CASE STUDY DATA-DRIVEN MODELS

# Use of weights of evidence modeling for landslide susceptibility mapping

Version: January 2002

By:
**C.J. van Westen**
**International Institute for Geoinformation Science and Earth Observation (ITC),**
**P.O. Box 6, 7500 AA Enschede, The Netherlands**
**Tel: +31 53 4874263, Fax: +31 53 4874336, e-mail: WESTEN@ITC.NL**

## Summary

This case study deals with methods for for data-driven modeling in a GIS. Several approaches to analyze multiple maps are introduced by means of a number of basic exercises that follow closely the chapter "Tools for map analysis: multiple maps" from the book Geographic Information Systems for Geoscientists, by Graeme Bonham-Carter (1994). The various methods, in the book illustrated with a hypothetical landfill site selection problem and mineral potential mapping, will be treated in our case study with a real world data set from the Chinchina Area, Central Colombia, South America A landslide susceptibility assessment problem is used as an example. The same method can be applied for a wide range of other spatial prediction applications. In this exercise you will learn how to calculate prior probability, conditional probability and the positive and negative weights used in the weights of evidence modeling. Weights are combined for a number of factor maps, and a total weight map will be generated. This map is classified using the "success rate". The predicting power is checked by analyzing the "prediction rate". Also emphasis is given to the automatic calculation of weights using ILWIS script files.

## Getting started

The data for this case study are stored on the EREGdata computer in the directory **\Ereg 2001-2002\exercises\module 6\data_driven models**

☞

- Copy the data for the exercise to a user directory on your hard disk first!

# 1 Problem definition: landslide susceptibility mapping

In the methods we have used so far, we were interested in finding the optimal location of a landfill site. We used subjective weighting, based on expert opinion, so called "knowledge-driven" modelling.

In this exercise we will assign weights by statistical procedures. The weights of different classes are determined by calculating densities. The method is therefore "data-driven".

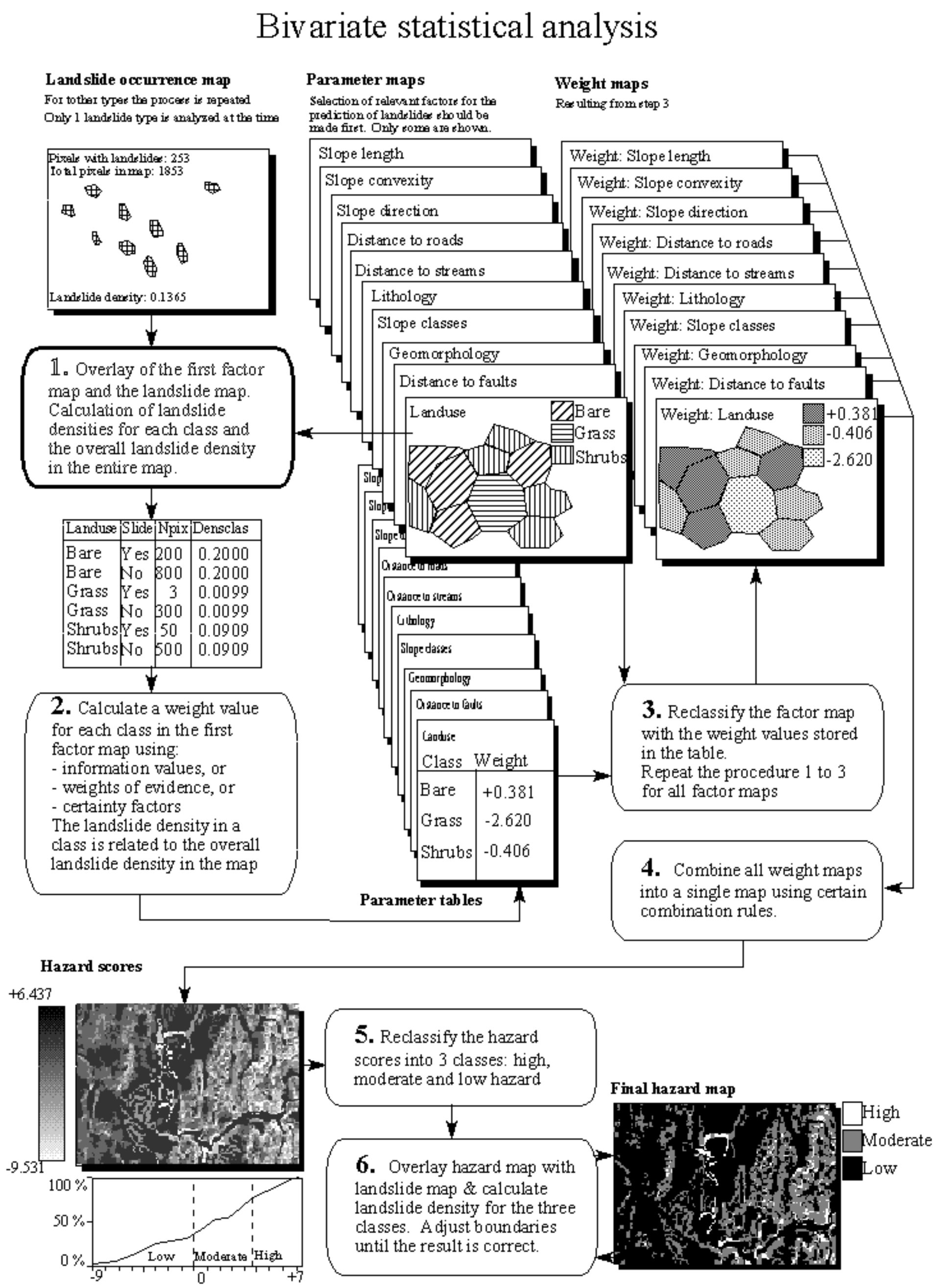
The landfill study case is not useful to demonstrate these "data-driven" techniques, since we cannot evaluate the best site for a landfill by evaluating the conditions at existing landfills (there is only one in the area).

Therefore we will use another problem for the statistical analysis: the prediction of areas that are susceptible for the occurrence of landslides. In the statistical modeling of landslides we use the rule "the present and past are the key to the future". This means we can look at the conditions under which landslides have occurred in the past, and use the critical combinations for predicting the possible occurrence of landslides where the same conditions prevail, but which are still landslide free.

A flowchart of the method is presented in figure 1.

Table 1: Available data

| File | Description | Filetype | Domain | Attribute table |
|---|---|---|---|---|
| `Slide` | Evidence map: landslide map | Raster | ID: Slide | Slide |
| `Slide60` | Landslide map from the 1960 | Raster | ID: Slide60 | Slide60 |
| `Slide80` | Landslide map from the 1980 | Raster | ID: Slide80 | Slide80 |
| `Slope` | Slope classes (10 degreees) | Raster | Class:slope | No |
| `Geom` | Simplified Geomorphology | Raster | Class:geom | No |
| `Geol` | Lithology | Raster | Class:geol | No |
| `Landuse` | Landuse classes | Raster | Class:landuse | No |
| `Aspcl` | Slope direction classes | Raster | Class: aspcl | No |
| `Contour` | Contour lines | segments | Value | no |
| `Drain` | Drainage network | segments | Class:drain | No |
| `Fault` | Geological faults & lineaments | segments | Class:fault | No |



For tother types the process is repeated
Only 1 landslide type is analyzed at the time

Selection of relevant factors for the prediction of landslides should be made first. Only some are shown.

Resulting from step 3

1. Overlay of the first factor map and the landslide map. Calculation of landslide densities for each class and the overall landslide density in the entire map.

| Landuse | Slide | Npix | Densclas |
|---|---|---|---|
| Bare | Yes | 200 | 0.2000 |
| Bare | No | 800 | 0.2000 |
| Grass | Yes | 3 | 0.0099 |
| Grass | No | 300 | 0.0099 |
| Shrubs | Yes | 50 | 0.0909 |
| Shrubs | No | 500 | 0.0909 |

2. Calculate a weight value for each class in the first factor map using:
- information values, or
- weights of evidence, or
- certainty factors

The landslide density in a class is related to the overall landslide density in the map

Landuse

| Class | Weight |
|---|---|
| Bare | +0.381 |
| Grass | -2.620 |
| Shrubs | -0.406 |

3. Reclassify the factor map with the weight values stored in the table. Repeat the procedure 1 to 3 for all factor maps

4. Combine all weight maps into a single map using certain combination rules.

5. Reclassify the hazard scores into 3 classes: high, moderate and low hazard

6. Overlay hazard map with landslide map & calculate landslide density for the three classes. Adjust boundaries until the result is correct.

Figure 1: Simplified flowchart for bivariate statistical analysis.

# 2 Checking the input data

The input data (table1) consists of a map containing information on the features you would like to predict (in this case landslides), which is called an evidence map. And the data set contains a number of factor maps that may control the occurrence of the feature you would like to predict.

☞

- Display the map `Slide`.
- Move through the map and press the left mouse button for information on the various units. As you can see the area outside of the landslides reveals a ? (undefined) when you press the left mouse button. These areas are called *undefined*. This means that no information is stored for the non-landslide areas. The landslides themselves all have a unique code.
- Move your mouse pointer to one of the landslides and double click on it. Now the information from the table connected to the map `Slide` is displayed.

The map `Slide` has a so-called identifier domain. This means that each unit (landslide) from this map has a unique code.

When you move the mouse pointer to one of the landslides, you will see that the attribute table contains information for `Type`, `Subtype`, `Activity`, `Depth`, `Vegetation` and `Scarp`.

☞

- Each time you double click on a part of the map, the information from the table for that unit will be displayed. Try this out for several different units. Close the Edit Attribute window.
- Open the pixel information window and drag-and-drop the map `Slide` into it. Now if you move with the mouse pointer the information is shown without the need to double-click.
- To see what the table looks like, go to the main ILWIS window and open the table `Slide` by double-clicking it. Have a look at the different columns. If you double-click the name of a column you get information on the column type.
- Close the table window.

The columns Type, Subtype, Activity, Depth, Vegetation, Scarp are class domain columns. These names are defined in the domain files. The various domain items of these columns are shown in table 2.

Table.2: List of domain items for mass movement characteristics

| | **Type** | **Subtype** | **Activity** | **Depth** | **Vegetation** | **Scarp** |
|---|---|---|---|---|---|---|
| **0** | Unknown | Unknown | Unknown | Unknown | Unknown | Unknown |
| **1** | Slide | Rotational | Stable | Shallow | Bare | Scarp |
| **2** | Flowslide | Translational | Dormant | Deep | Low vegetation | Body |
| **3** | Flow | Complex | Active | | High vegetation | |
| **4** | Debris avalanche | | | | | |
| **5** | Creep | | | | | |

☞

- Open the domain Activity by double-clicking it. As you can see each class has a *code*, which corresponds to the values in the left column of table 1. Each class domain also contains a representation, in which the colors for each class are defined.
- Open the representation Activity and have a look at the content. After that close the representation and the domain.

You can also display the map Slide with an attribute from its table.

☞

- Make the landslide map Slide active.Press the right mouse button while in the map, and select: 1.map Slide. In the following Display Options dialog box click on Attribute and select the column Activity. Press OK. Now the map is redisplayed, with the colors from the representation Activity. If you click on a landslide you will see the activity information displayed.
- Also try this with some other columns (Type, Subtype, Activity, Depth, Vegetation and Scarp).
- Close the map.

Along side the landslide map you also have five factor maps that might be controlling the locations of landslides: Geol (geological units) , Slope (slope classes) , `Landuse`, `Aspcl` (slope direction),and `Geom` (Geomorphological units). These maps have a class domain.

☞

- Open the map `Geol` and consult the information from the map and the accompanying table.
- Add the maps `Slide`, `Geol`, `Slope`, `landuse`, `aspcl` and `Geom` to the pixel information window. When you move through the map you can simultaneously read the information from all three maps and their tables.
- Close the map windows and the pixel information window.

So far you have only been looking at the content of the maps. You will now start with the actual analysis.

## 3 Creating a landslide distribution map

Previously you displayed the activities of the landslides in the study area. However, you did not actually make a new map showing these activities. This is what you will do now, by renumbering the map `Slide` with the attribute `Activity`.

☞

- Create a raster map `Activit` by reclassifying the map `Slide` with the attribute `Activity`.
- To remove the undefined values from the map `Activit` type the following formula in the command line:

```
   Activity=iff(isundef(Activit),"unknown",
Activit)↵
```

- Calculate how many landslides are present in the area, with the different activity classes (active, dormant, stabil), and what is their area. Include this in the report for this exercise.

## 4 Calculating prior probability

Consider that you would only have the landslide map of the area, without any additional data. How to answer the question: what is the chance of having a landslide in a particular part of the area (in a randomly selected pixel)?

If there is no additional information, i.e. if you do not know whether you are in a geological unit with many landslide, or on a steep slope, we speak of the prior probability.

The prior probability that a certain pixel in the map will have a landslide is simply the density of landslides in the entire area. In other words, the number of pixels with landslides, divided by the total number of pixels in the map.

$$P_{Prior} = P\{\ S\ \} = \frac{Npix(\ Slide\ )}{Npix(\ Total\ )}$$

In a formula that is:

where:

| | |
|---|---|
| Pprior = P{S} | Conditional probability of having a landslide S |
| Npix(Slide) | Number of pixels with landslides in the map |
| Npix(Total) | Total number of pixels in the map |

☞

- Calculate the prior probability for all landslides in the study area
- Calculate the prior probability of the active landslides in the study area
- Calculate the prior probability for all active landslides with type 4 (debris avalanches)

## 5 Calculating conditional probability

If we have more information available than just the landslide map, we can see what will be the probability that we may have a landslide, given that we are in a certain geomorphological unit, for example. If we are in the geomorphological unit "Romeral fault zone" the probability that there may be a landslide is obviously much larger, then when we are on flat river terraces. This is what we call conditional probability. Considering the relationship between a binary variable map (B) and a landslide map: (S) The probability given a certain condition. It can be expressed as:

$$P\{\ S\,|\,B\ \} = \frac{P\{\ S \cap B\ \}}{P\{\ B\ \}} = \frac{Npix\{\ S \cap B\ \}}{Npix\{\ B\ \}}$$

**P{S | B}** the conditional probability of having a landslide while you are in unit B.

In other words, the conditional probability of having landslides, given that you are in a certain unit, is the density of landslides within that unit, calculated as the number of pixels with landslides in the unit, divided by the total number of pixels in the unit.

☞

- Create a bit map showing only the active landslides: `active.` Make sure that the map doesn't have any undefined values, but only values 1 (active landslides) and 0 (rest of the area)
- Cross the geomorphological map `geom` with the map `active` and calculate the conditional probability for each of the classes.
- Evaluate which classes have a higher value for the conditional probability than the prior probability calculated in the previous exercise.

# 6 Calculating positive and negative weights

In this study the weights of evidence method (Bonham-Carter, 1994) was selected for the indirect landslide susceptibility assessment. In this method positive and negative weights (Wi+ and Wi- ) are assigned to each pixel of the factor maps (e.g. each lithological unit within a lithology map), which are defined as:

$$W_i^{+} = \log_e \frac{P\{B_i|S\}}{P\{B_i|\overline{S}\}} \qquad \text{[eq. 1]}$$

and

$$W_i^{-} = \log_e \frac{P\{\overline{B_i}|S\}}{P\{\overline{B_i}|\overline{S}\}} \qquad \text{[eq. 2]}$$

where,

Bi = presence of a potential landslide conditioning factor,

Bi = absence of a potential landslide conditioning factor,

S = presence of a landslide, and

S = absence of a landslide

The method can be performed using individual factor maps, which only contain two classes, representing the presence or absence of the factor. Since this requires the analysis of many individual maps (e.g. each lithological unit separately) it is more convenient to work with multi-class maps, containing many factors (e.g. all lithological units together in one map).

For each factor, Wi+ is used for those pixels of a factor (represented as a class in a multi-class map) to indicate the importance of the presence of the factor for the occurrence of landslides. If Wi+ is positive the presence of the factor is favourable for the occurrence of landslides, and if Wi+ is negative it is not favourable.

Wi- is used to evaluate the importance of the absence of the factor for the occurrence of landslides. When Wi- is positive the absence of the factor is favourable for the occurrence of landslides, and when it is not. Weights with extreme values indicate that the factor is a useful for the susceptibility mapping, while factors with a weight around zero have no relation with the occurrence of landslide.

For each factor there are four possible combinations, of which the frequency, expressed as number of pixels, can be calculated with a GIS (see table 2).

| | | **$B_i$ : Potential landslide conditioning factor** | |
|---|---|---|---|
| | | (Present) | (Absent) |
| **S: Landslides** | Present | $Npix_1$ | $Npix_2$ |
| | Absent | $Npix_3$ | $Npix_4$ |

*Table 2: Four possible combinations of a potential landslide conditioning factor and a landslide inventory map. Npix = number of pixels.*

Based on equations [1] and [2] the weights of evidence can be written in numbers of pixels as follows:

$$W_i^+ = \log_e \frac{\dfrac{\mathrm{Npix}_1}{\mathrm{Npix}_1 + \mathrm{Npix}_2}}{\dfrac{\mathrm{Npix}_3}{\mathrm{Npix}_3 + Npix_4}} \qquad \text{[eq. 3]}$$

and

$$W_i^- = \log_e \frac{\dfrac{\mathrm{Npix}_2}{\mathrm{Npix}_1 + \mathrm{Npix}_2}}{\dfrac{\mathrm{Npix}_4}{\mathrm{Npix}_3 + Npix_4}}$$

[eq. 4]

In order to calculate the weights using equations 3 and 4 you need to create the following columns:

nmap = total number of pixels in the map  
nslide = number of pixels with landslides in the map  
nclass = number of pixels in the class  
nslclass = number of pixels with landslides in the class  
The values needed for the weight formulas are:  
npix1 = nslclass  
npix2 = nslide-nslclass  
npix3 = nclass-nslclass  
npix4 = nmap-nslide-nclass+nslclass

The weights are calculated as:

W+ =ln((npix1/(npix1+npix2))/(npix3/npix3+npix4))  
=lng((npix1*(npix3+npix4)/((npix1+npix2)*npix3)

W- =ln((npix2/(npix1+npix2))/(npix4/npix3+npix4))  
=ln((npix2*(npix3+npix4)/((npix1+npix2)*npix4)

☞

- Calculate by hand the various steps and write the results in the table on the next page.

| Slope | Active | Npix | nmap | nslide | nclass | nslclass | Npix 1 | Npix2 | Npix3 | Npix4 | Wplus | Wmin |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 2000 | | | | | | | | | | |
| 1 | 1 | 10 | | | | | | | | | | |
| 2 | 0 | 1500 | | | | | | | | | | |
| 2 | 1 | 20 | | | | | | | | | | |
| 3 | 0 | 1200 | | | | | | | | | | |
| 3 | 1 | 50 | | | | | | | | | | |
| 4 | 0 | 1000 | | | | | | | | | | |
| 4 | 1 | 100 | | | | | | | | | | |
| 5 | 0 | 900 | | | | | | | | | | |
| 5 | 1 | 90 | | | | | | | | | | |
| 6 | 0 | 800 | | | | | | | | | | |
| 6 | 1 | 60 | | | | | | | | | | |
| 7 | 0 | 700 | | | | | | | | | | |
| 7 | 1 | 50 | | | | | | | | | | |
| 8 | 0 | 100 | | | | | | | | | | |
| 8 | 1 | 10 | | | | | | | | | | |
| 9 | 0 | 50 | | | | | | | | | | |
| 9 | 1 | 3 | | | | | | | | | | |

Let us now calculate with ILWIS the weights for the map `geom`. We will only use the landslides that are classified as *active*, since some of the old landslide have occurred in other climatic conditions. If they would also be used, the results would be difficult to interpret.

☞

- Cross the map `geom` with the map `active` indicating only the active slides. Save the cross table as `geomact`
- Carry out the same calculation steps as you did in the manual example, but now as a series of Table calculation statements and aggregation function within the crosstable `geomact`. Calculate the weights for the classes of the map `geom`. Make sure to write down all command that you perform (the actual commands on the command line)
- Repeat the procedure but now do the calculation not in the crosstable, but in the attribute table linked to the domain `geom`. You will have to use aggregation and join functions for that. Write down the command line statements.
- Evaluate the results.

# 7. Calculating final weights and contrast factors

Since all of the maps are multi-class maps, containing several factors (classes), the presence of one factor (e.g. one specific geomorphological unit) implies the absence of the other factors of the same map (e.g. geomorphological map). Therefore in order to obtain the total weight of each factor, the positive weight of the factor itself should be added to the negative weight of the other factors in the same map.

This is done by first adding up all negative weights of the classes of one map, in a column using an aggregation function (SUM).

The final weight is then calculated as:

**Wmap = Wplus + Wmintotal – Wmin**

In which Wmintotal = the total of all negative weights in a multiclass map.

☞

- Calculate t he final weights and store them in the attribute table `geom`.

To quantify the spatial association between a map class and the occurrence of landslides, the contrast factor as mentioned in Bonham-Carter (1994) is defined:

$$C_w = W^+ - W^-$$

The contrast factor is 0, when the landslide pattern and map class pattern overlap only by the expected amount due to chance, positive when there is a positive association between the two patterns and negative when there is a negative association between the two patterns.

☞

- Calculate the contrast factors and store them in the attribute table `geom`.

# 8. Use of scripts

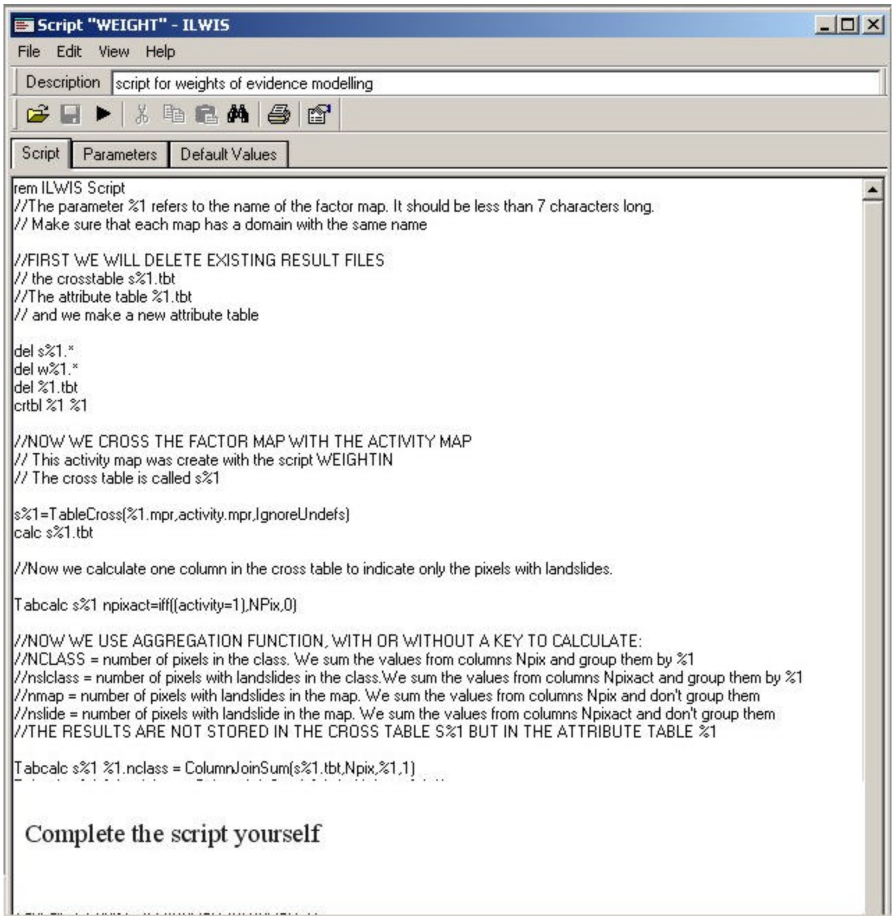
Because the same calculation has to be performed for the other maps, it would be useful to automate the procedure. This can be done with a SCRIPT, i.e. a text file in which the exact procedure followed in the program and the formulas are stored.

You can find all operations that you have done with ILWIS in the so-called ILWIS.LOG file. The commands from this file can be copied to a script file in ILWIS.

☞

- Repeat the procedure, writing down the exact command line procedures
- Create the script. Use %1 in the script in stead of the name of the map, so you can use it as variables. Name the script: `weight`.
- Run the script for the map `geom` by typing : **run weight geom**
- If the script "hangs" check where the error occurs and correct it in the file, then run the script again, until everything is correct.

The initital part of the script for weight of evidence modeling is shown in figure 2.



Description: script for weights of evidence modelling

```
rem ILWIS Script
//The parameter %1 refers to the name of the factor map. It should be less than 7 characters long.
// Make sure that each map has a domain with the same name

//FIRST WE WILL DELETE EXISTING RESULT FILES
// the crosstable s%1.tbt
//The attribute table %1.tbt
// and we make a new attribute table

del s%1.*
del w%1.*
del %1.tbt
crtbl %1 %1

//NOW WE CROSS THE FACTOR MAP WITH THE ACTIVITY MAP
// This activity map was create with the script WEIGHTIN
// The cross table is called s%1

s%1=TableCross(%1.mpr,activity.mpr,IgnoreUndefs)
calc s%1.tbt

//Now we calculate one column in the cross table to indicate only the pixels with landslides.

Tabcalc s%1 npixact=iff((activity=1),NPix,0)

//NOW WE USE AGGREGATION FUNCTION, WITH OR WITHOUT A KEY TO CALCULATE:
//NCLASS = number of pixels in the class. We sum the values from columns Npix and group them by %1
//nslclass = number of pixels with landslides in the class.We sum the values from columns Npixact and group them by %1
//nmap = number of pixels with landslides in the map. We sum the values from columns Npix and don't group them
//nslide = number of pixels with landslide in the map. We sum the values from columns Npixact and don't group them
//THE RESULTS ARE NOT STORED IN THE CROSS TABLE S%1 BUT IN THE ATTRIBUTE TABLE %1

Tabcalc s%1 %1.nclass = ColumnJoinSum(s%1.tbt,Npix,%1,1)
```

Complete the script yourself

Figure 2: The first part of the script for Weights of evidence modeling

When the script runs without problems, it can  also be used for the other maps.

☞

- Also include the cross operation in the initial part of the script.
- Use the script to calculate the weights for the other factor maps: `geol`, `landuse` and `slope`.
- Indicate which classes are most important for the prediction of landslides. Order the ten most important classes in a table.
- Include this table and the script within your report for this exercise.

# 9. Combining weights of different factor maps

In real world situations, the occurrence of the phenomena of our interest (landslides, mineral deposits etc.) will seldom be associated with only one spatial variable. Landslide occurrence for example is not only dependent on slope steepness but also on the geological unit on which it occurs, and many other factors. Ultimately we want make a prediction based on a number of maps that are likely to be useful to predict the occurrence of landslides.

We can use the weights calculated for the individual factor map for an assessment which maps are important for the prediction in combination with our expert knowledge. In addition we can use the calculation of contrast factors as criteria to convert the input maps to binary patterns, in which the 1's indicate areas favorable for the occurrence of the phenomena of interest and the 0's the areas unfavorable for the occurrence of the same phenomena.

We have to make sure that the output tables with the weights can be linked to the factor maps. That is why the weight calculation is done in the attribute tables, and not in the cross tables.

The weights of the individual factor maps can be summated to obtain a total weight map. The higher the values in this map, the higher the susceptibility for landslides.

☞

- Generate attribute maps of the attribute `Wmap` for the various factor maps `slope`, `geom`, `geol` and `landuse`.
- Add a last line to your script `Weight`, that does this attribute map creation automatically.
- Sum up all the weight maps. Calculate a histogram of the final weight map
- How would you classify this map in the classes `high`, `moderate` and `low`?.

# 10. Final analysis and report writing

In the previous exercises you have learned the various steps for executing a weights of evidence analysis. However, the final map was not made by careful consideration of the input data. Nor were all the input layers used.

For the final assignment you are asked to perform weights of evidence modelling on the same data set more accurately.

The following input maps are at your disposal:

`geom` : the geomorphological map
`geol` : the geological map
`landuse` : the landuse map
`slope` : the slope map
`aspect`: the slope direction map
`roads` : a vector map with the roads
`fault` : a vector map with faults
`drainage` : a vector map containing drainage lines with different stream orders

Two landslide maps are available:
slide60 = landslides mapped from airphotos from the 1960's
slide80 = landslides mapped from airphotos from the 1980's, and field check

The analysis will be done on all active landslides.

The evaluation of the input data should be done by calculating the weights and contrast factors for the different classes and to calculate the most optimal buffers around linear and point features.

The calculation of weights will be done using the active landslides from the 1960's.

☞

- Work in groups of 2 persons.
- Evaluate all available input data.
- Use the script `Weight` to calculate the final weights for all factor maps. You can also use another script that contains the names of the factor maps and that runs the script`weight`.
- Sum up all the weight maps. Calculate a histogram of the final weight map. Create a final classified weight map
- Write a report on this exercise of maximum 4 pages .

# 11. Additional exercise: success rate & prediction rate

The “predictive power” of the resulting weight maps can be tested by analysing their success rate and prediction rate. The success rate is calculated by ordering the pixels of a susceptibility map in a number of classes, from high to low values, based on the frequency information from the histogram. After that an overlay is made with the landslide inventory map, and the joint frequency is calculated. The success rate indicates how much percentage of all landslides occurs in the pixels with the highest values in the different combination maps. For example, 50 percent of all landslides are predicted by 10 percent of the pixels with the highest value in the map.

Success rates are useful for two purposes:

Classification of the weight map into classes with a predefined percentage of all landslides. For example: you can then define high susceptibility as the class where 80% of all landslides are occurring, moderate as the class where 19% of all landslides happen, and low as the class where 1% of all landslides can happen.

The success rate is also used to check how well the final weight map can “predict” the landslide pattern with which it was made. It is of course essential that most of the landslides occur in the locations with the highest weights.

Apart from the success rate, the prediction rate really gives a good estimation of the predictive power of the map. The prediction rate is carried out with two landslide inventory maps of different periods. The oldest one is used to generate the model and the weight map is than combined with the recent landslide inventory map in order to test how well it predicts.

☞

- Create a script for the calculation of the success rate, using the information from the lecture handouts.
- Calculate the success rate using the landslide inventory map of the sixties: `slide60.` Make a graph from it and decide on the best boundary values for dividing the map in high, moderate and low susceptibility..
- Calculate the prediction rate using the landslide inventory map of the eighties: slide80.
- Include it in the report and you will get some more points.

## References

Bonham-Carter, G.F. (1994). Geographic Information Systems for Geoscientists. Modelling with GIS. *Computer Methods in the Geosciences*, 13: pp 267- 302. Pergamon.

# Annexures: scripts

## 1. Input script for Weight

```
rem ILWIS Script

//FIRST WE MAKE THE EVIDENCE MAP
//We make a column in the table slide indicating the active landslides only
tabcalc slide activeonly {dom=value.dom; vr=0:1:1}=iff((activity="active"),1,?)

//We make an attribute map from it
activ.mpr{dom=value.dom;vr=0:1} = MapAttribute(slide,activeonly)

//We make sure that the undefined values in the map become 0 so that the map has either 0 or 1 values
activity {dom=value.dom;vr=0:1} = iff(isundef(activ),0,1)
calc activity.mpr

//NOW WE RUN THE SCRIPT WEIGHT FOR THE INPUT MAPS
run weight slope
run weight geol
run weight geom
run weight aspcl
run weight landuse

//THE TOTAL WEIGHT MAP IS CALCULATED
wfinal {dom=value.dom ; vr = -10:10:0.001}=wslope+wgeol+wgeom+waspcl+wlanduse
calc wfinal.mpr

open wfinal.mpr
```

## 2. Weights of evidence modeling

```
rem ILWIS Script
//The parameter %1 refers to the name of the factor map. It should be less than 7 characters long.
// Make sure that each map has a domain with the same name

//FIRST WE WILL DELETE EXISTING RESULT FILES
// the crosstable s%1.tbt
//The attribute table %1.tbt
// and we make a new attribute table
```

```
del s%1.*
del w%1.*
del %1.tbt
crtbl %1 %1

//NOW WE CROSS THE FACTOR MAP WITH THE ACTIVITY MAP
// This activity map was create with the script WEIGHTIN
// The cross table is called s%1

s%1=TableCross(%1.mpr,activity.mpr,IgnoreUndefs)
calc s%1.tbt

//Now we calculate one column in the cross table to indicate only the pixels with landslides.

Tabcalc s%1 npixact=iff((activity=1),NPix,0)

//NOW WE USE AGGREGATION FUNCTION, WITH OR WITHOUT A  KEY TO CALCULATE:
//NCLASS = number of pixels in the class. We sum the values from columns Npix and group them by
%1
//nslclass = number of pixels with landslides in the class.We sum the values from columns Npixact and
group them by %1
//nmap = number of pixels with landslides in the map. We sum the values from columns Npix and don't
group them
//nslide = number of pixels with landslide in the map. We sum the values from columns Npixact and
don't group them
//THE RESULTS ARE NOT STORED IN THE CROSS TABLE S%1 BUT IN THE ATTRIBUTE TABLE
%1

Tabcalc s%1 %1.nclass = ColumnJoinSum(s%1.tbt,Npix,%1,1)
Tabcalc s%1 %1.nslclass = ColumnJoinSum(s%1.tbt,Npixact,%1,1)
Tabcalc s%1 %1.nmap = ColumnJoinSum(s%1.tbt,Npix,,1)
Tabcalc s%1 %1.nslide = ColumnJoinSum(s%1.tbt,Np ixact,,1)

//NOW WE CALCULATE THE FOUR VALUES NPIX1 - NPIX4 AS INDICATED IN THE EXERCISE
BOOK. THIS IS DONE IN THE ATTRIBUTE TABLE
// We correct for the situation when Npix1 - Npix3 might be 0 pixels, and change it into 1 pixel

Tabcalc %1 npix1 =IFF((nslclass>0),nslclass,1)
Tabcalc %1 npix2 = IFF((nslide-nslclass)=0,1,nslide-nslclass)
Tabcalc %1 npix3 = IFF((nclass-nslclass)=0,1,nclass-nslclass)
Tabcalc %1 npix4 = nmap-nslide-nclass+nslclass

//NOW WE CALCULATE THE WEIGHTS IN THE ATTRIBUTE TABLE
Tabcalc %1 wplus {dom=value.dom; vr=-10:10:0.00001} =
LN((npix1/(npix1+npix2))/(npix3/(npix3+npix4)))
```

```
Tabcalc %1 wminus {dom=value.dom; vr=-10:10:0.000001}  =
LN((npix2/(npix1+npix2))/(npix4/(npix3+npix4)))

//NOW WE CALCULATE THE CONTRAST FACTOR
Tabcalc %1 Cw = wplus-wminus

//NOW WE CALCULATE THE FINAL WEIGHT
//The final weight is the sum of the positive weight and the negative weights of the other classes
Tabcalc %1 WminSum=aggsum(wminus)
Tabcalc %1 Wmap=wplus+Wminsum-Wminus

//NOW WE MAKE AN ATTRIBUTE MAP OF THE FINAL WEIGHTS
w%1.mpr = MapAttribute(%1,%1.Wmap)
calc w%1.mpr
```

## 3. Success rate calculation

```
rem ILWIS Script
%1.his = TableHistogram(%1)
calc %1.his
tabcalc %1.his cumpctmax = ColumnAggregateMax(npcumpct,,1)
tabcalc %1.his npixcumpct:=npcumpct*(100/cumpctmax)
tabcalc %1.his inverse:=100-npixcumpct
tabcalc %1.his classes = CLFY(inverse,classes)

tabcalc classes %1 = ColumnJoinMax(%1.his,value,classes,1)

Rem Now make a class groupdomain %1 using the values from the table

rem ILWIS Script
REM %1 is the weight map
REM %2 is the landslide map
Rem Use slicing operation to classify weight map with percentage class boundaries

del %1.tbt
crtbl %1 %1

%1s.mpr{dom=%1.dom} = MapSlicing(%1,%1.dom)
calc %1s.mpr

Rem Cross the sliced map with the landslide map
%1c.tbt = TableCross(%1s,%2,IgnoreUndefs)
calc %1c.tbt
```

```
Tabcalc %1c.tbt npixact = iff(%2="landslide",npix,0)
Tabcalc %1.tbt npixact = ColumnJoinSum(%1c.tbt,npixact,%1s,1)
tabcalc %1.tbt npixactsum = ColumnAggregateSum(npixact,,1)
tabcalc %1.tbt percent=100*(npixact/npixactsum)
tabcalc %1.tbt cumpercent=cum(percent)
tabcalc %1.tbt classmax=value(%k)
tabcalc %1.tbt reorder=100-cumpercent
open %1.tbt

REM NOW Show the graph
```