how macro in the MS Excel works and how the classifier in MS Excel is used.

Excel 2003 - Excel 2013

The problem of old and new (till 2003, and after 2003) Excel nearly does not exist. Workbook IINC.xls can be read in the new Office and then saved as IINC.xlsm (macros enabled). Vice versa, IINC.xlsm can be opened in the "old" Excel with compatibility pack installed and then saved as IINC.xls. Without these conversions (that are not necessary as in the IINC pack there are both versions) IINC may run extremely slow.

The use of the IINC-Excel for classification

Instructions for use are shortly described in Table 1 and in sheet "Help" in the workbook.

Two workbooks are supposed, the IINC.xls or IINC.xlsm and workbook with your data. In the workbook with your data must be added sheets named "wrk" and "norm".

You have to say to IINC.xls/.xlsm workbook how data in your workbook are organized. So, you have to fill-in some data at the sheet IINC in workbook IINC.xls/.xlsm. These data are:

- Names of sheets with data, eventually the name of a single sheet with data.
- The first and the last line with your training and testing set. This allows you to have all data at one sheet, say in rows 1 till 150 the learning set and lines No. 200 till 299 the testing set.
- The column where the class label is given. Note that class labels must be 1, 2, 3 and so on without omitting any number. Thus for two-class problem labels are 1 and 2. For four class problem 1, 2, 3, 4, but not e.g. 1, 2, 5, 6.
- The list of classes under the colored part on the sheet IINC.
- The first and the last column of data. Also this allows you to have all data at one sheet, say in columns A till K (1-11) the learning set with class label in column K (i.e. 11), and in columns M (13) till V (22) the testing set.
- The column where results (classes found by classifier) are written. Results are written at the same place (to the same rows) where there is the testing set, and you can specify to what column results should be written. In this column the class found by classifier is written and in next columns are written estimated probabilities of individual classes

to which a sample may belong. So, count with space for (1+No.of classes) columns starting with column for classes found by classifier. Note that resulting class and probabilities are written to the sheet without asking and will overwrite any data there.

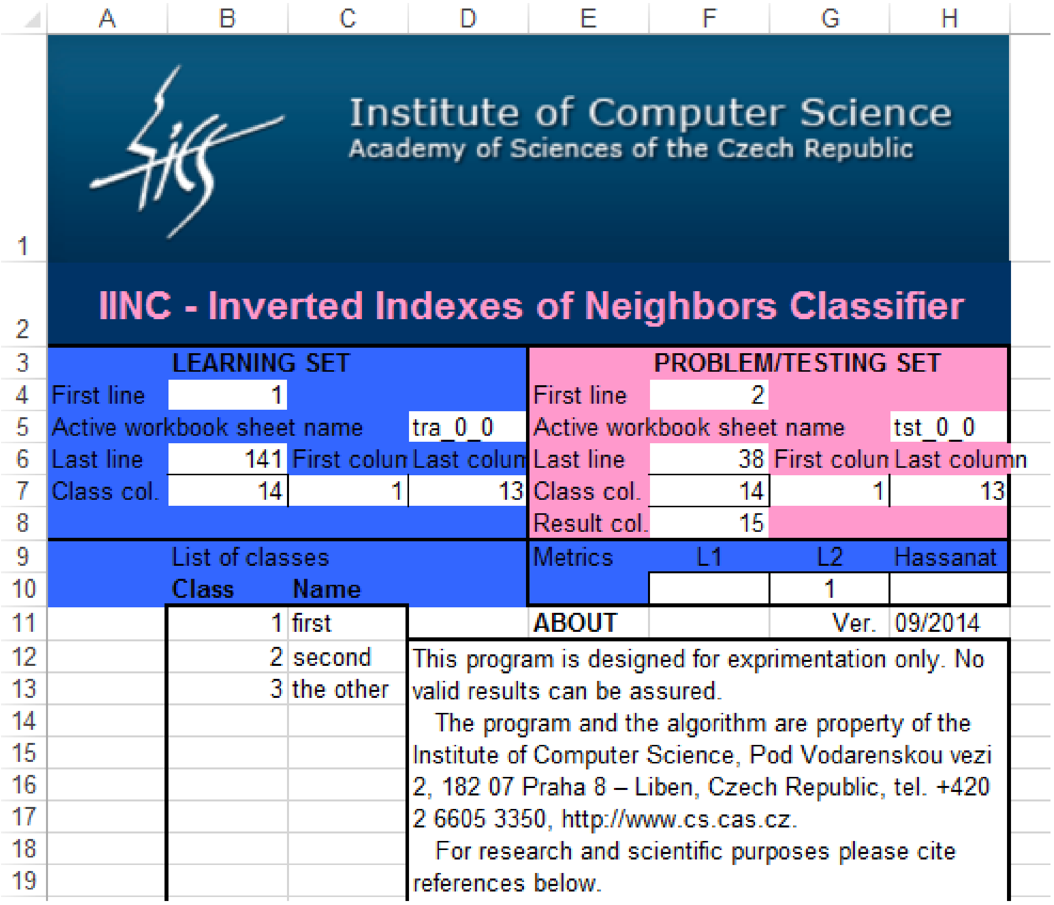
- You can choose a metric for distances computation, L1, L2, and Hassanat's metrics [6]. We found Hassanat's metrics usually the best, L1 the second best, and L2 is mostly used in distance-based classifiers. We found that differences among these three metrics are often rather small.

To start computation make the workbook with your data active (with active any sheet) and then hit Ctrl-d (we suppose that this short-key is not used in your workbook otherwise than as default, i.e. fill-in down).

When computation ends, you will see message box "Ready.". You can break computation using Ctrl-Pause/Break keys. When repeating the computation you need not clear sheets "wrk", "norm" or what was already written in the sheet with your testing data.

Step	To do
1	The Excel workbook with data (i.e. a different workbook than IINC.xls or IINC xlsm but in the same directory) must have a sheet named "wrk" and a sheet named "norm". The contents of these sheets will be overwritten.
2	All data may be at one sheet. (The learning set as well as the analysis/testing set.)
3	Fill-in form on sheet IINC including list of class labels. All entries must be used. A single exception is metrics used, there must be written "1" in one corresponding cell only.
4	Warning! Results are written to the sheet or part of sheet corresponding to the analysis/testing set. In the column denoted "Results col." the resulting class number is written. Columns next to right show probabilities of individual classes. Rows are the same as rows of the samples related to.
5	Class labels should be integers in the ascending order and with no deleted numbers. Class names are arbitrary strings for your convenience.
6	Switch the workbook with your data to be active.
7	Hit Ctrl-d to start computation. Computation ends with the messagebox "Ready."
8	Look for results on the sheet with your analysis/testing data.

Table 1. Short instructions for use of the IINC Excel classifier.



	A	B	C	D	E	F	G	H
1					Institute of Computer Science Academy of Sciences of the Czech Republic			
2	IINC - Inverted Indexes of Neighbors Classifier							
3		LEARNING SET				PROBLEM/TESTING SET		
4	First line	1			First line	2		
5	Active workbook sheet name			tra_0_0	Active workbook sheet name			tst_0_0
6	Last line	141	First colun	Last colun	Last line	38	First colun	Last column
7	Class col.	14	1	13	Class col.	14	1	13
8					Result col.	15		
9		List of classes			Metrics	L1	L2	Hassanat
10		Class	Name				1	
11		1	first		ABOUT		Ver.	09/2014
12		2	second	This program is designed for exprimentation only. No valid results can be assured. The program and the algorithm are property of the Institute of Computer Science, Pod Vodarenskou vezi 2, 182 07 Praha 8 – Liben, Czech Republic, tel. +420 2 6605 3350, http://www.cs.cas.cz. For research and scientific purposes please cite references below.				
13		3	the other					
14								
15								
16								
17								
18								
19								

Figure 2. An example of sheet IINC.

Legal Statement

This program is designed for experimentation only. No valid results can be assured.

The program and the algorithm are property of the Institute of Computer Science, Pod Vodarenskou vezi 2, 182 07 Praha 8 – Liben, Czech Republic, tel. +420 2 6605 3350, http://www.cs.cas.cz.

For research and scientific purposes please cite references below.

For other purposes the algorithm that is a core of the program is the subject of the patent pending under number PV 2008-245; Z 7576 submitted on 22 April 2008 to the INDUSTRIAL PROPERTY OFFICE, Antonína Čermáka 2a, 160 68 Prague, Czech Republic. The owner of the invention is the Institute of Computer Science above.

Acknowledgement

The work was supported by Ministry of Education of the Czech Republic under INGO project No. LG 12020.

References

[1] M. Jiřina and M. Jiřina, jr.: Classification Using Zipfian Kernel. Journal of Classification (Springer), 2014 (in print).

[2] M. Jiřina, M. Jiřina, jr.: Classifier based on Inverted Indexes of Neighbors. Technical Report No. V-1034, Institute of Computer Science AS CR, 11 pp., November 2008.

[3] M. Jiřina, M. Jiřina, jr.: Classifier based on Inverted Indexes of Neighbors II. - Theory and Appendix. Technical Report No. V-1041, Institute of Computer Science AS CR, , 26 pp., November 2008.

[4] M. Jirina and M. J. Jirina, "Using Singularity Exponent in Distance Based Classifier," in Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA2010), Cairo, 2010, pp. 220-224.

[5] M. Jirina and M. J. Jirina, "Classifiers Based on Inverted Distances," in New Fundamental Technologies in Data Mining, K. Funatsu, Ed. InTech, 2011, vol. 1, Ch. 19, pp. 369-387.

[6] A. B. Hassanat: Dimensionality Invariant Similarity Measure. Journal of American Science 2014; Vol. 10, No. 8, pp. 221-226.

Appendix

Macro for MS Excel

This macro works without any changes for MS Excel till 2003, as well as for newer MS Excel using .xlsm sheet (with macros enabled).

```
Sub IINC()
'
' IINC Macro
' Macro first recorded 24.2.2010, UM400
'
' Short Key: Ctrl+d
'
    Dim S(100) As Double
    Dim C(100) As Double 'counts

    'take off info from the IINC sheet:
    lrnName = ThisWorkbook.Sheets("IINC").Cells(5, 4).Value
    lrnFrow = ThisWorkbook.Sheets("IINC").Cells(4, 2).Value
    lrnLrow = ThisWorkbook.Sheets("IINC").Cells(6, 2).Value
    lrnQcol = ThisWorkbook.Sheets("IINC").Cells(7, 2).Value
    lrnLcol = ThisWorkbook.Sheets("IINC").Cells(7, 3).Value
    lrnRcol = ThisWorkbook.Sheets("IINC").Cells(7, 4).Value
    tstName = ThisWorkbook.Sheets("IINC").Cells(5, 8).Value
    tstFrow = ThisWorkbook.Sheets("IINC").Cells(4, 6).Value
    tstLrow = ThisWorkbook.Sheets("IINC").Cells(6, 6).Value
    tstQcol = ThisWorkbook.Sheets("IINC").Cells(7, 6).Value
    tstLcol = ThisWorkbook.Sheets("IINC").Cells(7, 7).Value
    tstRcol = ThisWorkbook.Sheets("IINC").Cells(7, 8).Value
    tstResu = ThisWorkbook.Sheets("IINC").Cells(8, 6).Value
    MetrL1 = ThisWorkbook.Sheets("IINC").Cells(10, 6).Value
    MetrL2 = ThisWorkbook.Sheets("IINC").Cells(10, 7).Value
    MetrHa = ThisWorkbook.Sheets("IINC").Cells(10, 8).Value
    'count classes
    For i = 9 To 108
        If Trim(ThisWorkbook.Sheets("IINC").Cells(i, 2)) = "" Then GoTo a0
    Next
a0:
    classes = i - 9
    'compute means of all variables
    'sesit = ActiveWorkbook.Name
    ActiveWorkbook.Sheets(lrnName).Activate
    For v = lrnLcol To lrnRcol
        If v = lrnQcol Then GoTo a
        meann = EstMean(lrnName, v, lrnFrow, lrnLrow)
        ActiveWorkbook.Sheets("wrk").Cells(1, v).Value = meann
        ActiveWorkbook.Sheets("wrk").Cells(2,  v).Value  =  sqrt(EstVari(lrnName,  v,
lrnFrow, lrnLrow, meann))
        ActiveWorkbook.Sheets(lrnName).Cells(1, v).Activate
a:
    Next

    'normalize to norm sheet
    For v = lrnLcol To lrnRcol
        If v = lrnQcol Then GoTo b
        For r = lrnFrow To lrnLrow
            x = ActiveWorkbook.Sheets(lrnName).Cells(r, v).Value
            x   =   (x   -   ActiveWorkbook.Sheets("wrk").Cells(1,   v).Value)   /
ActiveWorkbook.Sheets("wrk").Cells(2, v).Value
            ActiveWorkbook.Sheets("norm").Cells(r, v).Value = x
        Next
        ActiveWorkbook.Sheets(lrnName).Cells(1, v).Activate

b:
    Next

    'take a test sample
    'Compute distances for IINC:
    ActiveWorkbook.Sheets(tstName).Activate
    For t = tstFrow To tstLrow
        'ActiveWorkbook.Sheets(tstName).Cells(t, 1).Activate
        'normalize the sample as the third row on wrk sheet
        For v = lrnLcol To lrnRcol
            If v = lrnQcol Then GoTo C
            x = ActiveWorkbook.Sheets(tstName).Cells(t, v).Value
            x   =   (x   -   ActiveWorkbook.Sheets("wrk").Cells(1,   v).Value)   /
ActiveWorkbook.Sheets("wrk").Cells(2, v).Value
            ActiveWorkbook.Sheets("wrk").Cells(3, v).Value = x
C:
        Next
        'compute distances from all normalized samples of the learning set
        For r = lrnFrow To lrnLrow
            dist2 = 0
            For v = lrnLcol To lrnRcol
                If v = lrnQcol Then GoTo d
                If MetrL1 = 1 Then
                    dist2  =  dist2  +  Abs(ActiveWorkbook.Sheets("norm").Cells(r,
v).Value - ActiveWorkbook.Sheets("wrk").Cells(3, v).Value)
                End If
                If MetrHa = 1 Then
                    ai = ActiveWorkbook.Sheets("norm").Cells(r, v).Value
                    bi = ActiveWorkbook.Sheets("wrk").Cells(3, v).Value
                    minai = ai
                    maxai = ai
                    If minai > bi Then minai = bi
                    If maxai < bi Then maxai = bi
                    If minai >= 0 Then
                        dist2 = dist2 + (1 - (1 + minai) / (1 + maxai))
                    Else
                        dist2 = dist2 + (1 - (1 + minai + Abs(minai)) / (1 + maxai
+ Abs(minai)))
                    End If
                End If
                If MetrL2 = 1 Then
                    dist2   =   dist2   +   Na2(ActiveWorkbook.Sheets("norm").Cells(r,
v).Value - ActiveWorkbook.Sheets("wrk").Cells(3, v).Value)
                End If
            Next
            If MetrL2 = 1 Then dist2 = sqrt(dist2)
            ActiveWorkbook.Sheets("wrk").Cells(3 + r, 1).Value = dist2
            'add corresponding class
            ActiveWorkbook.Sheets("wrk").Cells(3      +      r,      2).Value      =
ActiveWorkbook.Sheets(lrnName).Cells(r, lrnQcol).Value
d:
        Next
        'sort distances
        Sheets("wrk").Activate
        Range(Cells(4, 1), Cells(lrnLrow - lrnFrow + 4, 2)).Select
        Selection.Sort Key1:=Range("A4"), Order1:=xlAscending, Order2:=xlAscending,
Header:=xlGuess, OrderCustom:=1, _
            MatchCase:=False, Orientation:=xlTopToBottom
        'MsgBox "after sort"
        'compute sums of reciprocals
        For clas = 1 To classes 'set nulls
            S(clas) = 0
            C(clas) = 1 ''0
        Next
        For r = 1 To lrnLrow - lrnFrow + 1
            clas = ActiveWorkbook.Sheets("wrk").Cells(r + 3, 2).Value
            S(clas) = S(clas) + 1 / r 'reciprocals according to class
            ''C(clas) = C(clas) + 1 'counting samples of individual classes
        Next
        'compute probabilities, select best, and write them to tstName sheet
        Sums = 0
        maxS = 0
        For clas = 1 To classes
            ss = S(clas) / C(clas) 'recomputing to one sample of the learning set
            Sums = Sums + ss
            If maxS < ss Then
                maxS = ss
                maxClas = clas
            End If
        Next
        'best class:
        ActiveWorkbook.Sheets(tstName).Cells(t, tstResu).Value = maxClas
        For clas = 1 To classes 'probabilities
            ActiveWorkbook.Sheets(tstName).Cells(t, tstResu + clas).Value = S(clas)
/ (C(clas) * Sums)
        Next
    Next
    'ActiveWorkbook.Sheets(tstName).Cells(1, tstResu).Activate
    MsgBox ("Ready.")
End Sub
```

***