ArcSIE

User's Guide

SIE LLC

Spatial Inference Enterprises

Last modified on March 19, 2013

Xun Shi
29 Heidi Ln
Bow, NH 03304
USA
dsm@arcsie.com

Table of Contents

Introduction

ArcSIE (SIE stands for Soil Inference Engine) is a toolbox for digital soil mapping. ArcSIE functions as an "Extension" of ArcMap.

ArcSIE generates soil maps based on the soil-environment model:

$$S = f(E)$$

This model states that the information about soil (S) can be derived from the information about the soil formative environment (E), including topography, geology, climate, vegetation, etc.

The most critical component in this model is the relationship between the soil and its environment (*f*). ArcSIE supports a *knowledge-based* approach to establishing this relationship. ArcSIE provides tools for soil scientists to formalize the relationship based on their knowledge of local soils. ArcSIE works with two types of knowledge:

- ***Rules*** defined by environmental feature values; and
- ***Cases*** defined in geographical space. Cases can be represented by **point**, **line**, **polygon**, and **cells**.

ArcSIE applies *rules* and/or *cases* to environmental data (stored as GIS data layers) to generate soil maps. The inference based on *rules* is called *rule-based reasoning* (**RBR**), and the inference based on *cases* is called *case-based reasoning* (**CBR**).

Cases can be applied to the entire mapping area (*global cases*) or a local region (*local cases*).
It is recommended that the user first applies RBR or global CBR (i.e., CBR using *global cases*) and creates draft maps. The user can then create and use *local cases*, as needed, to *fine-tune* the draft map in **exceptional** areas. This procedure can achieve both high efficiency (through the RBR or global CBR) and high accuracy (through the local CBR).

ArcSIE performs soil inference using fuzzy logic. The initial output from the inference is a series of *fuzzy membership maps* in raster format, one for each soil type under consideration.

In addition to the soil inference function, ArcSIE also provides tools for result validation, terrain analysis, pre- and post-processing for raster data, vetorization, and data format conversion.

Launch ArcSIE

Once you have installed ArcSIE, you can launch ArcSIE in the same way as you launch other ArcMap Extensions.

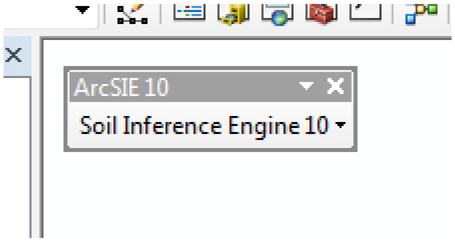
Like other ArcMap Extensions, before launching ArcSIE, you need to activate it. You only need to activate it once, unless you reinstall it or have just installed a new version.

If you are using ArcGIS 10.x, the steps to activate ArcSIE are as follows:

1. In the **Customize** menu of ArcMap, select **Extensions** and check **ArcSIE 10.0**.

2. In the **Customize** menu of ArcMap, select **Toolbars** and check **ArcSIE 10**.

Now you should see the ArcSIE menu bar appear in ArcMap:



Chapter 1. Inference

1.1. Fuzzy Soil Mapping Using RBR and CBR

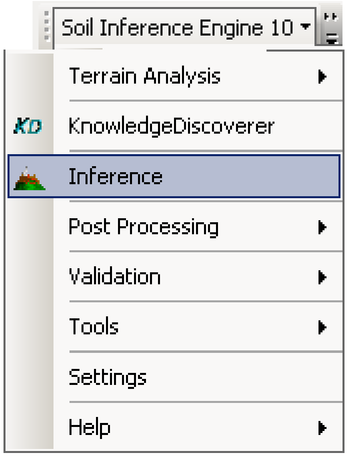
At the heart of ArcSIE is an *inference engine* performing *fuzzy* soil mapping based on the concept of **fuzzy soil classification**. Fuzzy soil mapping does not label the soil at a given location with a single soil name, but assigns to it the soil fuzzy membership values for all the soil types under consideration. The fuzzy membership values represent the similarities of the soil to those soil types. Rule-based reasoning (RBR) and case-based reasoning (CBR) are the two inference methods implemented by ArcSIE for calculating fuzzy membership values.

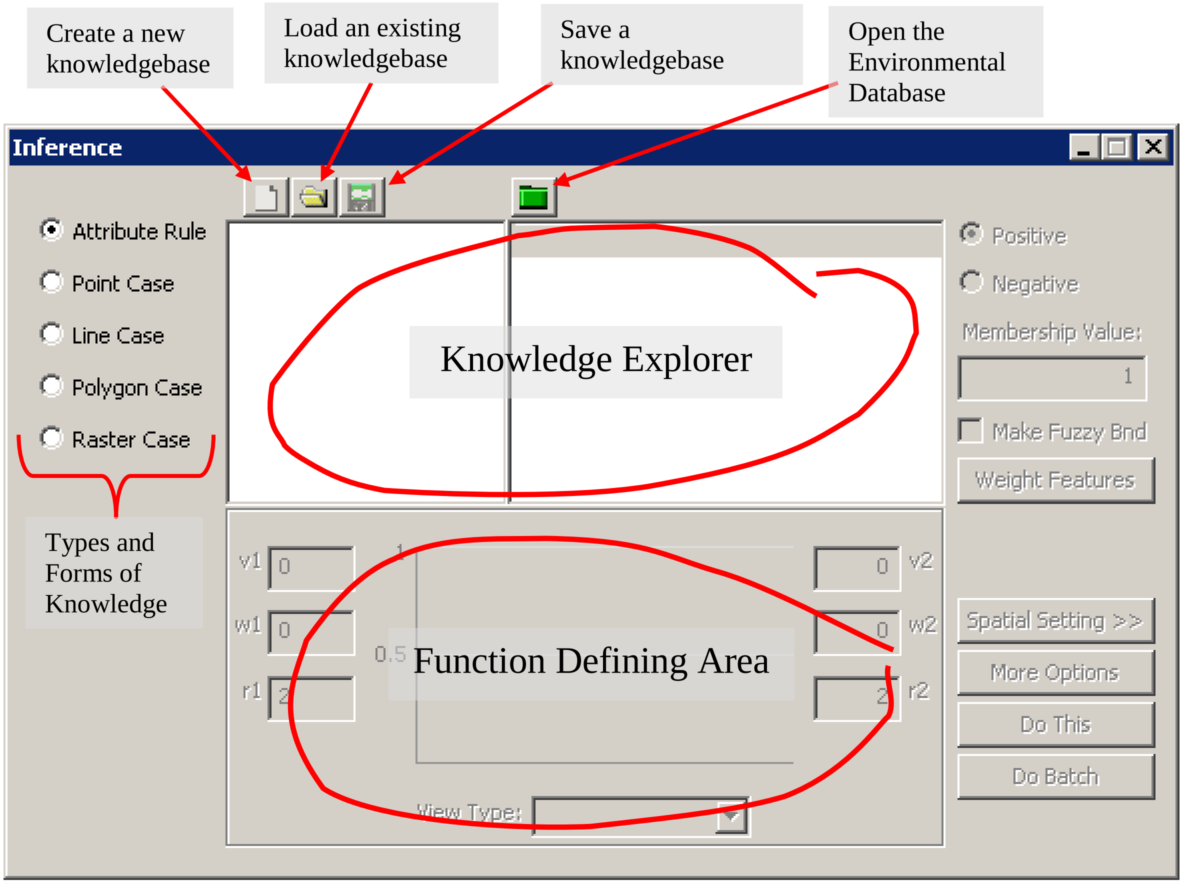
RBR is based on *rules* like "if the elevation is 1000 ft, then the soil is typical for type A". In this example, ArcSIE will use the GIS database to identify all the locations where elevations are 1000 ft, and assign full fuzzy membership to those locations for soil type A. The typical soil in this example is defined using only one environmental feature, elevation. More often, ArcSIE uses multivariate inference to determine the fuzzy membership based on more than one environmental feature. In ArcSIE, a rule is represented by a fuzzy membership function defining the relationship between the values of an environmental feature and the values of fuzzy membership for a given soil type.

In RBR, the 1000 ft elevation in the above example is explicitly specified by the soil scientist. In **CBR,** this value is retrieved from the GIS database at the location(s) pinpointed or delineated by the soil scientist. The soil scientist identifies these *case* locations as places where the soil is typical for the given soil type. CBR is useful when explicit rules based on environmental feature values are not easy to state, but the soil scientist is able to visually identify the landscape units or locations for the given soil type using a topographic map, DEM, or orthophoto.

1.2. Launching the Inference Engine

ArcSIE provides an integrated interface for RBR and CBR. In the Soil Inference Engine menu, clicking *Inference* opens the Inference dialog box.





1.3. Preparing the *Environmental Database*

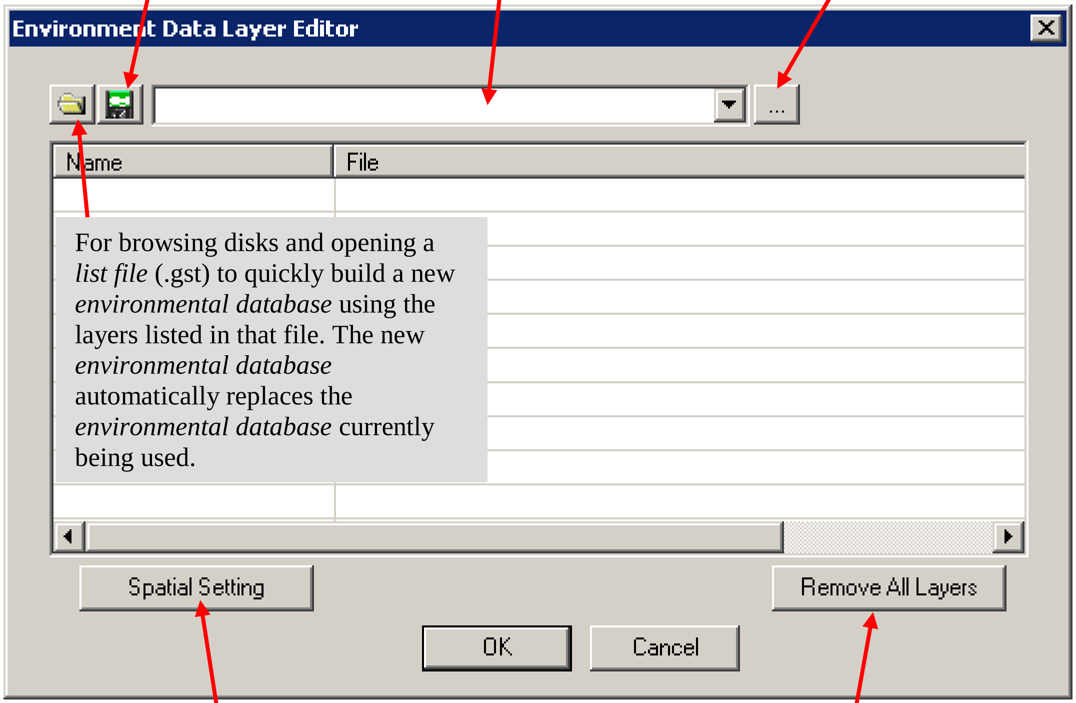
Before any inference can be performed, environmental data layers must be loaded to ArcSIE to build the *environmental database*. The *environmental database*:

- provides data for the *environmental features* used in the soil inference; and
- defines the spatial extent and resolution of the resulting raster soil maps.

For saving the feature names and corresponding Grid names currently in the *environmental database* into a *list file* (.gst, standing for "grid list") for future use.

For selecting from the layers currently loaded in ArcMap to add to the *environmental database*.

For browsing disks to choose a data layer to add to the *environmental database*.



For browsing disks and opening a *list file* (.gst) to quickly build a new *environmental database* using the layers listed in that file. The new *environmental database* automatically replaces the *environmental database* currently being used.

For opening the Spatial Setting dialog box to set spatial extent and cellsize for the *environmental database* (Sec. 1.3.2).

Remove all data layers from the current *environmental database*.

Once you have built an *environmental database*, you can use it across different types of inferences. For example, when you switch from "Attribute Rule" to "Point Case", you do not re-build the *environmental database*; the CBR will use the same *environmental database* currently loaded. In other words, once the *environmental database* is built, it will be shared by all different types of inferences.

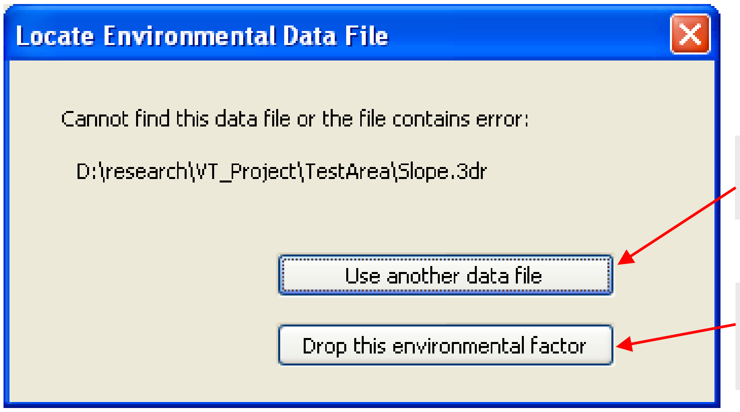
1.3.1. Creating and Editing the *Environmental Database*

All environmental data layers must be raster data in the ArcInfo Grid format.

In the Inference dialog box, clicking 🟩 launches the Environmental Data Layer Editor dialog box (see next page).

If a layer is successfully added to the *environmental database*, its name and data source will appear in the form at the middle of the dialog box.

If an error occurs when ArcSIE tries to load a data layer specified in the *list file*, e.g., the Grid cannot be found in the specified folder, ArcSIE will display the following dialog box:


Locate Environmental Data File
Use another data file
Drop this environmental factor

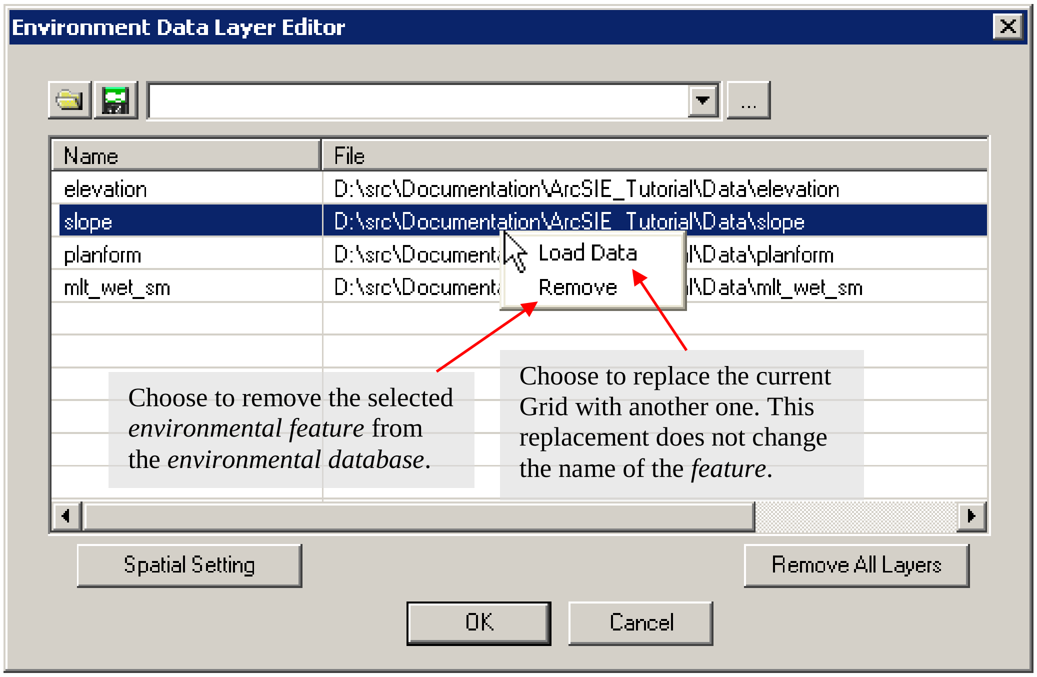
Cannot find this data file or the file contains error:

D:\research\VT_Project\TestArea\Slope.3dr

Choose to browse disk to select another Grid.

Choose to not include this feature in the *environmental*

Once the data have been added, right-clicking on a layer opens a popup menu:



Name	File
elevation	D:\src\Documentation\ArcSIE_Tutorial\Data\elevation
slope	D:\src\Documentation\ArcSIE_Tutorial\Data\slope
planform	D:\src\Document l\Data\planform
mlt_wet_sm	D:\src\Document l\Data\mlt_wet_sm

Choose to remove the selected *environmental feature* from the *environmental database*.

Choose to replace the current Grid with another one. This replacement does not change the name of the *feature*.

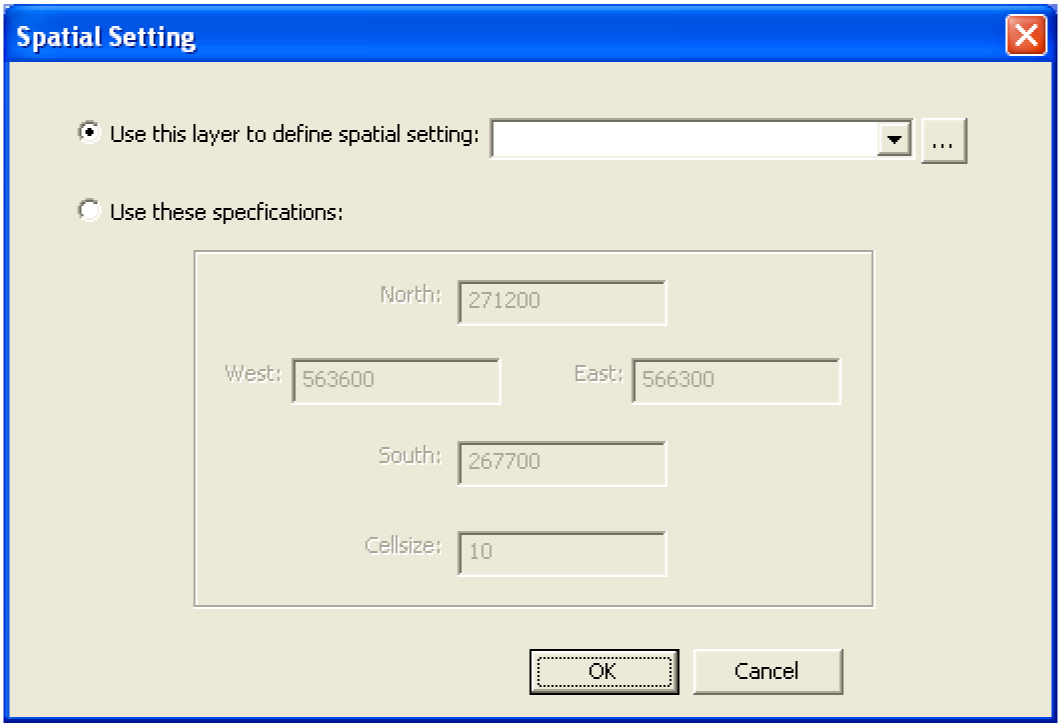
1.3.2. Spatial Setting of the *Environmental Database*

You can load data layers with different spatial extents and different spatial resolutions to the *environmental database*. Once they are loaded, they will be automatically aligned based on the *spatial setting* of the *environmental database*.

However, these data layers MUST be already in the same coordinate system, and this coordinate system should also be the system of the current mxd of ArcMap.

The *spatial setting* of the *environmental database* determines the spatial extent and resolution of the inference result, but does not affect the output from other tools in ArcSIE.

The default *spatial setting* is determined by the first loaded data layer. You can customize the *spatial setting* using the Spatial Setting dialog box. Clicking Spatial Setting in the Environmental Data Layer Editor brings the Spatial Setting dialog box to the desktop.



Spatial Setting

● Use this layer to define spatial setting:

○ Use these specfications:

North: 271200

West: 563600

East: 566300

South: 267700

Cellsize: 10

On this dialog box, you can choose to use a data layer as the template to set spatial extents and cellsize, or you can set arbitrary values for these settings

1.3.3. Naming an Environmental Data Layer

The default name of a layer is the name of the Grid for that layer.

In the Environmental Data Layer Editor dialog box, clicking on the name of a data layer makes the name editable, allowing you to rename the data layer.

When you create a new *knowledgebase* (see sec. 1.4.3) based on the current *environmental database*, the names of the environmental data layers in the current *environmental database* will be used as the names of the *environmental features* in the new knowledgebase.

When you load an existing knowledgebase, the names of *environmental features* in this knowledgebase must match the names of the environmental data layers in the current *environmental database*. ArcSIE automatically checks the names when you load an existing knowledgebase and presents a warning if the two sets of names do not exactly match.

1.3.4. Finishing Creation of an *Environmental Database*

Clicking OK closes the Environmental Data Layer Editor dialog box. This completes the creation of the *environmental database*, and returns you to the Inference dialog box.

The Environmental Data Layer Editor dialog box can also be used to edit the *environmental database* after it is created. Changes made to the *environmental database* will be reflected in the *feature* list in the Inference dialog box.

1.4. Specifying Knowledge Type

1.4.1. Types and Forms of Knowledge

ArcSIE supports two general types of knowledge: a *rule* that is directly defined in the attribute (i.e., *environmental feature*) space and a *case* that is initially defined in the geographical space. A *case* can be in point, line, polygon, and cell forms.

Attribute Rules are used to perform rule-based reasoning (RBR). They can be used to represent the soil scientist's explicit knowledge between soils and *environmental features*. *Attribute rules* are created in ArcSIE and all their values are directly specified by the user.

Point Cases (also called *tacit points*) are used to perform point case-based reasoning (point CBR). They can be used to represent the soil scientist's knowledge of the soils at specific locations.

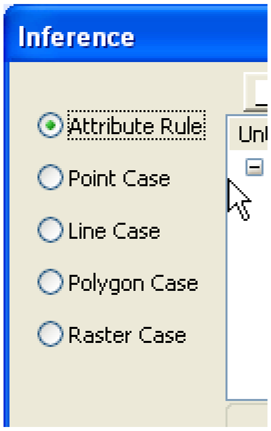
Line Cases can be used to represent the soil scientist's knowledge about features that have linear shapes. For example, they can be used to represent typical locations of a soil that occur along ridgelines. In ArcSIE, a *line case* is treated as a series of independent *point cases*. The locations of these "points" are determined by the length and orientation of the line and the cellsize of the *environmental database*. The inference settings specified for a *line case* (see sections 1.6.5–1.6.8) are applied to every "point" in the series.

Polygon Cases can be useful if the soil scientist wants to delineate the typical locations of a soil as polygons rather than points. In ArcSIE, a *polygon case* is treated as an agglomeration of independent *point cases*. The locations of these "points" are determined by the coverage of the polygon and the cellsize of the *environmental database*. The inference settings specified for a *polygon case* (see sections 1.6.5–1.6.8) are applied to every "point" in the series.

Cell Cases are usually derived from a terrain analysis process. For example, streamlines derived from a DEM can be used as the typical locations for valley areas. In ArcSIE, one *cell case* is treated as *point case*. The inference settings specified for a *raster casebase* (see sections 1.7.4–1.7.5) are applied to every *cell case* in this *casebase*.

1.4.2. Specifying Knowledge Type and Form

Before creating or loading a knowledgebase and conducting the inference, you should specify which type and form of knowledge you are using on the Inference dialog box:



Inference

● Attribute Rule
○ Point Case
○ Line Case
○ Polygon Case
○ Raster Case

Attribute Rule for RBR is the default option when you first launch the Inference dialog box.

1.4.3. Structure of the ArcSIE Knowledgebase

ArcSIE organizes different representations of soil scientists' knowledge, including *rules* and *cases*, into different *knowledgebases*. A knowledgebase that contains *rules* and for performing rule-based reasoning (RBR) is called a *rulebase,* and a knowledgebase that contains *cases* and for performing case-based reasoning (CBR) is called a *casebase*.

A *rulebase* in ArcSIE has the following structure:

```
rulebase
        |
        soil type (instance list)
                |
                instance
                        |
                        rule
```

A *rulebase* contains knowledge of one or more *soil types*. A *soil type* may have one or more *instances*, each describing a unique environmental configuration for the *soil type*. For example, a soil may occur on both south-facing and north-facing slopes, but the elevation and slope gradient conditions for the soil on the two aspects may be different. These two environmental configurations form two *instances* of the soil. An *instance* may have one or more *rules*, each characterizing the relationship between the fuzzy membership of the *soil type* and a specific *environmental feature*.

A knowledgebase for performing case-based reasoning (CBR) is called a *casebase*. A *casebase* containing *vector cases* (point, line, and polygon) has the following structure:

```
casebase
        |
        soil type (case list)
                |
                case  (either point, line, or polygon)
                        |
                        rule and spatial setting
```

A *casebase* contains *raster (cell) cases* has the following structure:

```
casebase
        |
        case (cell)
                |
                rule and spatial setting
```

A *case* differs from an *instance* in that a *case* has a *spatial setting*. A *spatial setting* contains the spatial information about a *case*, including its location and some parameter values for performing local CBR. In terms of knowledge acquisition, the fundamental difference between an *instance* and a *case* is on how their *central values* are determined: the *central values* of an *instance* are directly specified by the user, whereas the *central values* of a *case* are identified by the inference engine according to the location of the *case*.

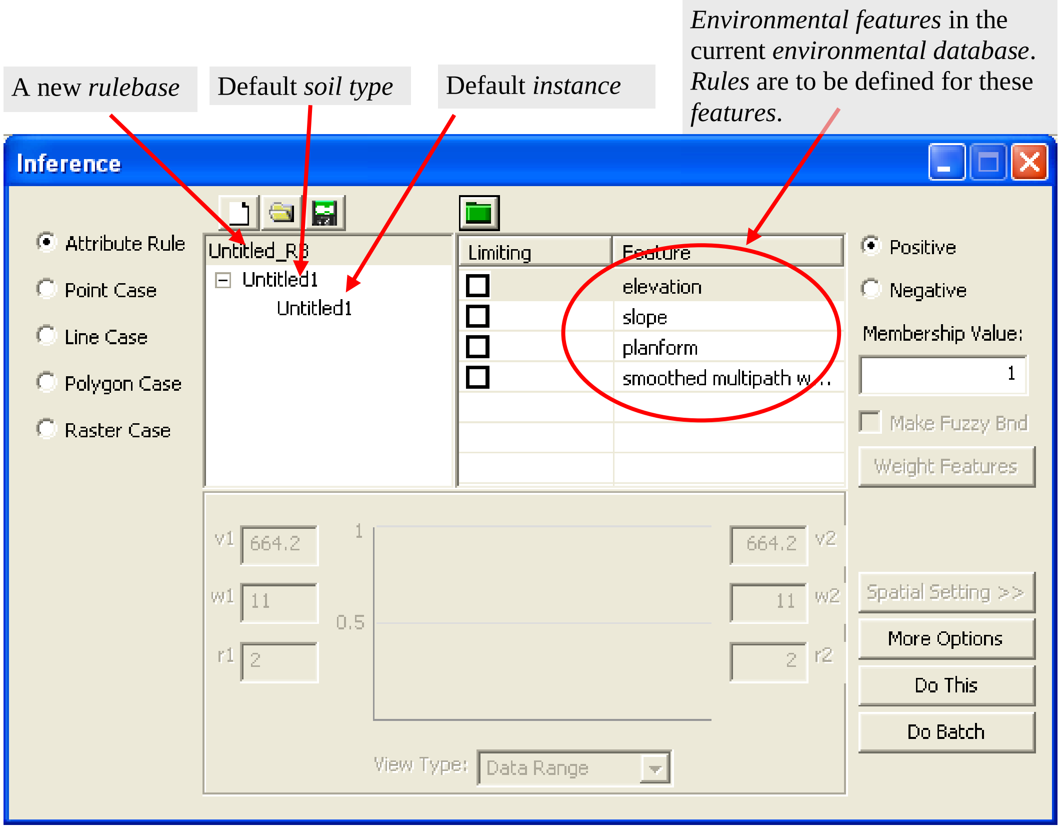
1.5. Preparing a *Rulebase*

1.5.1. Creating a New *Rulebase*

Before a new *rulebase* can be created, an *environmental database* must have been loaded (see 1.3). The loaded *environmental database* determines for what *environmental features* new *rules* will be created.

1.5.1.1. Creating a New *Rulebase*

In the Inference dialog box, click to create a new *rulebase*. The new *rulebase*, whose structure appears in the left pane of the Inference dialog box, contains one default *soil type*, and the default *soil type* contains one default *instance*. Default *rule(s)* will be created for that *instance* based on the *environmental features* contained in the current *environmental database*:


A new *rulebase*

Default *soil type*

Default *instance*

Environmental features in the current *environmental database*. *Rules* are to be defined for these *features*.

1.5.1.2. Creating a New *Soil Type*

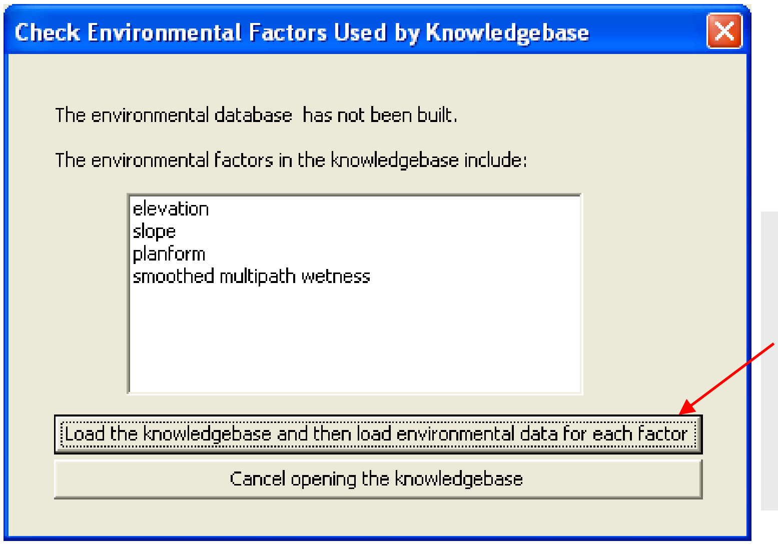
Right-click the *rulebase* name in the Knowledge Explorer and select *New Soil Type* from the drop-down menu. The new *soil type* will be appended to the end of the list of *soil types*. It will contain a default *instance*.

1.5.1.3. Creating a New *Instance*

Right-click the *soil type* for which you want to create a new *instance* and select *New Instance*. The new *instance* is appended to the end of the list of *instances* in the current *soil type* and is assigned a unique default name. The new *instance* contains default *rule(s)* for the *environmental features* supported by the current *environmental database*.

1.5.2. Loading an Existing *Rulebase*

Click 📂 on the Inference dialog box to load an existing *rulebase*. The file format of a *rulebase* is DBF (.dbf). If you try to open an existing *rulebase* before creating or loading the *environmental database*, a dialog box will appear to guide you to load both:



Check Environmental Factors Used by Knowledgebase

The environmental database has not been built.

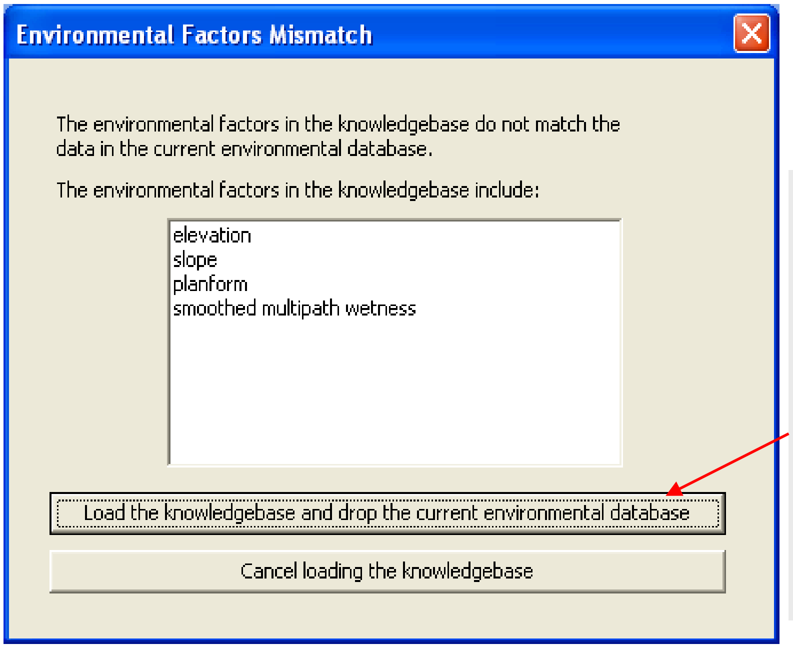
The environmental factors in the knowledgebase include:

elevation
slope
planform
smoothed multipath wetness

Load the knowledgebase and then load environmental data for each factor
Cancel opening the knowledgebase

Opens the *Environment Data Layer Editor*. The *environmental features* required by the *rulebase* will be listed in the Editor. Right-click on a *feature* and select *Load Data* to load the data file for that *feature*.

If the knowledgebase you are opening contains *environmental features* that do not match the *features* in the current *environmental database*, a dialog box will appear to show options for resolving the mismatch.



Environmental Factors Mismatch

The environmental factors in the knowledgebase do not match the data in the current environmental database.

The environmental factors in the knowledgebase include:

elevation
slope
planform
smoothed multipath wetness

Load the knowledgebase and drop the current environmental database
Cancel loading the knowledgebase

Choose to drop the original *environmental database* and load a new one. This will open the *Environment Data Layer Editor*. The *environmental features* required by the *rulebase* will be listed in the Editor. Right-click on a *feature* and select *Load Data* to load the

At any time, ArcSIE works with only one *rulebase*. Creating a new *rulebase* or loading an existing *rulebase* from hard drive will unload the *rulebase* currently being used, if any. You will be prompted to save the changes you made to the current *rulebase* before it is unloaded.

1.5.3. Saving a *Rulebase*

Clicking 💾 saves the current *rulebase* into a *rulebase* file. A *rulebase* file is in the format of dbf table (.dbf). All *soil types*, *instances*, and *rules* in the current *rulebase* will be saved into this file.

1.5.4. Unloading a *Rulebase* and Removing a *Soil Type* and *Instance*

1.5.4.1. Unloading the *Rulebase*

Right-click the *rulebase* name and select *Remove* to unload the current *rulebase* from ArcSIE. You will be prompted to save the changes you made to the current *rulebase* before it is unloaded.

1.5.4.2. Removing a *Soil Type*

Right-click the *soil type* you want to remove and select *Remove* to remove the *soil type*. Note: This will physically and permanently remove all the content of this *soil type*.

The last *soil type* in a *rulebase* cannot be removed.

1.5.4.3. Removing an *Instance*

Right-click the *instance* you want to remove and select *Remove* to remove the *instance*. Note: This will physically and permanently remove all the content of this *instance* from the *rulebase*.

The last *instance* in a *soil type* cannot be removed.

1.5.5. Changing Names

To change or edit the name of an existing *rulebase*, *soil type*, or *instance*, click on it and type in the new name.

In a *rulebase*, each *soil type* must have a unique name; and in a *soil type*, each *instance* must have a unique name. However, *instances* in different *soil types* can have identical names.

1.5.6. Editing Knowledge

The equation below describes how the knowledge of a given *soil type* will be used in an RBR process:

$$s_{ij,k} = \overset{n}{\underset{g=1}{T_k}} \{ \overset{m}{\underset{a=1}{P_g}} [E_{g,a} (z_{ij,a} , z_{g,a})]\}$$

The meanings of the symbols in the above equation are given as follows:

Symbol	Meaning
$s_{ij,k}$	The fuzzy membership value at location (*i, j*) for soil *k*.
m	The number of *environmental features* used in the inference.
n	The number of *instances* for soil type *k*.
$z_{ij,a}$	The value of the a^{th} *environmental feature* at location (*i, j*).
$Z_{g,a}$	The most optimal range given by *instance g*, defining the most favoring condition of feature *a* for soil *k*.
E	The function for evaluating the optimality value at the environmental feature level.
P	The function for evaluating the fuzzy membership at the *instance* level.
T	The function for deriving the final fuzzy membership value for soil *k* at site (*i, j*) based on all the *instances* for soil *k*.

For the *T* function, ArcSIE implements the *max* function for RBR, i.e., if more than one *instance* is used for a soil type and these *instances* give different fuzzy membership values for that soil type at a given location, the maximum value will be assigned to that location.

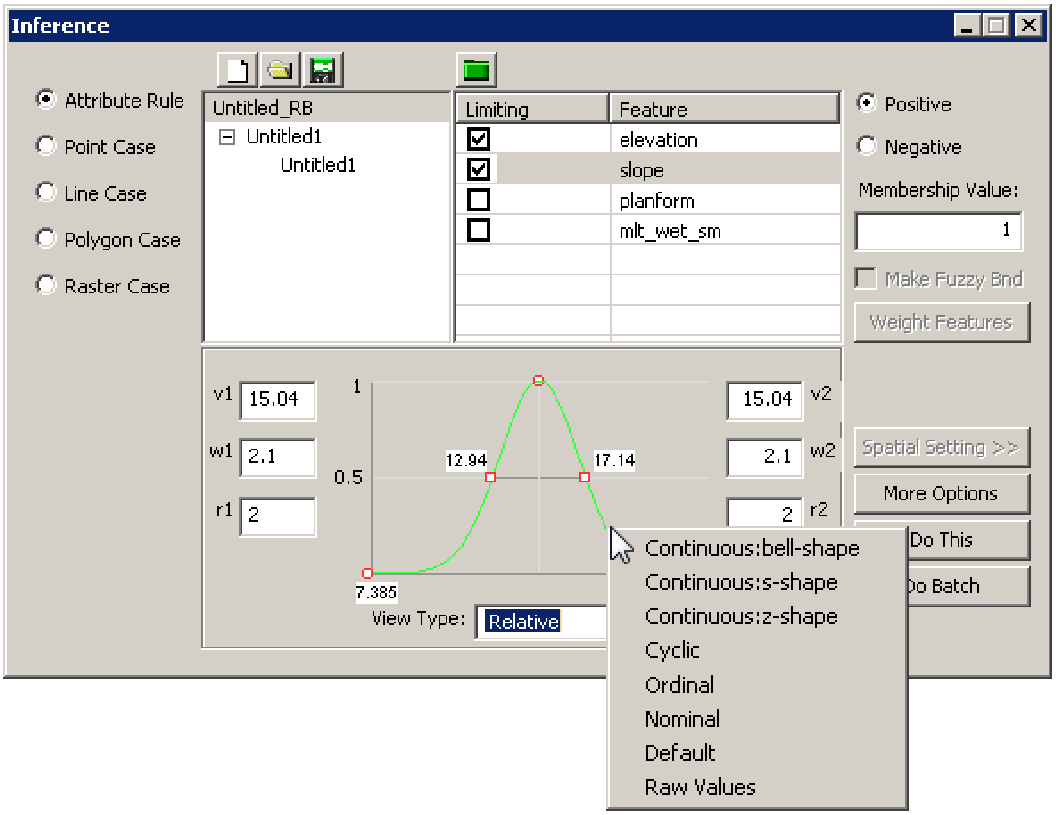
ArcSIE allows the user to adjust the *E* and *P* functions.

1.5.6.1. Adjusting the *E* Function

1.5.6.1.1. Selecting a Function Type

ArcSIE provides five choices for E in the general inference equation: *continuous (three types)*, *cyclic*, *ordinal*, *nominal*, and *raw values*, based on the nature of the *environmental feature*. There is also a *default* choice.

To assign a function to the current *environmental feature*, move the cursor into the function defining area of the Inference dialog box and right-click the mouse button. A pop-up menu will open and show a list of the *E* functions implemented by ArcSIE. Choosing a function from the list will assign that function to the current *environmental feature*.



Limiting	Feature
☑	elevation
☑	slope
□	planform
□	mlt_wet_sm

Membership Value: 1

v1 15.04
w1 2.1
r1 2
v2 15.04
w2 2.1
r2 2

View Type: Relative

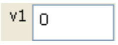
After selecting the function type, you can adjust the parameter values for the function to precisely define it.

1.5.6.1.1.1. Continuous Function

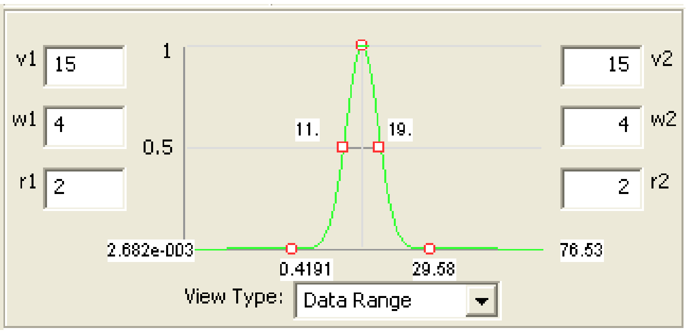
The *continuous* function is applicable to *environmental features* with interval or ratio values (e.g., temperature, elevation, and slope gradient). The mathematical representation of a continuous *E* function is as follows:

$$\begin{cases} s_{ij,k,a} = \max \times e^{[(z_{ij,a} - v_1)/w_1]^{r_1} \ln(c_1)} & if \;\; z_{ij,a} < v_1 \\ s_{ij,k,a} = \max & if \;\; v_1 \le z_{ij,a} \le v_2 \\ s_{ij,k,a} = \max \times e^{[(z_{ij,a} - v_2)/w_2]^{r_2} \ln(c_2)} & if \;\; z_{ij,a} > v_2 \end{cases}$$

This equation group defines two halves of a Gaussian-style function curve (illustrated by the graphic in the following pages). It uses two sets of parameters to provide the flexibility of defining an asymmetric curve. the symbols in the equation group are explained in the table below.

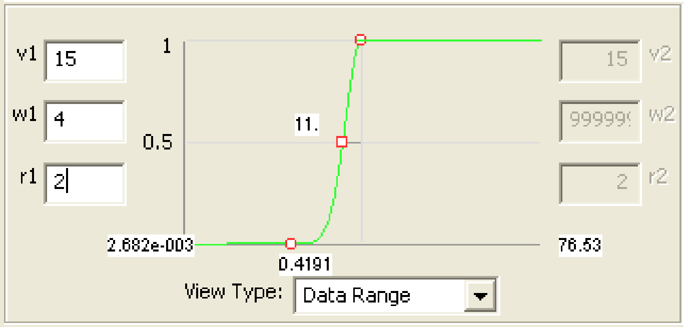
Symbol	***Meaning***
$s_{ij,k,a}$	The optimality value of *environmental feature a* at location (*i*, *j*) for soil type *k*. This is the output from the *E* function.
max	The maximum fuzzy membership specified in Membership Value: 1 , typically to be 1.
e	The base of the natural logarithm (2.71828 …).
$z_{ij,a}$	The value of *environmental feature a* at location (*i*, *j*), read from the *environmental database*.
v_1 and v_2	The two central values that define the lower and upper limits of the *most optimal range* of *environmental feature a* for soil *k*. In other words, if $z_{ij,a}$ falls between v_1 and v_2, then location (*i*, *j*) gets the maximum optimality value for soil *k* regarding *environmental feature a*. Technically, v_1 and v_2 determine the width of the flat top of the function curve. These two values can be adjusted by editing the values in  v1 0 and 0 v2. **It is required that v_1 is less than or equal to v_2.**
w_1 and c_1	Parameters for adjusting the shape of the left half of the curve. Specifically, if $z_{ij,a}$ is smaller than v_1 and the difference between them is w_1, the output optimality value is c_1. For simplification, ArcSIE uses a fixed value, **0.5**, for c_1. Therefore, w_1 is used to specify what value of the current *environmental feature* should correspond to an optimality value of 0.5. For example, if the *environmental feature* is elevation, and the soil scientist specifies v_1 = 800 ft and w_1 = 20 ft, then a location with elevation of 780 (800-20) ft will get an optimality value of 0.5.
w_2 and c_2	Parameters for adjusting the shape of the right half of the curve. Specifically, if $z_{ij,a}$ is greater than v_2 and the difference between them is w_2, the output optimality value is c_2. For simplification, ArcSIE uses a fixed value, **0.5**, for c_2. Therefore, w_2 is used to specify what value of the current *environmental feature* should correspond to an optimality of 0.5. For example, if the *environmental feature* is elevation, and the soil scientist specifies v_2 = 1000 ft and w_2 = 20 ft, then a location with elevation of 1020 (1000+20) ft will get an optimality value of 0.5.
r_1 and r_2	The two values controlling the flatness of the top parts of the curve that are beyond the v_1-to-v_2 range and the steepness of the side parts of the curve. The higher these two values, the flatter the tops and the steeper the sides.
ln	The natural logarithm.

ArcSIE provides three basic function curves based on which you can further fine-tune the curve shape:



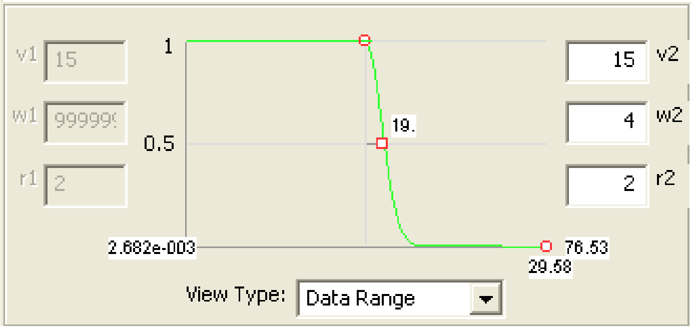
v1 15
w1 4
r1 2
v2 15
w2 4
r2 2
View Type: Data Range

bell-shape: A symmetric shape. The optimality value decreases as the difference between the *environmental feature* value and the central values (v_1 and v_2) increases. The example below shows a curve defining that 15% slope is optimal, i.e., receiving the highest membership, and the membership value decreases at the same rate on both sides as slope increases or decreases from 15%.



v1 15
w1 4
r1 2
v2 15
w2 99999(
r2 2
View Type: Data Range

s-shape: A the-higher-the-better shape. The optimality will always get the maximum value if the *environmental feature* values are greater than v_2. The example at the top of the next page shows a curve defining that slopes steeper than 15% are always optimal, i.e., receiving the highest membership.

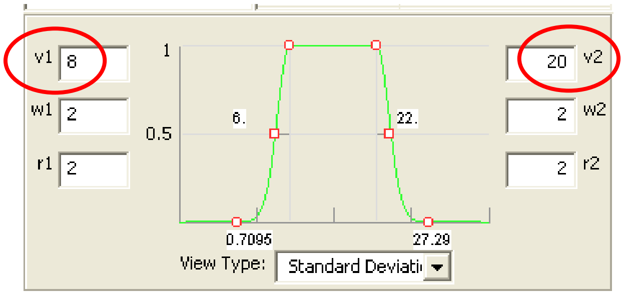


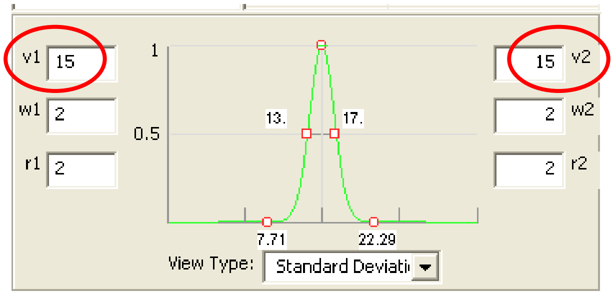
v1 15
w1 99999(
r1 2
v2 15
w2 4
r2 2
View Type: Data Range

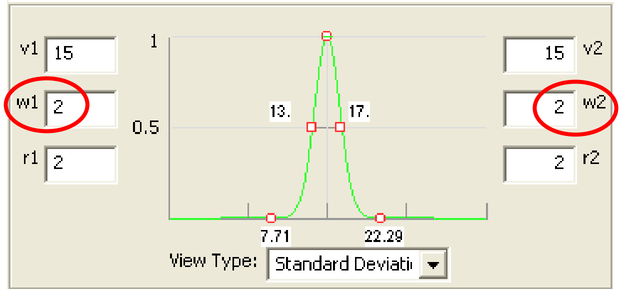
z-shape: A the-lower-the-better shape. The optimality will always get the maximum value if the *environmental feature* values are smaller than v_1. The example below shows a curve defining that slopes gentler than 15% are always optimal, i.e., receiving the highest membership.

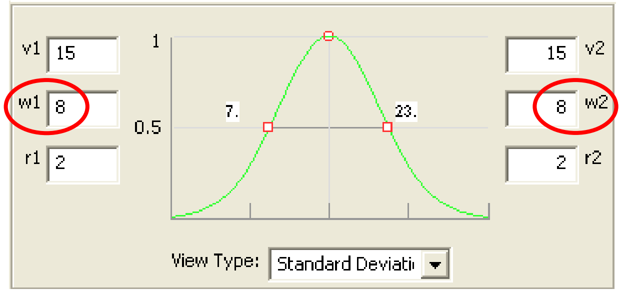
The following examples show how to change the function curve by adjusting different parameters.

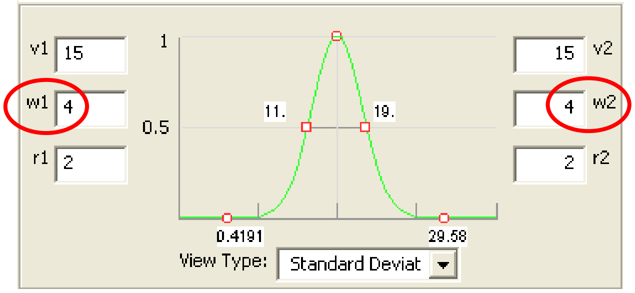
The two examples on the left show the effects on the function curve by adjusting *v*.


v1 8
w1 2
r1 2
v2 20
w2 2
r2 2
View Type: Standard Deviati


v1 15
w1 2
r1 2
v2 15
w2 2
r2 2
View Type: Standard Deviati

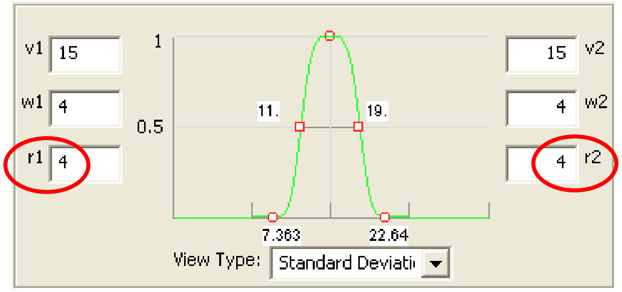

v1 15
w1 2
r1 2
v2 15
w2 2
r2 2
View Type: Standard Deviati


v1 15
w1 8
r1 2
v2 15
w2 8
r2 2
View Type: Standard Deviati

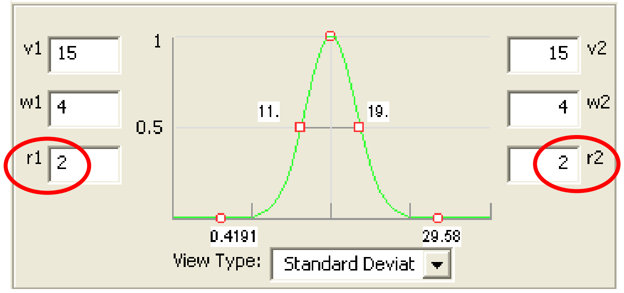

v1 15
w1 4
r1 2
v2 15
w2 4
r2 2
View Type: Standard Deviat

The three examples on the right show the effects on the function curve by adjusting *w*.

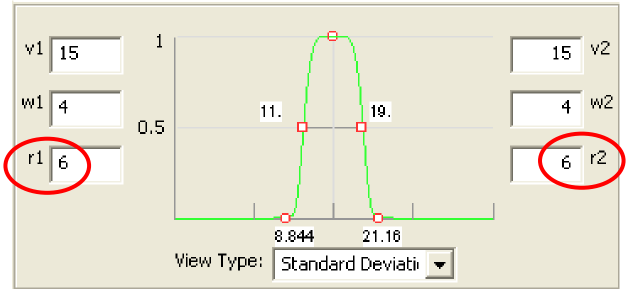
The three examples on the left show the effects on the function curve by adjusting *r*:



v1 15
w1 4
r1 4
v2 15
w2 4
r2 4
View Type: Standard Deviati

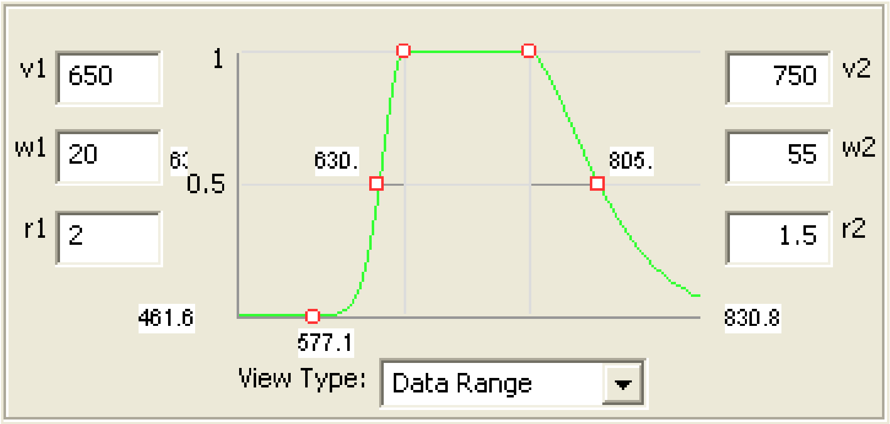


v1 15
w1 4
r1 2
v2 15
w2 4
r2 2
View Type: Standard Deviat



v1 15
w1 4
r1 6
v2 15
w2 4
r2 6
View Type: Standard Deviati

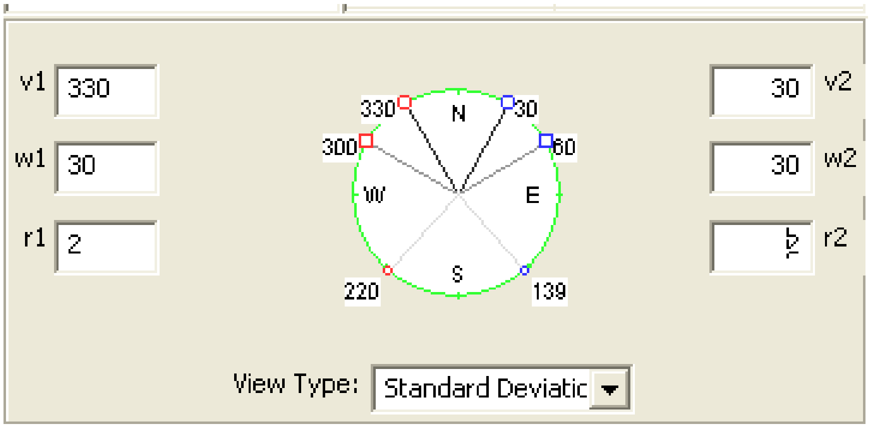
The example below shows an example of asymmetric function curve:



v1 650
w1 20
r1 2
v2 750
w2 55
r2 1.5
View Type: Data Range

1.5.6.1.1.2. Cyclic Function

The cyclic function is for a directional *feature* such as slope aspect. It is similar to the continuous function except that the maximum value goes back to the minimum value. ArcSIE represents the "curve" of a cyclic function using a circle:



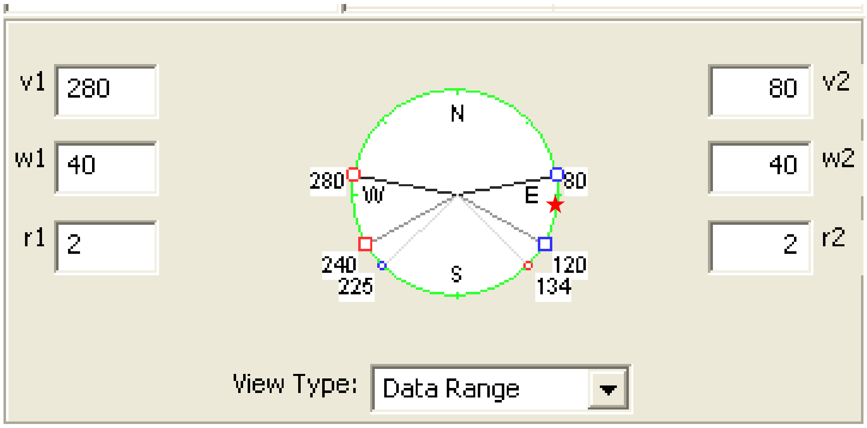
v1 330
w1 30
r1 2
v2 30
w2 30
r2 2
View Type: Standard Deviatic

On the circle:

- Red color represents values on the "left" side, i.e., v_1, w_1, and r_1.
- Blue color represents values on the "right" side, i.e., v_2, w_2, and r_2.
- The two bigger circular marks represent v_1 and v_2. Their positions on the circle are determined by the values of v_1 and v_2.
- The two square handles correspond to w_1 *and* w_2. Like a continuous function curve, you can adjust the curve by dragging these two handles. The handles will slide along the circle.
- The two smaller circular marks label the feature values at which the membership will be very low (corresponding to the two circular marks at the bottom of a continuous function curve). The positions of these two marks on the circle will be automatically calculated based on the current values of the *v*, *w*, and *r* parameters.

For a cyclic feature:

- The measure of direction is in degrees, i.e., v_1, v_2, w_1, and w_2 must be between 0 and 360.
- The measure of direction starts from N (0˚), increases clockwise, and goes back to N (360˚).
- If z_{ij}^{v} falls into the range between v_1 and v_2 (the range is defined along the clockwise direction), the function outputs the highest membership value (e.g., 1); Otherwise, the membership value will be calculated using the continuous function specified by the *v*, *w*, and *r* values.
- The membership at a location can be calculated from either side, clockwise or counterclockwise. In ArcSIE, the greater value will be used (see the example in next page).



v1 280

w1 40

r1 2

v2 80

w2 40

r2 2

View Type: Data Range

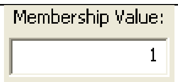
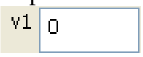
In the above graphic, the location represented by the red star has aspect = 100˚. When calculated from the red side (counterclockwise and using v_1, w_1, and r_1), the optimality value for this location will be close to 0, since it is beyond that small red circle (134˚); however, when calculated from the blue side (clockwise and using v_2, w_2, and r_2), its value will be > 0.5, since it does not pass the blue square (120˚). The value > 0.5 will be used as the output value.

1.5.6.1.1.3. Ordinal Function

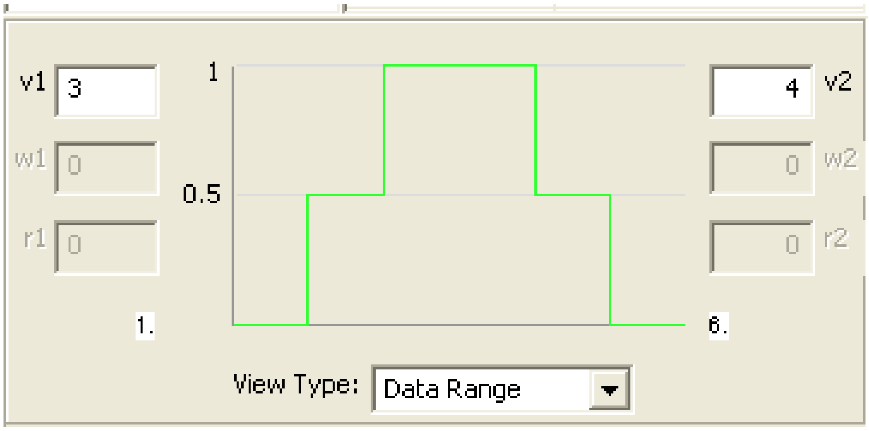
ArcSIE uses the following function for ordinal data:

$$\begin{cases} s_{ij,k,a} = \max\times(z_{ij,a} - z_{min}) / (v_1 - z_{min}) & \textit{if } z_{ij,a} < v_1 \\ s_{ij,k,a} = \max & \textit{if } v_1 \le z_{ij,a} \le v_2 \\ s_{ij,k,a} = \max\times(z_{max} - z_{ij,a}) / (z_{max} - v_2) & \textit{if } z_{ij,a} > v_2 \end{cases}$$

The symbols in the equation group are explained as follows:

Symbol	***Meaning***
$s_{ij,k,a}$	The optimality value of *environmental feature a* at location (*i*, *j*) for soil type *k*. This is the output from the *E* function.
max	The maximum fuzzy membership the user specifies in  Membership Value: 1 . This value is typically 1.
$z_{ij,a}$	The value of *environmental feature a* at location (*i*, *j*). This value is read from the *environmental database*.
z_{min}	The minimum value in the *environmental database* for *environmental feature a.*
z_{max}	The maximum value in the *environmental database* for *environmental feature a.*
v_1 and v_2	These are two user-specified central values that define the lower and upper limits of the *most optimal range* of *environmental feature a* for soil *k*. In other words, if $z_{ij,a}$ falls between v_1 and v_2, then location (*i*, *j*) will get the maximum optimality value for soil *k* in terms of *environmental feature a*. Technically, v_1 and v_2 determine the width of the flat top of the function curve. These two values can be adjusted by editing the values in  v1 0 and 0 v2 . **It is required that v_1 is less than or equal to v_2.**

The idea here is to use a linear function to quantify the difference between ordinal values, which is convenient for converting ordinal values into continuous values. Because of the nature of ordinal data, the ordinal function assumes the values for this *environmental feature* to be integers, and v_1 and v_2 should also be integers. It is also required that v_1 is less than or equal to v_2. Since w and r are not part of the mathematical function used for the ordinal data (see previous page), on the interface they are not adjustable for the ordinal function. Below is an example of using the ordinal function.



v1: 3
w1: 0
r1: 0
v2: 4
w2: 0
r2: 0
View Type: Data Range

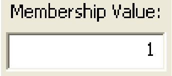
In the example illustrated by the above graphic, the user defines an ordinal function for a re-classified slope gradient layer. In the layer, slope gradients have been classified into six classes (1–6). The user specifies that classes 3 and 4 are optimal, and membership decreases in a stairstep fashion on either side. Classes 1 and 6 have the lowest membership values (zero, in this case).

1.5.6.1.1.4. *Nominal Function*

Nominal or categorical data do not have quantitative meaning. Values are only for labeling or categorizing different things. A typical example is geologic data. Technically, ArcSIE requires nominal data to be represented by numerical values, particularly integers. With numerical representation, ArcSIE uses the following function for nominal data:

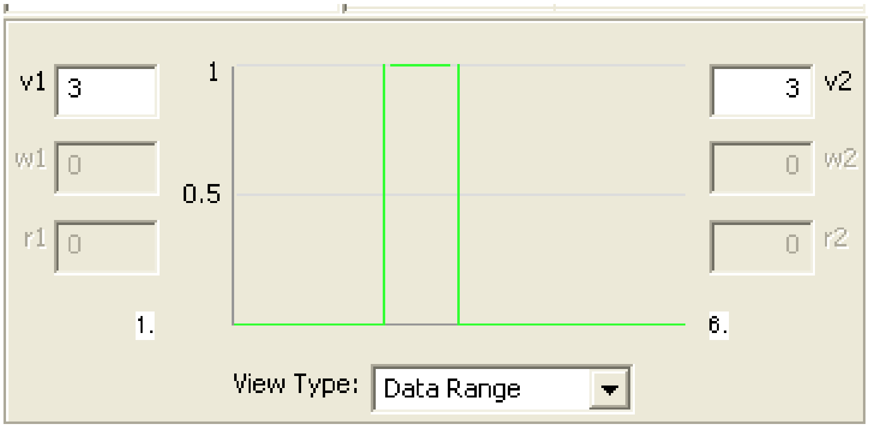
$$\begin{cases} s_{ij,k,a} = \max & \text{\textit{if} } v_1 =< z_{ij,a} =< v_2 \\ s_{ij,k,a} = 0 & \text{\textit{otherwise}} \end{cases}$$

The symbols in the equation group are explained as follows:

Symbol	**Meaning**
$s_{ij,k,a}$	The optimality value of *environmental feature a* at location (*i*, *j*) for soil type *k*. This is the output from the *E* function.
max	The maximum fuzzy membership the user specifies in  Membership Value: 1 . This value is typically 1.

$z_{ij,a}$	The value of *environmental feature a* at location (*i*, *j*). This value is read from the *environmental database*.
v_1 and v_2	These are two user-specified central values that define the lower and upper limits of the *most optimal range* of *environmental feature a* for soil *k*. In other words, if $z_{ij,a}$ falls between v_1 and v_2, then location (*i*, *j*) will get the maximum optimality value for soil *k* in terms of *environmental feature a*. Technically, v_1 and v_2 determine the width of the flat top of the function curve. These two values can be adjusted by editing the values in v1 0 and 0 v2. **It is required that v_1 is less than or equal to v_2.**

Due to the nature of nominal data, the nominal function assumes the values for this *environmental feature* to be integers, and v_1 and v_2 should also be integers and usually $v_1 = v_2$ (only one optimal type). If, however, $v_1 \neq v_2$, it is required that v_1 is less than v_2 and any types whose values are between v_1 and v_2 will be included into the optimal range. Below is an example illustrating the use of the nominal function:



v1 3

w1 0

r1 0

v2 3

w2 0

r2 0

View Type: Data Range

In the example illustrated by the above graphic, the user defines a nominal function for a geology layer, in which different bedrock types are represented by numbers 1–6. The user specifies that bedrock type 3 to be the most optimal one. All the other bedrock types have the lowest membership values (zero, in this case). This specifies that the current soil type only occurs within the area of bedrock type 3.

1.5.6.1.1.5. Default Function

The default function used by ArcSIE is a *continuous bell-shape* function with parameter values as follows:

- $v_1 = v_2$ = mean of the values in the *environmental database* for the current *environmental feature*
- $w_1 = w_2 = 0.2$ * standard deviation of the values in the *environmental database* for the current *environmental feature*
- $r_1 = r_2 = 2$

1.5.6.1.1.6. Use Raw Values

This option is typically used for data layers that are output from a previous inference; i.e., values already represent optimality. Because there is no need to convert these values again, the *Raw* option simply uses the original layer values as the optimality values. This provides you with an easy way to include previous models into a new inference.

Note: The *Raw* option expects all values to be between 1 and 100, so values between 0 and 1 will not output correctly. To avoid this problem, you may wish to convert the layer to integer values prior to including it in your *environmental database*. Input values >100 will not occur if you are using output from an SIE inference.

1.5.6.1.2. Three Ways to View the Function Curve

The function curve of E can be viewed in three ways:

- *Data Range*: Zooms the view to the entire data range (from the minimum data value to the maximum data value) for the current *environmental feature*.

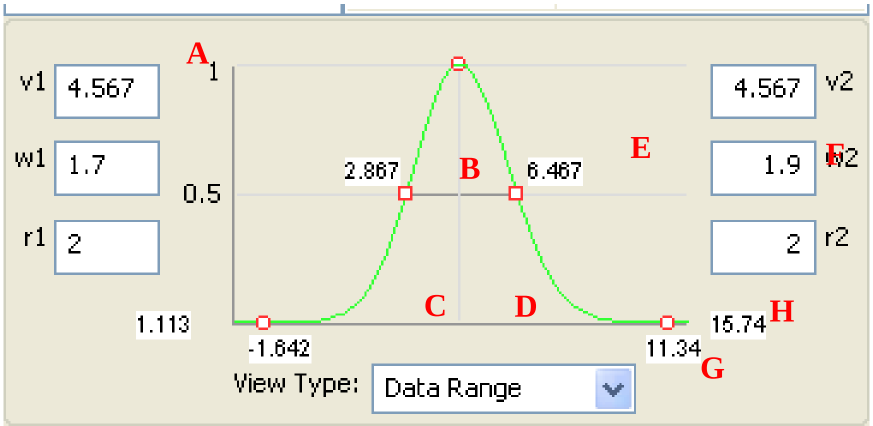
- *Relative*: Zooms the view to the entire function curve.

- *Standard Deviation*: Zooms the view to the range of four standard deviations of the data for the current *environmental feature* (two standard deviations on each side of the function curve). This option brings a scale bar to the x axis, with each interval representing the width of one standard deviation of the data.

You can switch among these three options in the Function Defining Area, using the dropdown menu View Type: Data Range.

1.5.6.1.3. Values Displayed on the Function Curve

Once you *check on* an *environmental feature*, a green curve appears in the *E* function defining area on the Inference dialog box. This green curve is a graphic representation of the *E* function for the current *environmental feature*. There are some small red handles attached to the curve. The two round handles on the top of the curve correspond to v_1 and v_2; the two round handles at the bottom of the curve correspond to the *environmental feature* values giving very small optimality values (0.0001); and the two square handles on the sides of the curve indicate the positions where c_1 and c_2 equal to 0.5, respectively. On the image in next page, the values associated with the function curve are labeled by letters A – H and are explained in the following table:


v1 4.567
w1 1.7
r1 2
v2 4.567
w2 1.9
r2 2
View Type: Data Range

Letter	Meaning
A	The maximum optimality the *rule* can generate. The default value is 1.
B	This is the c_1 and c_2 values for the *E* equation (see 1.5.6.1.1.1). To simplify the parameter setting, for c_1 and c_2 ArcSIE always uses the middle value between the maximum and the minimum optimality values that the *rule* can generate. Since the default maximum optimality value is 1, the default middle optimality value is 0.5.
C	The minimum value in the data of the current *environmental feature*. This value only shows up under the "Data Range" option.
D	The value of the *environmental feature* at the left bottom red handle. This is the "left" (smaller) value of the *environmental feature* that gives a very small optimality value (set to be 0.0001).
E	The value of the *environmental feature* at the left middle red handle. This is the "left" (smaller) value of the *environmental feature* that gives the "middle" optimality value. E = $v_1 - w_1$.
F	The value of the *environmental feature* at the right middle red handle. This is the "right" (larger) value of the *environmental feature* that gives the "middle" optimality value. F = $v_2 + w_2$.
G	The value of the *environmental feature* at the right bottom red handle. This is the "right" (larger) value of the *environmental feature* that gives a very small optimality value (set to be 0.0001).
H	The maximum value in the data for the current *environmental feature*. This value only shows up under the "Data Range" option.

1.5.6.1.4. Adjusting the Function Curve

1.5.6.1.4.1. Adjusting the Curve Using the Input Fields

You can directly type in values for v_1, w_1, r_1, v_2, w_2, and r_2 in their corresponding input fields to adjust shape of the function curve. ArcSIE uses a fixed value, 0.5×the value in (in the Inference dialog box), for both c_1 and c_2, so there are no input fields for them.

Membership Value: 1

1.5.6.1.4.2. Adjusting the Curve Using the Graphic Tool

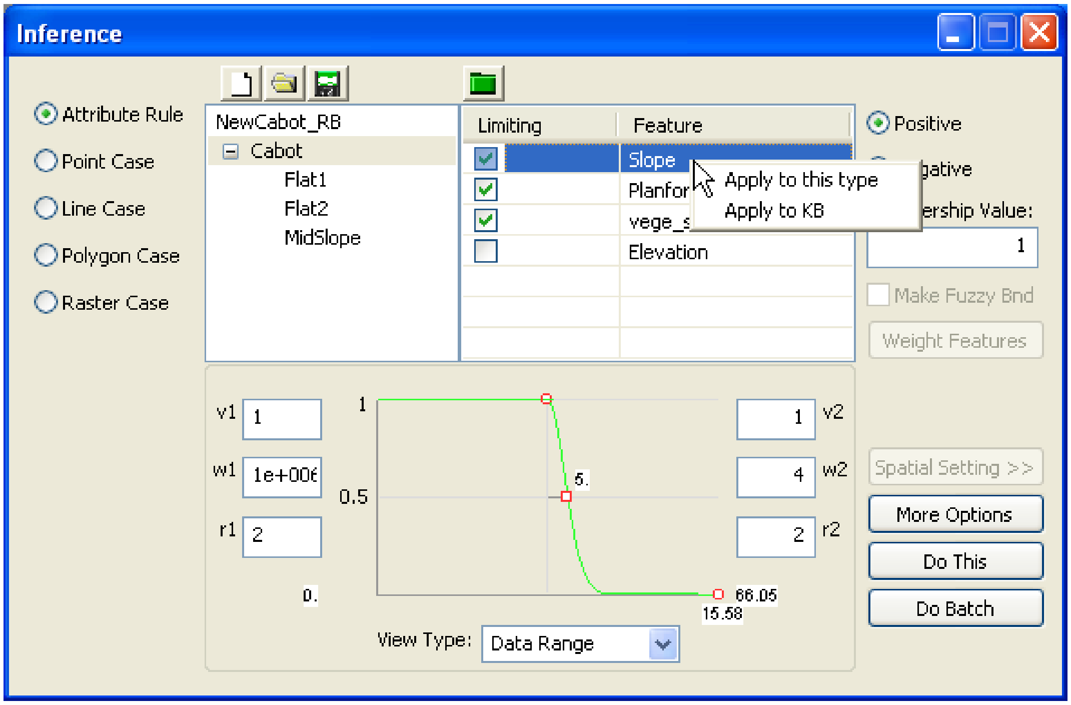
For the *continuous* and *cyclic* functions, you can use the mouse to adjust the shape of the curve in a *click-and-drag* manner. To adjust the curve, put the cursor close enough to either of the two red square handles on the sides of the green curve, click, hold, and drag. Dragging a handle also causes the value in the corresponding *w* field and the "middle" and "bottom" values on the curve graphic to change accordingly.

You cannot use the graphic tool to adjust the *ordinal* and *nominal* functions. To make changes to these functions, you must explicitly type in values for v_1 and v_2 in their corresponding input fields.

1.5.6.1.5. Applying an E Function to all the Instances of a Soil Type or Rulebase

Right-clicking an *environmental feature* in the *environmental feature* list of the Knowledge Explorer opens a pop-up menu (see graphic in next page). This menu allows you to apply the setting of the *E* function for the selected *feature* of the current *instance* to the counterpart *features* in all the *instances* in the current *soil type* (*Apply to this type*) or even the entire *rulebase (Apply to KB).*

When you apply the setting for a feature in the current *instance* to other *instances*, only the values of *w* and *r* in the corresponding *features* of other *instances* will be modified. The original values of v_1 and v_2 of those *instances* will not change.



Inference

● Attribute Rule
○ Point Case
○ Line Case
○ Polygon Case
○ Raster Case

NewCabot_RB
⊟ Cabot
Flat1
Flat2
MidSlope

Limiting	Feature
☑	Slope
☑	Planfor
☑	vege_s
□	Elevation

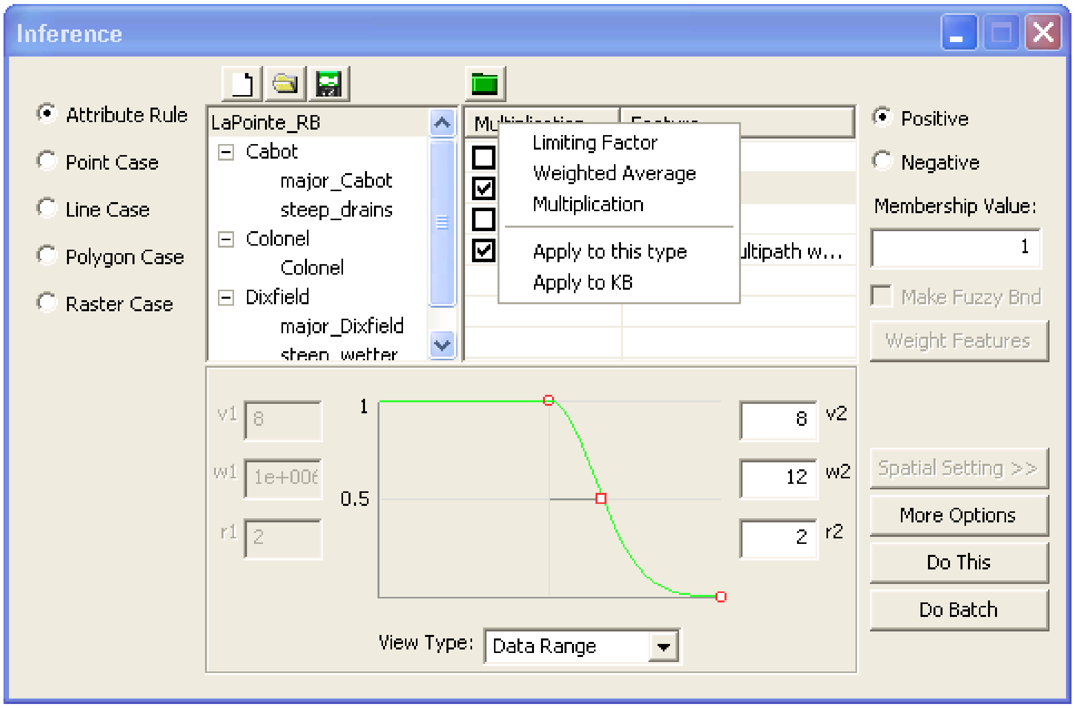
● Positive
○ ...gative
...ership Value: 1
□ Make Fuzzy Bnd

v1: 1
w1: 1e+00€
r1: 2
v2: 1
w2: 4
r2: 2

View Type: Data Range

1.5.6.2. Adjust the *P* Function

The *P* in the general inference equation (see 1.5.6) integrates the optimality values based on individual *environmental features* and generate an optimality value for the whole *instance*. ArcSIE implements three methods for the *P* function: Limiting-Factor, Weighted-Average, and Multiplication. Right-click on the header of the first column in the *feature* list of the Inference dialogbox, and a menu will appear for you to select one of the three methods.



● Attribute Rule
○ Point Case
○ Line Case
○ Polygon Case
○ Raster Case

● Positive
○ Negative

Membership Value: 1

□ Make Fuzzy Bnd

v1: 8
w1: 1e+006
r1: 2
v2: 8
w2: 12
r2: 2

View Type: Data Range

Once you selected a method for the current *instance*, using the two "Apply to …" options in the same menu, you can specify to apply the selected method for the P function to all the *instances* in the current *soil type* or all the *instances* in the entire *rulebase*.

1.5.6.2.1. Limiting-Factor

For the limiting factor method, ArcSIE chooses the minimum among all the optimality values of individual *environmental features* as the overall optimality value of the whole *instance*. The theoretical basis for this option is the limiting-factor principle in ecology.

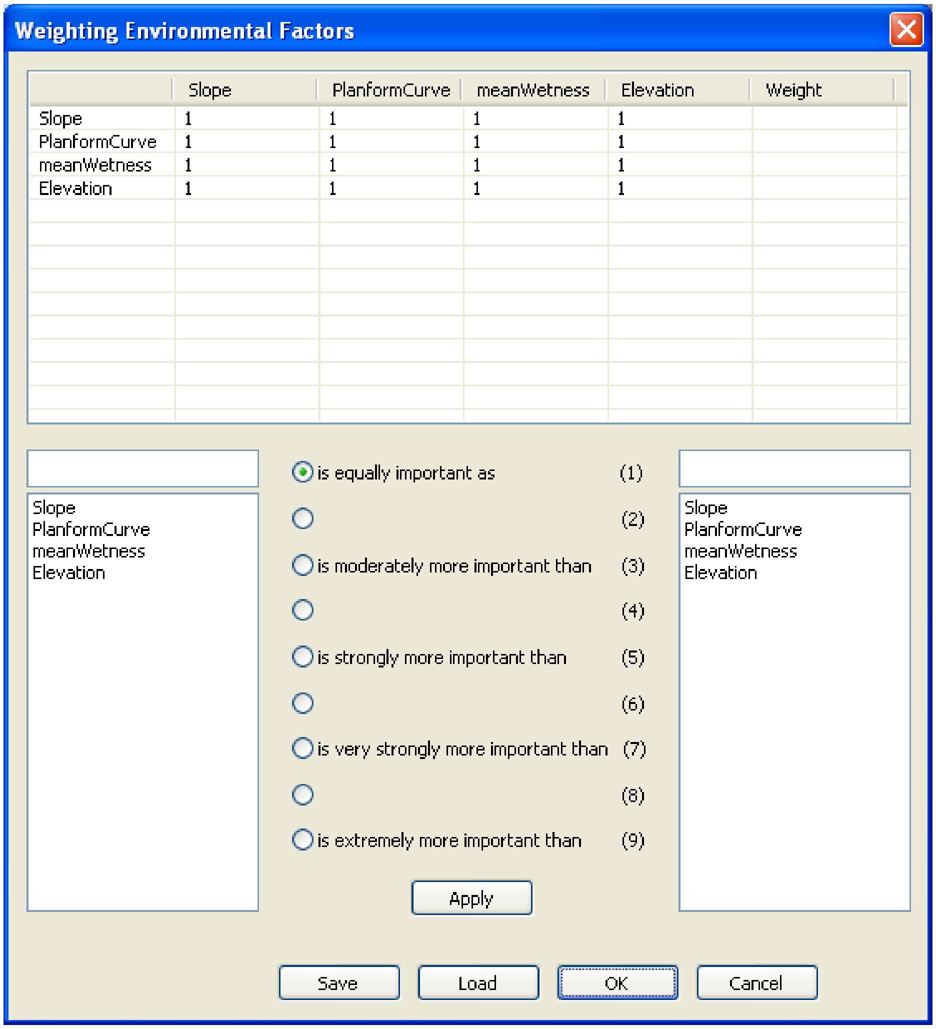
If you choose to use the limiting-factor method, the appearance of the *feature* list window will change accordingly allowing you to specify if an *environmental feature* in the current *feature* list should or should not to be used in the inference. You do this by checking on or off an individual *feature*.

1.5.6.2.2. Weighted-Average

Under the Weighted-average method, ArcSIE calculates a linear weighted average of the optimality values of individual *environmental features* to get the overall optimality value for the *instance*. If you choose to use this method, the interface of the *feature* list will change accordingly and allow you to assign a weight to each *feature*. ArcSIE provides two ways for you to specify weights:

- directly typing in weight values in the *feature* list window.
- using the Analytic Hierarchy Process (AHP) tool to determine the weights in a structured way.

When you choose to use the weighted average method, Weight Features in the Inference dialog box will become available. Pressing this button brings the AHP interface to the front:



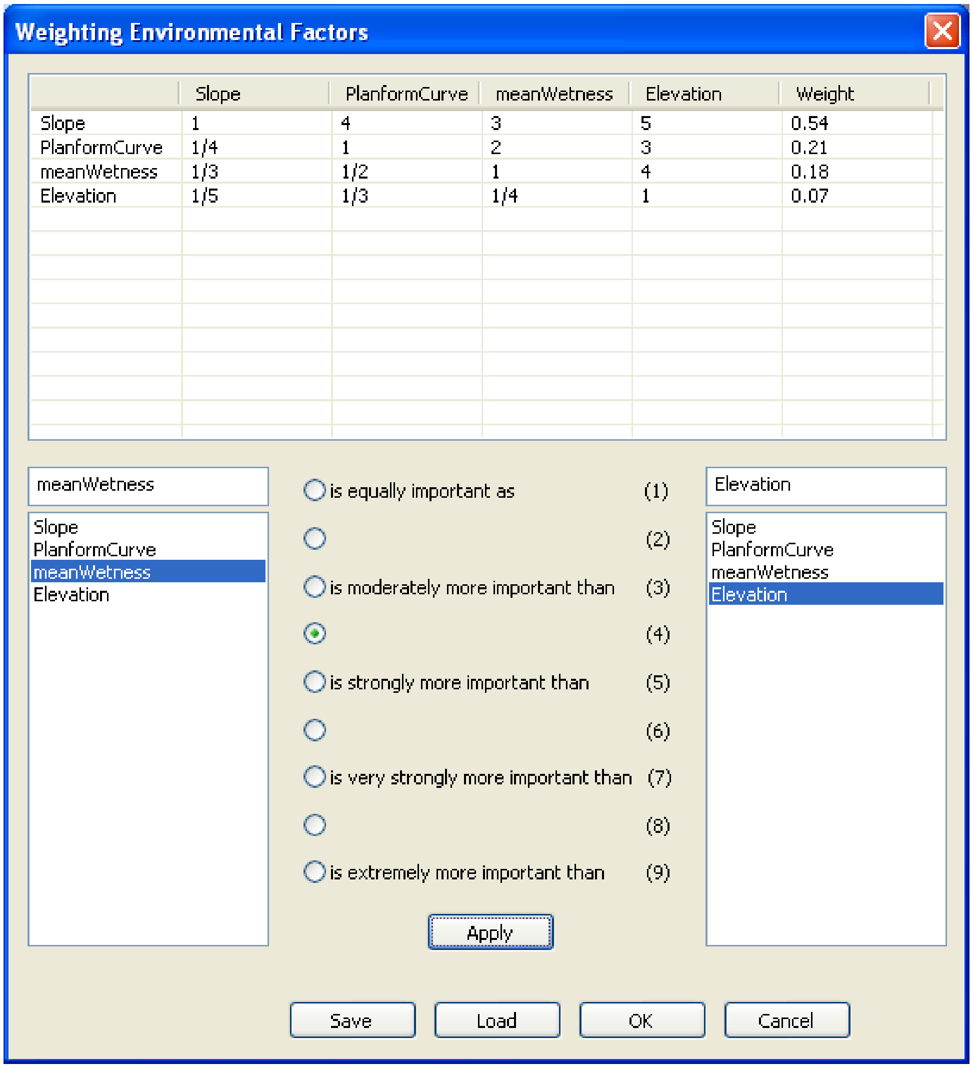
	Slope	PlanformCurve	meanWetness	Elevation	Weight
Slope	1	1	1	1	
PlanformCurve	1	1	1	1	
meanWetness	1	1	1	1	
Elevation	1	1	1	1	

Slope
PlanformCurve
meanWetness
Elevation

● is equally important as (1)
○ (2)
○ is moderately more important than (3)
○ (4)
○ is strongly more important than (5)
○ (6)
○ is very strongly more important than (7)
○ (8)
○ is extremely more important than (9)

Slope
PlanformCurve
meanWetness
Elevation

The AHP method contains two basic steps: first, the user conducts pair-wise comparisons on the *features* using a nine-score scale, and second, the computer performs matrix calculations using the score matrix to determine the weight for each *feature*. Referring to the image in the next page, for example, you can specify that "slope is strongly more important than elevation". The computer will then quantify this specification by assigning the score of slope against elevation as 5, and the score of elevation against slope as 1/5. In the same way, you compare all the pairs that can be formed for the *environmental features* in the *current environmental database*. When you have all pairs completed, the weights of all the *environmental features* will be calculated by the computer:



	Slope	PlanformCurve	meanWetness	Elevation	Weight
Slope	1	4	3	5	0.54
PlanformCurve	1/4	1	2	3	0.21
meanWetness	1/3	1/2	1	4	0.18
Elevation	1/5	1/3	1/4	1	0.07

meanWetness

Slope
PlanformCurve
meanWetness
Elevation

○ is equally important as (1)
○ (2)
○ is moderately more important than (3)
● (4)
○ is strongly more important than (5)
○ (6)
○ is very strongly more important than (7)
○ (8)
○ is extremely more important than (9)

Elevation

Slope
PlanformCurve
meanWetness
Elevation

To conduct the comparison using the AHP tool in ArcSIE, click an *environmental feature* in the left list box to select it; select a score from the score list, referring to the natural language descriptions of the scores; select the other environmental feature from the right list box; and then click the "Apply" button. The result of the operation will appear in the list window. The design of the tool requires you to always select the "stronger" *environmental feature* in a pair from the left list box.

You can save the AHP matrix you create into an AHP file by clicking the "Save" button and "Load" it in another time.

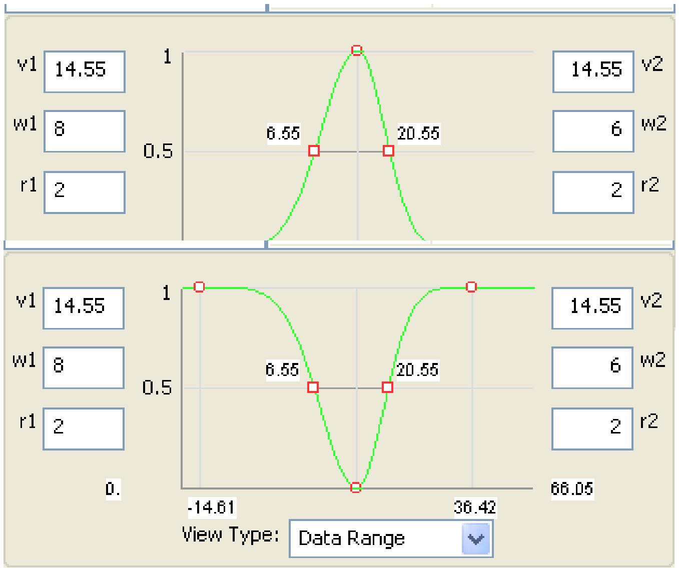
When you are done, click the "OK" button and return to the Inference dialog box. The weights will appear in the *feature* list of the Knowledge Explorer in the Inference dialog box.

1.5.6.2.3. Multiplication

The *Multiplication* method calculates the product of the values from the individual *environmental features* and uses it as the overall output value from the whole instance.

The *Multiplication* method is useful when you want to represent interactions among *environmental features* or when you want to mask an inference result.

1.5.6.3. *Positive* Instance vs. *Negative* Instance

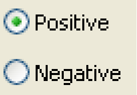


View Type: Data Range

An instance may be *positive* or *negative*. A *positive* instance defines how the optimality value decreases as the *environmental feature* value deviates from the most optimal condition. A typical *positive* function curve is shown below:

A *negative* instance defines how the optimality value increases as the *environmental feature* value deviates from the least optimal condition. A typical negative function is shown below:

By default, an instance is *positive*. You can specify an instance to be *positive* or *negative* by toggling between the two radio buttons ● Positive ○ Negative.



A *Soil Type* can contain either *positive* or *negative* instances or both.

- If it contains only *positive* instances, the results from individual instances will be integrated (the *T* function in 1.3.5.2) using the MAX operation.

- If it contains only *negative* instances, the results from individual instances will be integrated using the MIN operation.

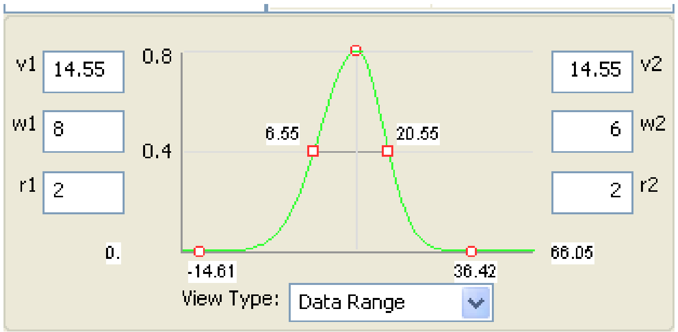
- If it contains both, the inference engine will first separate the two types of instances and run inferences with them separately. The results from the two types of instances will then be integrated using the MIN operation.

1.5.6.4. Partial Membership Instance

By default, the maximum optimality defined by a *positive* instance for the most optimal condition is 1, the full membership; and the minimum optimality defined by a *negative* instance for the least optimal condition is 0. You can specify other values for the maximum/minimum membership for an instance in Membership Value: 0.8 . The value you specify must be between 0 and 1.

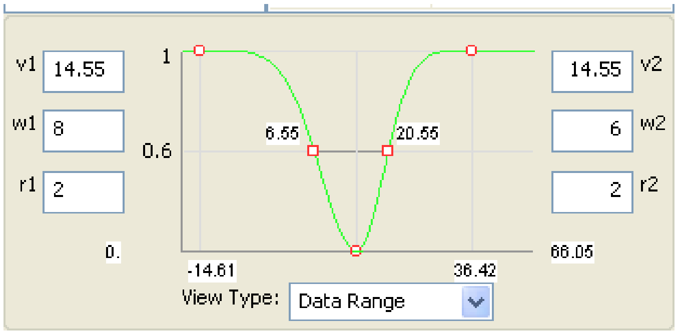
- The optimality value given by a *positive* instance will vary between 0 and the maximum value you specify.
- The optimality value given by a *negative* instance will vary between the minimum value you specify and 1.

The mark values on the vertical axis of the curve plot will change in accordance with the value you specify. For example, if you specify the maximum value for a *positive* instance is 0.8, the mark values will be as follows, where 0.4 is the middle value between 0 and 0.8.



v1 14.55
w1 8
r1 2
v2 14.55
w2 6
r2 2
View Type: Data Range

If you specify 0.2 as the minimum value for a *negative* instance, the mark values will be as follows, where 0.6 is the middle value between 0.2 and 1.



v1 14.55
w1 8
r1 2
v2 14.55
w2 6
r2 2
View Type: Data Range

1.5.6.5. Turning on/off an *Environmental Feature*

When the P function is *Limiting Factor* or *Multiplication*, you can turn on or turn off an *environmental feature* by checking the small box before the *feature* name in the *feature* list window.

If you turn off a *feature*, the *feature* will not be used in the inference and the function curve defining area will be grayed (disabled) for this *feature*.

If you turn off all the *features* of an *instance*, you eventually turn off the *instance*, i.e., the *instance* will not be used in the inference.

When you load a new *environmental database* or add new environmental data layers, the default state of the corresponding new *environmental features* is *off*.

1.6. Preparing a Vector (Point, Line, or Polygon) *Casebase*

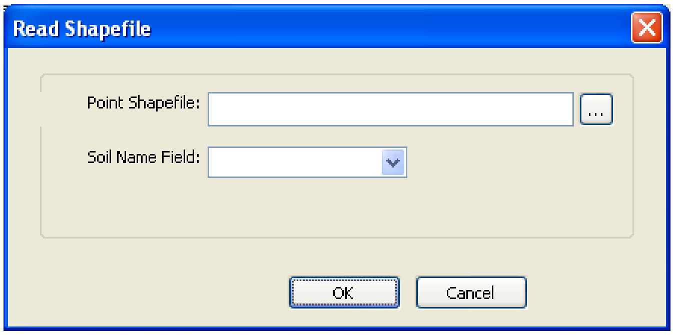
1.6.1. Creating a New Vector *Casebase*

Before a new *casebase* can be created, an *environmental database* must have been loaded (see 1.3). The loaded *environmental database* determines what *environmental features* will be used to characterize the new *cases*.

However, you do not have to re-build/re-load an *environmental database* every time you switch between knowledge types or forms (e.g., from "Attribute Rules" to "Point Case" or from "Line Case" to "Polygon Case"). The current *environmental database* can also be applied to the knowledge type or form you switch to.

ArcSIE uses a Shapefile to create a new vector *casebase*.

To create a new vector *casebase*, click 🗋. The Read Shapefile dialog box will appear for you to select the Shapefile. You also must select the field in the attribute table of the Shapefile containing the soil type names. The data type of the soil type field must be "string". The values in this field will be used to create *soil types* in the *casebase*. ArcSIE automatically checks the data types of all the fields in the attribute table and lists those fields whose types are "string" as candidates for you to select.

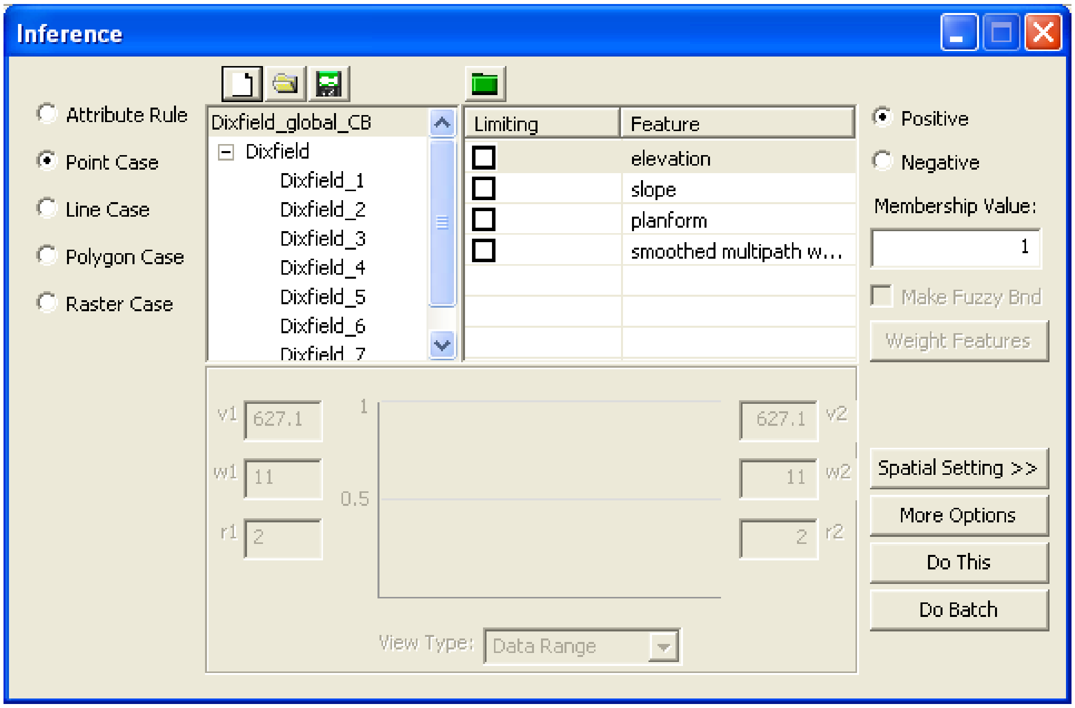


You CANNOT create new *soil types* or new *cases* for a *casebase* in ArcSIE. *Cases* and the *soil types* they belong to can only be created in a GIS, e.g., 3dMapper or ArcGIS.

At any time, ArcSIE works with only one *point casebase*, one *line casebase*, and one *polygon casebase*. Therefore, for example, if there is a *point casebase* currently loaded, it will be replaced by the new *point casebase* you are creating. You will be prompted to save the changes you made to the old *casebase* before it is unloaded.

In a vector *casebase*, all the *cases* for one soil type are organized into a *soil type* named after that soil type.

The default names for the *soil type* and *cases* are the soil names in the selected field from the Shapefile's attribute table. The name of a *case* is created by adding an integer to the soil name to make the *case*'s name unique. The screenshot in the next page shows an example of a point *casebase* created from a point Shapefile.



○ Attribute Rule
● Point Case
○ Line Case
○ Polygon Case
○ Raster Case

Dixfield_global_CB
⊟ Dixfield
Dixfield_1
Dixfield_2
Dixfield_3
Dixfield_4
Dixfield_5
Dixfield_6
Dixfield_7

Limiting	Feature
□	elevation
□	slope
□	planform
□	smoothed multipath w...

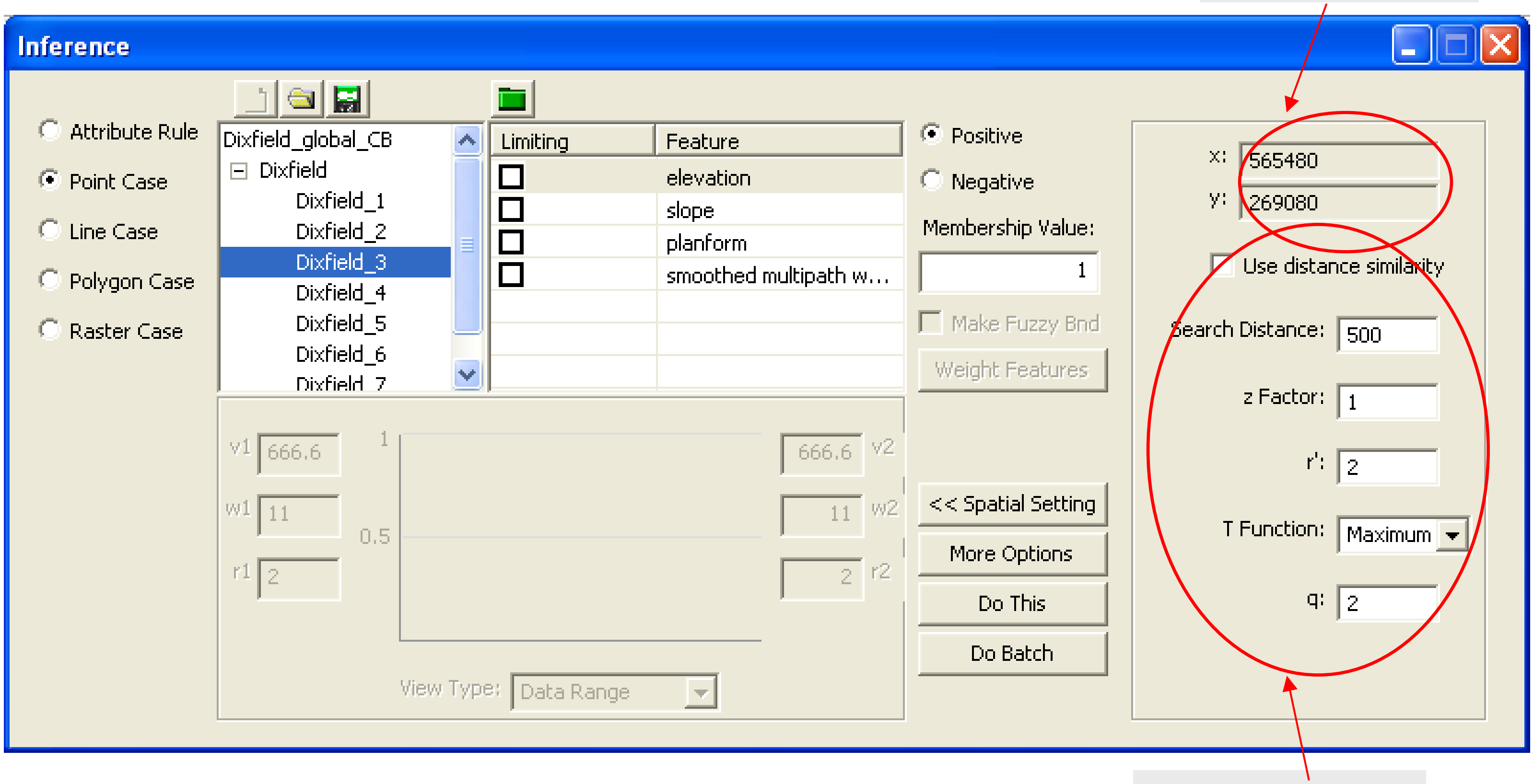
● Positive
○ Negative
Membership Value: 1
□ Make Fuzzy Bnd

v1 627.1 v2 627.1
w1 11 w2 11
r1 2 r2 2
View Type: Data Range

Technically, a *case* is simply an *instance* in a *rulebase* plus a *Spatial Setting*. The above screenshot shows the *rule* part of the *casebase*, and it looks exactly the same as a *rulebase*. Clicking on Spatial Setting >> opens the dialog box for *Spatial Setting* of the *casebase*, as shown in the screenshot in the next page. Clicking on << Spatial Setting again will close the dialog box for *Spatial Setting*.

The parameters of the *Spatial Setting* of a *case* are explained in 1.6.6.

Coordinates (location) of the currently selected *case*.



Inference

○ Attribute Rule
● Point Case
○ Line Case
○ Polygon Case
○ Raster Case

Dixfield_global_CB
⊟ Dixfield
Dixfield_1
Dixfield_2
Dixfield_3
Dixfield_4
Dixfield_5
Dixfield_6
Dixfield_7

Limiting	Feature
□	elevation
□	slope
□	planform
□	smoothed multipath w…

● Positive
○ Negative
Membership Value: 1
□ Make Fuzzy Bnd

x: 565480
y: 269080
□ Use distance similarity
Search Distance: 500
z Factor: 1
r': 2
T Function: Maximum
q: 2

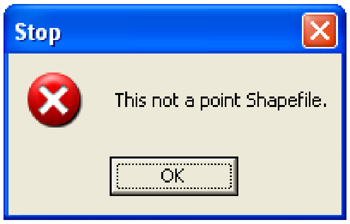
v1 666.6
w1 11
r1 2
v2 666.6
w2 11
r2 2
View Type: Data Range

Parameters for the *local case-based reasoning*.

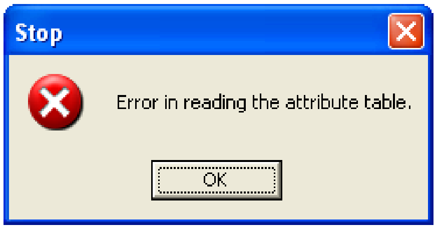
1.6.2. Loading an Existing Vector *Casebase*

Loading an existing vector *casebase* is similar to loading an existing *rulebase* (see 1.5.2), except that the file format for a *casebase* is Shapefile instead of DBF table. A Shapefile of *casebase* contains some special attribute fields added by ArcSIE.

If the type of the specified Shapefile does not match the specified *case* form, an error message will appear and the loading process will abort. For example, if you try to load a line Shapefile as a point *casebase*, you will get an error message as the right.



This not a point Shapefile.



Error in reading the attribute table.

If the required attribute fields for a *casebase* are not found in the attribute table of the Shapefile, you will get an error message as the left and the loading process will abort.

1.6.3. Saving Vector *Casebase*

Selecting the *casebase* name and clicking on 💾 saves the current vector *casebase* into a Shapefile. The information about the *soil types* and *cases* (both *rule* part and spatial setting) is saved into the attribute table of this Shapefile.

1.6.4. Unloading Vector *Casebase*

To unload the current vector *casebase* from ArcSIE, right-click on the *casebase* name and select *Remove*.

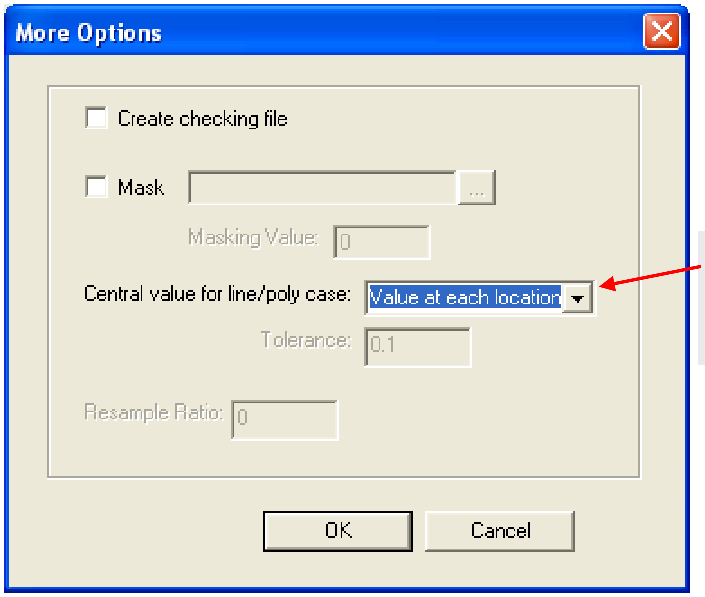
Unlike with the *instances* in a *rulebase*, you CANNOT remove individual *soil types* or *cases* from a vector *casebase*. You can only do this in a GIS, e.g., 3dMapper or ArcGIS.

1.6.5. Editing the *Rule* Part of a *Case*

Editing the *rule* part of a vector case is exactly the same as editing an *instance*. See 1.5.6.

For a *point case*, the default value of v_1 and v_2 of a *case* is the *environmental feature* value at the location of the *case* (The location is indicated by x and y). You can modify v_1 and v_2. Whenever necessary, you can reset them to the default value by choosing "Default" in the right-click pop-up menu.

For a *line case*, the *x* and *y* displayed in the Inference dialog box are coordinates of one of the "points" along the line (a line is considered as a series of points). The v_1 and v_2 fields will be grayed and the user will not be able to type in arbitrary values for these two parameters. For a *line case*, v_1 always equals v_2, and the value is either the value of an individual cell or a statistic of the values of those cells passed through by the *line case*. The user can specify which value to use in the More Options dialog box (shown below). To open the More Options dialog box, click on More Options in the Inference box.



□ Create checking file

□ Mask

Masking Value: 0

Central value for line/poly case: Value at each location

Tolerance: 0.1

Resample Ratio: 0

Set v_1 and v_2 for a *line case* or *polygon case.*

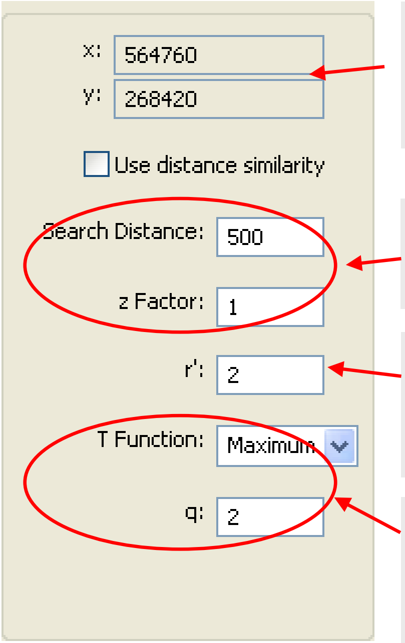
See 1.8.3.3 for a detailed explanation of the options for v_1 and v_2 of a *line case*.

For a *polygon case*, the *x* and *y* displayed in the dialog box are coordinates of one of the "points" in the polygon (a polygon is considered as an agglomeration of points). The v_1 and v_2 fields are grayed and the user will not be able to type in arbitrary values for these two parameters. For a *polygon case*, v_1 always equals v_2 and the value is either the value of an individual cell or a statistic of the values of those cells falling into the *polygon case*. The user can specify which value to use in the More Options dialog box (see previous page).

Please see 1.8.3.3 for a detailed explanation of the options for v_1 and v_2 of a *polygon case*.

1.6.6. Editing the Spatial Setting of a *Case*

The *Spatial Setting* of a case includes the geographical location of a *case*, its *Influence Region*, and other specifications for adjusting the membership values calculated based on *environmental features*.



x: 564760

y: 268420

□ Use distance similarity

Search Distance: 500

z Factor: 1

r': 2

T Function: Maximum

q: 2

The location of a *case* is specified by the user when creating the *case* in a GIS, and cannot be modified in ArcSIE.

These two parameters define the *Influence Region* of the currently selected *case*.

When "Use distance similarity" is on, this parameter determines how to adjust fuzzy membership based on distance.

These two parameters define the *T* function that integrates membership values from multiple *cases*.

1.6.6.1. *Global Case* vs. *Local Case*

Global CBR and *local CBR* are distinguished by whether the *cases* are applied to the entire mapping area or applied to only a limited region (*Influence Region*). *Cases* used for these two types of CBRs are called *global cases* and *local cases*, respectively.

A *global case* is considered exactly the same as an *instance* in RBR, except that the *central values* of the case are not directly specified by the user, but are retrieved by the inference engine from the environmental data layers according to the *case's* location. A *local case* contains certain parameter values for the *local* CBR, which makes it somewhat different from an *instance*.

Using Spatial Setting >>, you can toggle between using and not using a spatial setting, i.e., make the *cases global* or *local*. When the spatial setting panel of the Inference dialog box is open, the values of the parameters displayed in the panel will be applied to the currently selected *case* and the *case* becomes a *local case*. If you close the spatial setting panel, the *cases* in the current *casebase* become *global cases*, meaning that they do not have *influence regions* associated with them and during the inference they will be applied to the entire mapping area.

ArcSIE does not allow a mixed use of *global cases* and *local cases*. Whether the spatial setting panel (see below) is open or closed when you start the inference determines whether the *cases* in the current *casebase* are used as *global cases* or *local cases* in the inference. If the panel is open, all the *cases* will be used as *local cases*; and if closed, all the *cases* will be used as *global cases*.

For *local cases*, you must specify three spatial settings: *Influence Region* (defined by the *Search Distance* and *z Factor*), the way to adjust optimality based on distance (defined by the *Use similarity distance* toggle and the *r'*), and *T Function* (which includes a user-specified decay factor, *q*).

1.6.6.2. *Influence Region*

Influence Region defines the spatial extent influenced by a *case*.

For a *point case*, the *Influence Region* is defined around the point of the *case*.

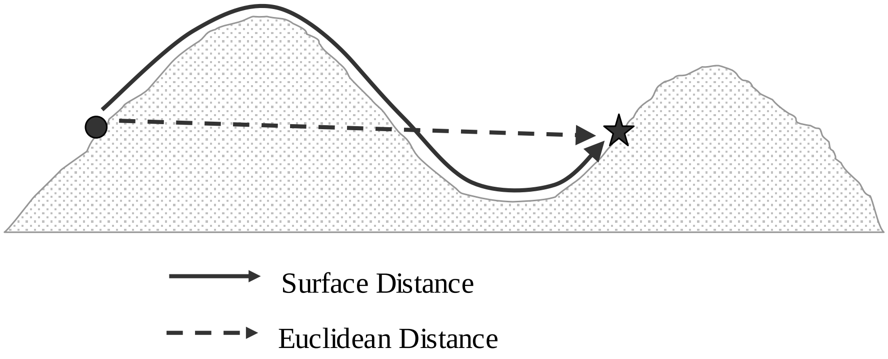
For a *line case*, the *Influence Region* is defined for each "point" (cell) passed through by the line.

For a *polygon case*, the *Influence Region* is defined for each "point" (cell) within the polygon.

The *Influence Region* is defined by two parameters: *Search Distance* and *z Factor*.

- *Search Distance*

Search Distance is the distance for defining a neighborhood around a *case* location. In ArcSIE, this distance is not Euclidean but calculated along the terrain surface, as illustrated by the figure below:



Measuring the distance in this way may result in an irregular-shaped *Influence Region*.

The unit of the *Search Distance* is the unit used by the *environmental database*.

- *z Factor*

The *z* factor adjusts the importance of the vertical variation (i.e., change of elevation) in calculating the *Search Distance*. A value > 1 for the *z Factor* exaggerates the importance of vertical variation. A *z Value* between 0 and 1 understates the importance.

1.6.6.3. The Way to Adjust Optimality Value Based on Distance

You can use distance to adjust the optimality value calculated by the *rule* part of a *case*. The basic idea is if a location is geographically closer to a *case*, it should be more similar to that *case*. The function of this adjustment is shown below and the symbols are explained in the following table:

$$s'_{ij,t} = s_{ij,t}\, e^{\left(d_{ij,t} / w'_{t}\right)^{r'} \ln(0.0001)}$$

Symbol	*Meaning*
$s_{ij,}{}'_{t}$	The distance-adjusted optimality value at location (*i,j*) calculated based on case *t*.
$s_{ij,t}$	The optimality value calculated using the *rule* part of case *t* (i.e., the output from function *P* in the general inference equation described in 1.5.6).
e	The base of the natural logarithm (2.71828 …).
$d_{ij,t}$	The *surface distance* between the location (*i*, *j*) and *case t*.
w_t’	The *Search Distance* of case *t*.
r’	Performs similar role as r_1 and r_2 in the continuous membership function (see 1.5.6.1.1.1).
ln	The natural logarithm.

When location (*i*, *j*) is within the *Search Distance* of *case t*, the value of the *e* part of the above equation varies between 0.0001 and 1 to make continuous adjustment of $s_{ij,t}$

The equation above defines a function curve by specifying that if the distance between a location and case *t* is the *Search Distance*, that location has a very small similarity value to *t* (0.0001)

Check □ Use distance similarity to specify that you want to perform this adjustment. To perform this adjustment, you must specify values for w_t’ (the *Search Distance*, discussed in 1.6.6.2) and *r’*.

1.6.6.4. *T Function*

Referring to the general inference equation (see 1.5.6), in CBR, the *T* function is used to integrate information from multiple *cases*. The *T* function is particularly important in local CBR, as a location can fall into the influence regions of multiple *cases* and the soil scientist may consider more complicated relationships among the *cases* than that in global CBR. The options implemented by ArcSIE for *T* are listed in the table on the next page.

Category	***Algorithm***	***Description***	***Priority***
Using a single value	*Dominant*	Always use the value calculated based on this *case* no matter what values from other *cases* are. This indicates that the current *case* has an exclusive influence on the locations within the *case*'s *Influence Region*.	1
	Maximum/Minimum	Among the values calculated based on different *cases*, use the maximum/minimum one. This indicates that the most similar *case* has an exclusive influence. For *positive* cases, use *maximum* operation; and for *negative* cases, use *minimum* operation.	2
	Nearest	Use the value calculated based on this *case* if this *case*, among all the cases whose influence regions cover the location under consideration, is geographically nearest to that location. This indicates that the nearest *case* has an exclusive effect.	3
	Supplement	Use the calculated value based on this *case* only when no other *cases* can provide non-zero values to the location under consideration.	1.5
Using multiple values	*Similarity-weighted*	Use the weighted average of the values calculated based on different *cases*, in which the weights are the values themselves. This indicates that a more similar *case* should have more influence.	4
	Inverse-distance	Use the weighted average of the values calculated based on different *cases*, in which the weights are inverses of the *surface distances* between the location under consideration and the *cases*. This indicates that a closer *case* should have more influence.	5
	Average	Use the simple average of the values calculated based on different *cases*.	6

Different functions have different priorities. If a location is within the *Influence Region* of multiple *cases* and these *cases* have been assigned different *T functions*, the *case* whose function has the highest priority will determine how to integrate the values calculated from different *cases*. The *Supplement* function will have effect only when the values from all the other cases are zero, therefore it can be considered to have conditional highest priority. When the functions of two cases have the same priority, the case that is more similar to the given location has a higher priority.

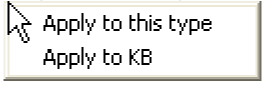
Both *Similarity-weighted* and *Inverse-distance* functions calculate weighted average, thus the general mathematical representation of both functions is:

$$s_{ij,k} = \frac{\sum_{t=1}^{n} (d_{ij,t})^{-q} s_{ij,t}}{\sum_{t=1}^{n} (d_{ij,t})^{-q}}$$

The symbols in this equation are explained the table below:

Symbol	***Meaning***
$s_{ij,k}$	The final optimality value at (*i, j*) for soil *k* (i.e., the output from function *T*).
$s_{ij,t}$	The optimality value calculated based on *case t*. This can be $s_{ij,}{}'_{t}$ from the adjustment described in 1.6.6.3.
n	The number of cases whose influence regions cover location (*i, j*).
$d_{ij,t}$	For the *Similarity-weighted* function, this is $s_{ij,t}$ again; for the *Inverse-distance* function, this is the *surface distance* between (*i, j*) and *t*.
q	The distance decay factor (for the *Similarity-weighted* function, "distance" means the similarity). The greater this value, the more important the distance is, i.e., the importance of a distant *case* drops faster. The default value for *q* is 2, defining a quadratic dropping curve.

1.6.6.5. Applying the Current Spatial Setting to all the *Cases* in a *Soil Type* or *Casebase*

Right-clicking in the spatial-setting area opens a pop-up menu 

and allows you to apply the spatial setting of the current *case* to all the *cases* in the current *soil type* or even the entire *casebase* (knowledgebase).

1.6.7. *Positive* Case vs. *Negative* Case

A *case* can be *positive* and *negative*. The way to define and work with *positive* and *negative cases* is the same as that of defining *positive* and *negative instances* (See 1.5.6.3.).

1.6.8. Partial Membership Case

If you pinpoint or delineate typical locations to create cases, those cases are usually assigned full fuzzy membership for the soil they represent. ArcSIE can also work with cases at non-typical locations, i.e., cases with partial memberships. The way to define and work with partial membership cases is similar to that of partial membership instances (See 1.5.6.4.).

1.7. Preparing a Raster *Casebase*

1.7.1. Components of a Raster *Casebase*

Different from a *rulebase* (a single DBF file) or a vector *casebase* (a single Shapefile), a raster *casebase* is composed of two parts:

- a DBF file containing *rules* and *spatial setting*;
- a raster layer containing cells.

A raster *casebase* only contains one set of *rules* and one *spatial setting*. This one set of *rules* and one *spatial setting* will be applied to all the *cell cases* in the raster layer.

1.7.2. Creating a Raster *Casebase*

ArcSIE uses an ArcInfo Grid to create a new raster *casebase*.

To create a new *casebase*, click 🗋. In the File Browsing dialog box specify the Grid you want to use.

The spatial extent and cellsize of the Grid must match the settings of the *environmental database*.

One Grid can only contain *cases* of one (soil/landform) type. The *case* cells should have value **1** and the *non-case* cells should have value **0**.

At any time, ArcSIE can host only one *raster casebase*. Therefore, if there is a *raster casebase* currently loaded, it will be replaced by the new *raster casebase*. You will be prompted to save the changes you made to the old *casebase* before it is unloaded.

With a *raster casebase* you cannot add a second set of *rules* or remove the current set of *rules*.

1.7.3. Loading an Existing *Raster Casebase*

Click 📂 to load an existing *raster casebase*. You only need to specify the DBF file. ArcSIE will automatically load the corresponding raster layer.

If there is a *raster casebase* currently being used, the current one will be replaced by the one you are opening. You will be prompted to save the changes you made to the current *raster casebase* before it is unloaded.

If the DBF file does not contain the required information for a raster *casebase*, an error message will appear and the loading process will abort.

1.7.4. Editing the *Rule* Part of *Cell Cases*

Editing the *rule* part of the *cell cases* is exactly the same as editing an *instance*. See 1.5.6.

Note: The *central* values (v_1 and v_2) and coordinates (*x* and *y*) displayed in the dialog box are the values and coordinates for one of the *cell cases* in the raster *casebase*. These values only provide general information about the *casebase*. The actual values of each *cell case* will be used in the inference.

1.7.5. Editing the *Spatial Setting* of *Cell Cases*

Editing the *spatial setting* of the *cell cases* is the same as editing the *spatial setting* of a *vector case*. See 1.6.6.

Note: The *Influence Region* is defined for each *cell case* in the raster *casebase*.

1.8. Running Inference

The soil maps created by ArcSIE are fuzzy membership maps, with each one indicating how similar the soils at different locations are to the most typical soil of a specified soil type. The maps use the raster data model and are in ArcInfo Grid format.

1.8.1. Creating a Map for the Currently Selected *Soil Type*

Click and give a name to the output soil map. ArcSIE will run inference for the currently selected *soil type* or the *soil type* that contains the currently selected *instance* or *case*.

If the current selection is the *knowledgebase* (*rulebase* or *casebase*) name, ArcSIE will run inference for the first *soil type* in the *knowledgebase*.

1.8.2. Creating Maps for all the *Soil Types*

Click and specify a folder to host the created maps. ArcSIE will then run inference and create a set of fuzzy membership maps, one for each *soil type* in the current *knowledgebase*.

The maps will be named after their corresponding *soil types*. If a Grid with the same name already exists in the specified folder, the new Grid's name will be formed as the *soil type* attached by a number (1, 2, 3, …) to distinguish it from the existing Grid. It is recommended that every time before you click Do Batch, you first create a new folder for the maps to be created.

Since all the *cell cases* are of the same (soil/landform) type, the "Do Batch" function is not applicable to a raster *casebase*.

1.8.3. More Options on the Output

Clicking More Options opens a dialog box in which you can specify:

- if you want to create a *check file*;
- if you want to use *mask file* in the inference;
- how you want to determine the *central values* for a line or polygon *case*; and
- if you want to resample the cell cases before running the inference.

1.8.3.1. *Check File*

A *check file* is a map showing two pieces of information:

- which *instance* or *case* determines the fuzzy membership at a location; and
- which *environmental feature* determines or makes the greatest contribution to the *fuzzy membership* at the location.

A *check file* is an ArcInfo Grid and is named by adding "chk" at the end of the output soil map.

Each value in a check map has two parts: The integer part is the *instance's* ID, indicating the determining *instance* at that location. The *instance's* ID corresponds to the order of the *instance* in the *soil type*. If none of the existing *instances* has an effect on a location, the integer part of the *check file* value at that location will be -1.

The decimal part is the *environmental feature*'s ID, indicating the determining *environmental feature* at that location. To find out the name of the *feature,* you need to check the look-up table created with the *check file*. The look-up table is a text file and has the same name as that of the *check file*, but has a suffix "lkt".

For example, if a check file has value 3.04 for a location and its associated look-up table is as follows:

0	Elevation
1	Slope
2	Aspect
3	ProfileCurve
4	PlanformCurve
5	Wetness

The user will know that the fuzzy membership at this location is determined by the third *instance* or *case* and planform curvature is the *environmental feature* that determines or makes the greatest contribution to the fuzzy membership.

If Limiting Factor is used for the *P* function, the *environmental feature* reported by the *check file* for a location is the one that has the smallest optimality value at that location.

If Weighted Average or Multiplication is used for the *P* function, the *environmental feature* reported by the *check file* for a location is the one that makes the greatest contribution to the fuzzy membership. For Weighted Average, this *environmental feature* is the one that has the maximum (optimality value × weight); For Multiplication, this *environmental feature* is the one that has the maximum optimality value.

1.8.3.2. *Mask File*

A *mask file* is for masking out areas that you do not want to include in the inference. Those masked areas will be assigned Nodata in the created soil map.

If the value of a cell in the *mask file* equals the specified *Masking Value*, the corresponding cell in the output soil map will be assigned Nodata.

This masking function is an "in or out" function, i.e., it simply classifies all the locations covered by the *environmental database* into two classes: locations that should be included in the inference and locations that should be excluded from the inference. This masking function is different from the masking function associated with an individual *soil type*, *instance* or *case*.

This masking function is useful for defining the mapping area when you want to set a special boundary for the output soil map, e.g., when you only want to map the soils within a watershed. It may also be used to limit the spatial domain of the model, e.g., to a specific geologic formation or physiographic area.

1.8.3.3. *Central value* for *line/poly case*

When using a *line/polygon case* to perform inference, the line or polygon will first be rasterized. That is, in the actual inference, a line will be treated as a series of cells and a polygon will be treated as an agglomeration of cells.

For a *line/poly case*, ArcSIE allows the user to specify what value to display in the v_1 and v_2 fields (for *line/poly cases*, v_1 always equals v_2) and to use in the inference. ArcSIE provides four options to specify this value:

(1) Use the local value at each cell;

(2) Use the mean of all the cells of this *case*;

(3) Use the median of all the cells of this *case*;

(4) Use the mode of all the cells of this *case*. "Mode" refers to the most frequently appearing value. When calculating the mode, all the values within the *tolerance* range of a value will be counted as if they are equal to that value; i.e., if a value is between x - tolerance and x + tolerance it will be considered equal to x.

For options (1) and (2), the v_1 and v_2 fields will show the mean value; for (3) and (4), the fields will show the corresponding values.

In the inference, each cell in the *line/poly case* is treated as a *point case*. Under option (1), the *central values* of such a *point case* are read from the cell's location; other than that, these *point cases* are exactly the same in terms of their *E*, *P* and *T* function settings, as they inherit these settings from the *line/poly case* they belong to. Under options (2), (3), and (4), all the *point cases* are exactly the same (including the *central values*), except the spatial locations.

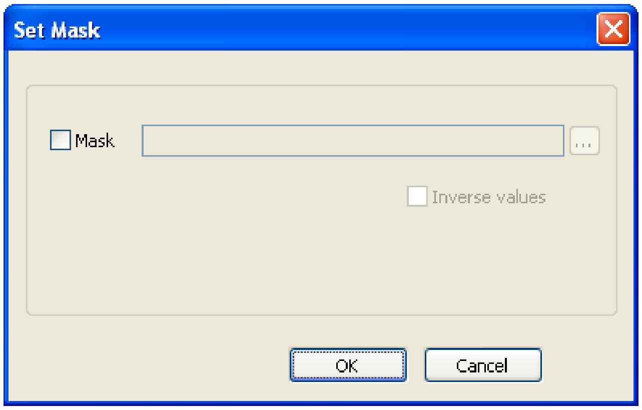
1.8.3.4. Resampling *cell cases*

When the spatial resolution of the raster *casebase* is high, the number of the *cell cases* can be big, which results in a slow inference. You can resample the *cell cases* to reduce the number of *cases* used in the inference.

The *Resample Ratio* should be an integer ≥ 1. When this ratio ≤ 1, no resampling will be performed. When it is 2, one from two nearby *cell cases* will be used in the inference; when it is 3, one from three nearby *cell cases* will be used in the inference; and so on.

1.8.4. Masking Function for a *Soil Type* or *Instance/Case*

You can associate a mask file to a *soil type*, *instance* or *case* to limit its effective area. To specify the settings for this function, right-click the name of the *soil type*, *instance* or *case* and select *Mask* from the popup menu. A dialog box will appear as follows:



Checking on the *Mask* checkbox enables the masking function for this *soil type* or *instance* and also activates the other options on this dialog box. The values in the mask file should vary between 0 and 1. In the inference, the inferred values for the *soil type* or from the *instance* or *case* will be multiplied by the corresponding values in the mask file. The multiplication operation allows the effective area to have a fuzzy boundary. If you check on *Inverse values*, the calculation (1 – value) will be performed on the mask file before it is applied to the inferred values.

Checking off the *Mask* checkbox disables the masking function for this *soil type*, *instance* or *case*, even if you have associated a mask file to the *soil type*, *instance* or *case*.

1.8.5. Two Types of Polygon-based CBR

ArcSIE implements two ways to use *polygon cases*:

- Running point CBR using each location within the polygon as a typical point *case*
- Making a fuzzy boundary of the polygon

1.8.5.1. Running point CBR using each location within the polygon as a typical point *case*

This inference assumes that the polygon defines a typical region for the given soil and every location within the polygon is a typical location for the soil. The inference engine will first rasterize the polygon and will use each cell within the polygon as a *point case* to conduct point CBR.

All the "*point cases*" from the same polygon *case* will share the setting of the *polygon case*, except v_1 and v_2, which will be the actual value at each specific point. These two values for each "*point case*" will be equal and cannot be modified. The v_1 and v_2 fields are disabled under the Polygon CBR option.

1.8.5.2. Making a fuzzy boundary of the polygon

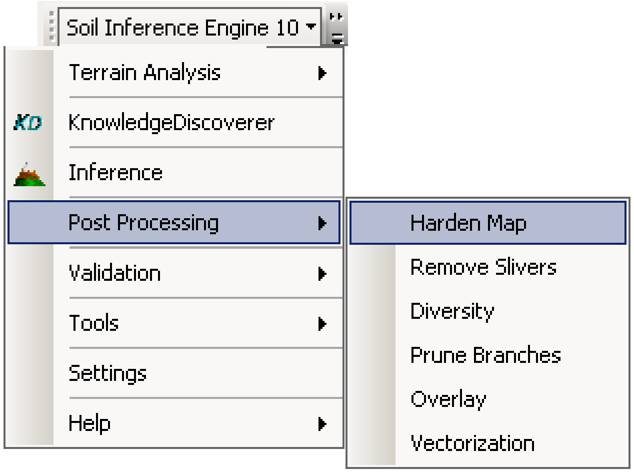
If you consider that the polygon only roughly defines the geographic region of a soil, and you assume that the closer to the central part of the polygon, the more typical is the condition for the soil, you can use the Making Fuzzy Boundary function, available under the Polygon CBR option.

ArcSIE provides ways for you to specify how fuzzy and how broad the fuzzy boundary should be. The process to make a polygon with fuzzy boundary is as follows:

a. Choose to conduct "Polygon Case" inference.
b. Load the *polygon case* Shapefile.
c. Specify *positive* or *negative* for each case, if you have not done that before.
d. Specify the "Membership Value" for each *polygon case*. Different from a normal CBR, this value will not be used as the maximum/minimum value for the *cases*, but will be used as the fuzzy membership values at the boundary of the polygons. For example, if you specify the value to be 0.4, then the program will assign 0.4 as the fuzzy membership to the cells right on the original boundary lines of the polygon and will base other inference computing on this setting. Under the fuzzy boundary operation, the maximum fuzzy membership value for a *positive* case is always 1 and the minimum value for a *negative case* is always 0.
e. Adjust environmental features. For making a fuzzy boundary, you should only use either *bell-shape* or *Default* function curves.
f. Check ☑ Make Fuzzy Bnd.
g. The Spatial Setting panel will automatically open, because this has to be a spatial operation. The "Spatial Setting" button is disabled to avoid accidentally closing the Spatial Setting panel. Adjust the parameter values on the spatial setting panel as needed.
h. Click the appropriate one of the two "Do" buttons to run the inference.

Chapter 2. Post Processing

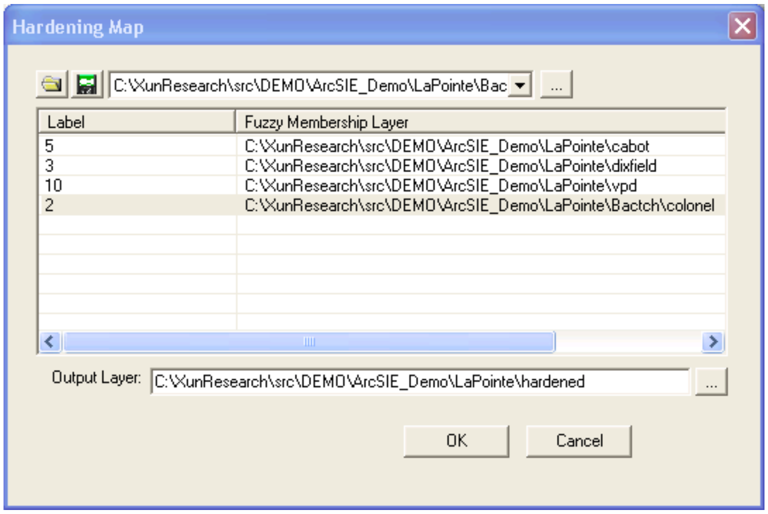
ArcSIE provides tools to process the raster layers generated by the Inference Engine or other tools. Most of these post-processing tools are for creating vector maps from the raster fuzzy membership layers that meet the current soil mapping standards. The submenus for the post-processing tools are shown in the graphic below:



2.1. Make Hardened Map

This tool integrates the fuzzy membership maps (for individual soil types) into a single raster map. The result can be used to create a traditional vector map.

The "hardening" process picks the soil type with the highest fuzzy membership value at a location as the representative soil type at that location. In a hardened map, a cell is only labeled with its representative soil type.



C:\XunResearch\src\DEMO\ArcSIE_Demo\LaPointe\Bac

Label	Fuzzy Membership Layer
5	C:\XunResearch\src\DEMO\ArcSIE_Demo\LaPointe\cabot
3	C:\XunResearch\src\DEMO\ArcSIE_Demo\LaPointe\dixfield
10	C:\XunResearch\src\DEMO\ArcSIE_Demo\LaPointe\vpd
2	C:\XunResearch\src\DEMO\ArcSIE_Demo\LaPointe\Bactch\colonel

Output Layer: C:\XunResearch\src\DEMO\ArcSIE_Demo\LaPointe\hardened

You must specify all the fuzzy membership maps to be included in the hardening process. For each map, you also must specify an integer number as the *label* of the soil type represented by the map. The *label* will be the value representing that soil type in the final hardened map. The Hardening Map window (shown left) facilitates this process.

For loading individual fuzzy membership maps one by one, use the dropdown list or the *browsing* button, and then edit the values in the Label column to give the proper label value to each layer.

Use 💾 to save the list of maps and their corresponding labels into a *fuzzy membership map list* file (fst) for future use.

Use 📂 to load the maps and their labels from a previously saved *gst* file.

This tool outputs three layers. Besides the hardened map itself, which will have the name as specified in the Output Layer box, there are two other layers created for representing the uncertainty caused by the hardening process. One of them, whose name is formed by attaching "ent" to the user-specified output name, contains *entropy* values calculated as follows:

$$\frac{-\sum_{k=1}^{n} \frac{s_k}{\sum_{k=1}^{n} s_k} \ln\left(\frac{s_k}{\sum_{k=1}^{n} s_k}\right)}{\ln(n)}$$

In the above equation, *sk* is the fuzzy membership value of soil type *k* at a given location, and *n* is the total number of soil types. The higher the entropy value at a location, the higher the uncertainty caused by the hardening process.

The other uncertainty layer, whose name is formed by attaching "exg", contains *exaggeration* values calculated as follows:

1-max(*s1, s2, ...sk*)

The higher the *exaggeration* value, the higher the uncertainty caused by the hardening process.

Note: Since an ArcInfo Grid can only have a name shorter than 12 letters, if you specify a long output name, the names of the two uncertainty layers may be truncated.

Reference:

Zhu, A.X. 1997. A similarity model for representing soil spatial information. Geoderma 77:217-242.

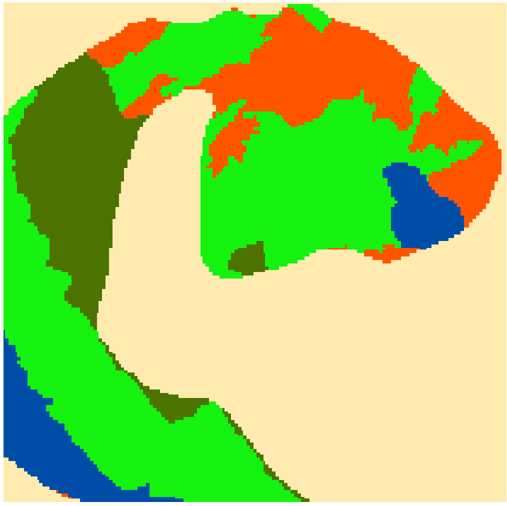
2.2. Remove Slivers

This tool merges small patches into their surrounding patches, which creates larger *patches* of cells with contiguous identical values. If vectorized, the small polygons (slivers) from these small patches may be cartographically unacceptable. The purpose of this process is to remove unwanted, excessive detail and "noise", and thereby "cleaning" the hardened map. It is typically used as a precursor to vectorization, so that the vector map can meet certain scale standard.

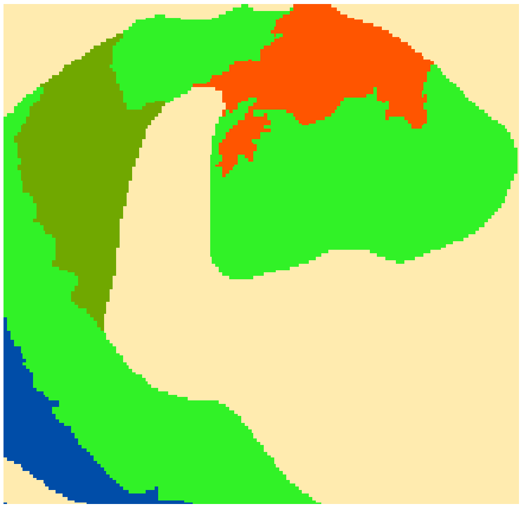
This tool removes slivers by progressively shrinking the sliver through allocating its cells to their adjacent patches.

The units of the thresholds are the unit of the input raster layer (e.g., meters).

Threshold for interior slivers: If the total size of a group of connected cells is smaller than this threshold, and none of the cells is on or touches a "borderline" (see ***Read borderline from*** for details), the patch formed by this group of cells will be considered as a *sliver*, and will be removed. If the specified value for this parameter is smaller than the size of one cell of the input raster, an error message will appear and the program will abort. The effect of sliver removal is illustrated by the graphics below:

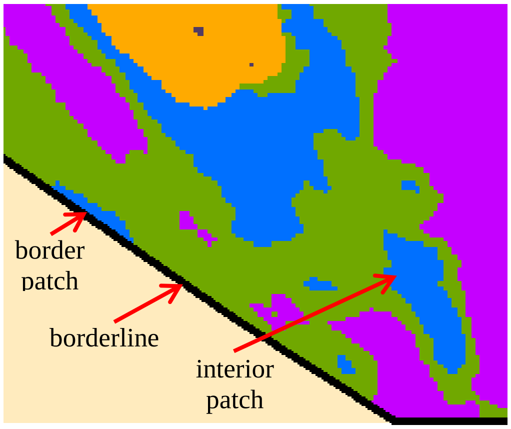


Before sliver removal

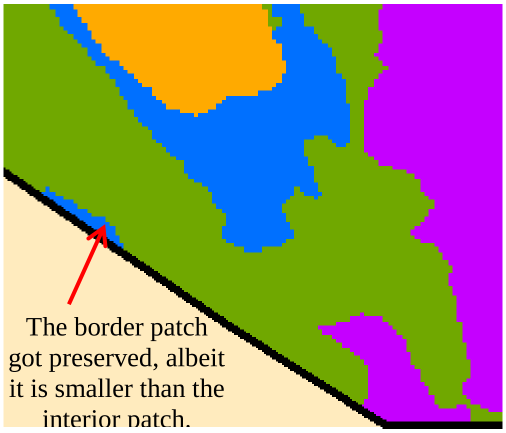


After sliver removal

Threshold for border slivers: If the total size of a group of connected cells is smaller than this threshold, and at least one of the cells is on or touches a "borderline" (see ***Read borderline from*** for details), the patch formed by this group of cells will be considered as a *sliver*, and will be removed. If the specified value for this parameter is smaller than the size of one cell of the input raster, an error message will appear and the program will abort. The reason for distinguishing the interior and border thresholds is that a *patch* at border may be part of a larger patch that crosses the boundary between two adjacent mapping areas, and therefore may not be a real sliver even if its size is small. For this reason, usually you want to specify a smaller value for the border threshold. The graphics below illustrate the effect of distinguishing the two thresholds:



Before sliver removal


The border patch got preserved, albeit it is smaller than the interior patch.

After sliver removal

Threshold to start with and ***Increment***: If the program simply uses the specified thresholds to determine and remove slivers, the results in areas where two slivers adjacent to each other may be undesirable. For this reason, it is recommended to start with a small threshold and progressively approach the final threshold. On the interface of the tool, you can specify the ***Threshold to start with*** and the ***Increment*** of thresholds between two consecutive steps in the progressive process. The specified values for these two parameters will be applied to both interior sliver removal and border slivers removal. If the ***Threshold to start with*** is greater than the ***Threshold for interior slivers*** or the ***Threshold for border slivers***, an error message will appear and the program will abort. If the specified ***Threshold to start with*** or the ***Increment*** is smaller than the size of one cell of the input raster layer, it will be automatically set to the size of one cell.

Read borderline from: By default, this option is not checked and the borderlines for determining "border slivers" are the edge lines of the input raster layer. You can check on this option to load a Shapefile that contains the user-defined borderlines. The type of the Shapefile can be either line or polygon. The Shapefile must be in the same coordinate system with the input raster. If you specify to use your own borderlines, the edge of the raster will not be used as borderlines.

Preserve isolated patches: If this box is checked, a patch surrounded by nodata will be preserved, regardless of its size.

Work on specific value(s): If this box is checked, the program will only check the cells with the specified value(s). This option allows you to only remove slivers of a certain soil type(s). If you want to specify multiple values, you can use comma (,) or semicolon (;) to separate values, and hyphen (-) or colon (:) to define range. For example:

1, 3, 7, 25
1; 3; 7; 25
1, 3; 7, 25
3-6, 10-20, 30-36
3:6, 10:20, 30:36
1, 3-6; 7, 10:20, 25; 30-36

4-connected vs. 8-connected: This option defines how contiguity will be defined when forming patches. If 4-connected is checked, only cells with identical values in the four cardinal directions will be included into a patch. If 8-connected is checked, cells with identical values in any of the eight neighboring cells will be included into a patch.

Note: The tool will not remove nodata slivers.

2.3. Calculate Diversity

This tool compares a polygon map and a raster (hardened) map, and reports percentages of different cell values within each polygon. Usually, the polygon map is the resulting map generated from the hardened raster map through the sliver-removing and vectorizing processes. This tool is thus usually used to report how much information has been dropped during the sliver-removing and vectorizing processes.

Polygon Layer: The input polygon map to compare. This is usually the polygon map generated by the vectorization process (see 2.6).

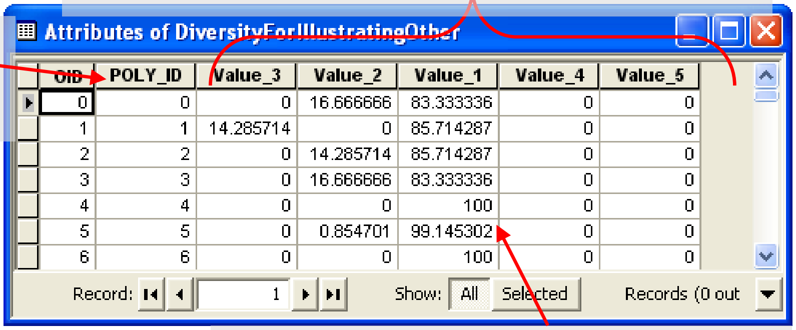
Polygon ID Field: The field in the attribute table of the Polygon Layer that contains a unique ID for each polygon. This ID is for linking the resulting diversity table back to the polygon map.

Raster Layer: The input raster layer to compare. This is usually the hardened map generated from fuzzy membership maps. You can send in the hardened map that has not gone through any sliver-removing processes or has gone through different sliver-removing processes (different parameter settings) to see how different settings drop diversity information about the polygons.

Output file: The output from the program, which is a table in DBF format. Each record in this table corresponds to a polygon in the vector map. Below is an example of the output table.

These fields correspond to the soil types in the raster map. The codes are from the raster map. The order of these fields in the table follows their total areas in the entire mapping area.

Polygon ID, unique for each polygon in the vector map.



Attributes of DiversityForIllustratingOther

OID	POLY_ID	Value_3	Value_2	Value_1	Value_4	Value_5
0	0	0	16.666666	83.333336	0	0
1	1	14.285714	0	85.714287	0	0
2	2	0	14.285714	85.714287	0	0
3	3	0	16.666666	83.333336	0	0
4	4	0	0	100	0	0
5	5	0	0.854701	99.145302	0	0
6	6	0	0	100	0	0

These values are the area percentages of those soils within each polygon.

2.4. Prune Branches

This tool cleans short streamlines from a stream network.

Network Layer: The input stream network layer. The streamlines must form a connected network, so it is recommended to use a stream layer created using O'Callaghan and Mark's algorithm (Sec. 3.4.3).

Flow Direction Layer: The flow direction layer corresponding to the input stream network layer (Sec. 3.3.1).

Minimum length: Any stream segment shorter than this value (in number of cells) will be removed.

Background value: The value of the background, i.e., cells with values different from this value will be considered as part of a stream.

2.5. Overlay

This tool overlays two raster layers. It takes only two input layers, but contains some algorithms that are not available from ArcGIS.

Max: Picks the larger value of the two corresponding cells in the input layers and assigns it to the corresponding cell in the Output Layer.

Min: Picks the smaller value of the two corresponding cells in the input layers and assigns it to the corresponding cell in the Output Layer.

Supplement: Picks the value of the two corresponding cells in the input layers that is not NoData and assigns it to the corresponding cell in the Output Layer. If neither value is NoData, outputs the value from the First Input Layer. If both values are NoData, outputs NoData.

Select: If the value of a cell in the Second Input Layer is equal to the specified "Value", outputs the value of the corresponding cell in the First Input Layer; otherwise, outputs NoData. This is like using the second layer to "select" values from the first layer.

Mask: If the value of a cell in the Second Input Layer is equal to the specified "Value", outputs NoData; otherwise outputs the value of the corresponding cell in the First Input Layer. This is like using the second layer to "mask" the first layer. It is the opposite of ***Select***.

+: Performs addition on the values from two corresponding cells in the two input layers, and assigns the result to the corresponding cell in the Output Layer.

-: Performs subtraction (First – Second) on the values from two corresponding cells in the two input layers, and assigns the result to the corresponding cell in the Output Layer.

*****: Performs multiplication on the values from two corresponding cells in the two input layers, and assigns the result to the corresponding cell in the Output Layer.

/: Performs division (First/Second) on the values from two corresponding cells in the two input layers, and assigns the result to the corresponding cell in the Output Layer.

2.6. Vectorization

This tool converts a raster map into a polygon map. A useful feature of this tool is that it will not break the polygon if two cells with the same value touch only at a corner point (i.e., are connected in a diagonal direction), thus avoiding a polygon map with numerous one-cell-sized polygons. It also integrates the removal of small isolated patches and the smoothing of the boundary lines into one process.

Input raster: The raster map to be vectorized. This is usually the hardened map (see Section 4.2) with slivers removed (see Section 4.3). The data format is Grid.
Output vector: The resulting polygon map. The attribute table of this Shapefile contains a field called "ID", and a value in this field is the majority of the values of the cells that are enclosed by the corresponding polygon.

Smoothing tolerance: Controls the degree to which polygon boundaries will be smoothed. If the tolerance is 0, the boundaries will have sharp angles reflecting the corners of cells. A good starting value to test appropriate smoothness is three times the raster resolution. For example, if the resolution is 10 m, you can start with 30 m.

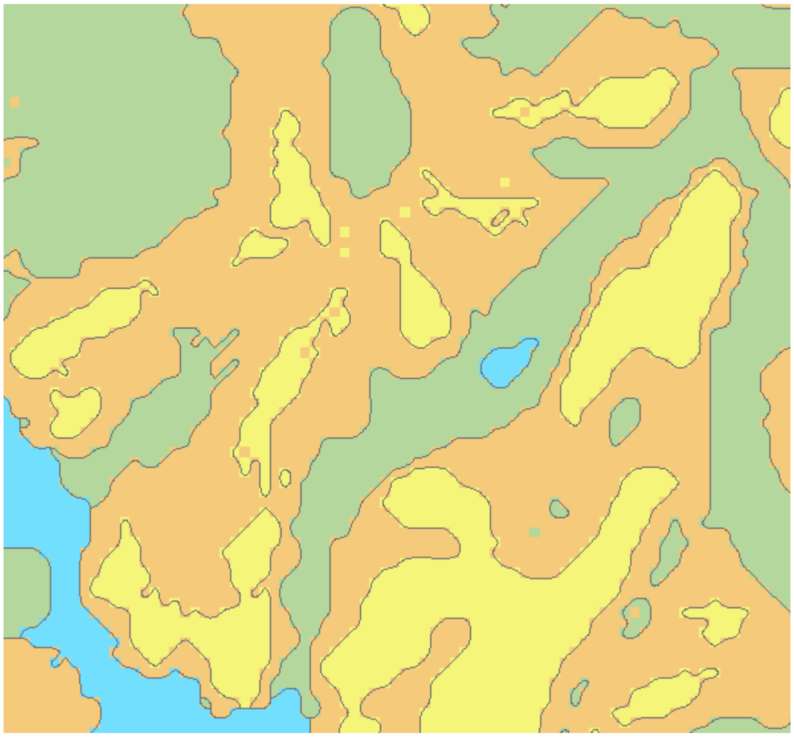
Threshold for interior slivers: Interior polygons smaller than this threshold will be removed. An "interior polygon" does not share an arc with the study area border.

Threshold for border slivers: Border polygons smaller than this threshold will be removed. A "border polygon" shares one or more arcs with the study area border.

Note: The difference between the sliver-removing function in the vectorization tool and that for raster data (see Section 2.2) is that the vector tool only removes "islands". An "island" is a small polygon that is within another larger polygon. If a small polygon shares boundaries with more than one polygon, the vector tool will not remove it. This tool is mainly used to clean the small polygons generated during the vectorization process.

Work on specific value: If this option is check on, the vectorization will only be performed on those cells with the specified value.

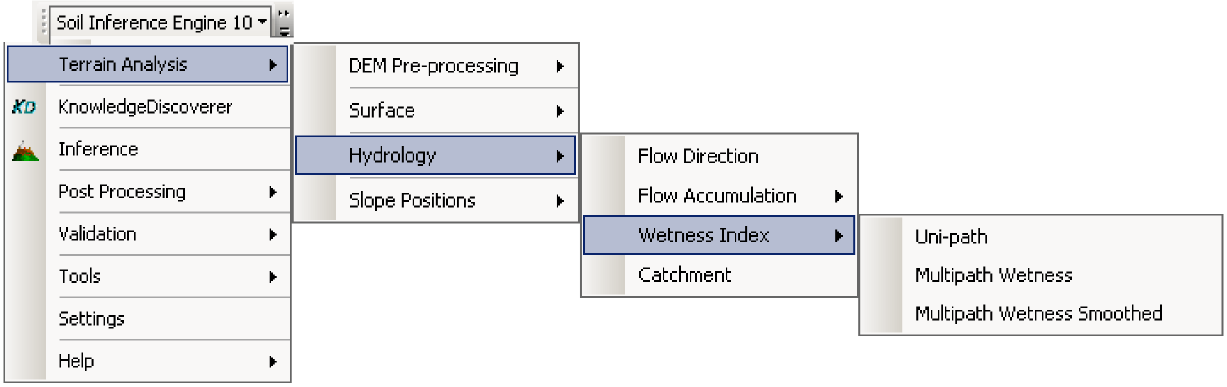
The graphic below shows a small portion of a polygon map generated by this tool. As a reference, its corresponding raster layer is put underneath the polygons.



Note: The tool will not create polygon for a nodata area.

Chapter 3. Terrain Analysis

ArcSIE provides tools for deriving terrain information from a raster digital elevation model (DEM). Some tools are unique to ArcSIE and some have been improved from their counterparts in other GIS packages. These tools were developed particularly to meet soil scientists' requests. The tools are organized into four groups, namely *DEM Pre-processing*, *Surface*, *Hydrology*, *Slope Positions*. The submenus of the tools are illustrated as below:



3.1. DEM Pre-processing

The tools in this group are for removing artifacts from DEM. They prepare the DEM for being used in terrain analysis.

3.1.1. Filling Pits

This tool removes unwanted pits from a DEM, which is a necessary pre-process in many terrain analysis operations.

Filling pits is mainly for avoiding unwanted stops when modeling flow paths using a DEM. This should be done before calculating flow direction, flow accumulation, catchment, and wetness index.

The user must specify the maximum depth of pits to be filled. All basin areas whose depths are smaller than the specified depth will be filled. A basin area is defined as an area wherein water accumulates, rather than flowing out through an outlet. The depth of a basin area is defined as the difference between the lowest point in the interior (i.e., the bottom) and the lowest point on the edge of the area (i.e., the outlet). After filling, all the interior cells will be assigned the elevation value of the outlet, thereby creating a flat region in the original basin area.

If you wish to fill all internal catchments so that all water flows to outlets at the edges of the DEM, you must specify a large value for the maximum depth.

3.1.2. Removing Spikes

This tool is designed to remove small bumps, such as outcrops, in a high-resolution DEM, and also reduce the number of unreasonably isolated ridge summit cells when creating a ridgeline layer from a DEM.

The process contains two general steps: identifying spikes and removing them.

The tool provides two methods for identifying spikes: automatic and user-specified. Once spikes are identified, the program replaces the values in a spike area with the values derived from its surrounding area so as to flatten the spike area.

Read spike locations from a point file: You can create a point Shapefile to specify the locations you see as spikes. For each point, you can also specify the size of the spike (stored in a "size" field in the attribute table), so that the program knows how big the area to process. The size is stored in a field in the attribute table of the Shapefile. It is represented by diameter (the program assumes that the shape of a spike is close to a circle) and is in the unit of the input DEM (e.g., meter). If the user does not provide the point Shapefile, the program will identify spikes using an automatic algorithm and apply a constant size (see *Default Size of Spike*) to every spike it finds.

Field for Size of Spike: The name of the field that contains the size of spike. If you do have size values for individual spikes, you want to specify which field in the attribute table contains the values. The size is represented by diameter (the program assumes that the shape of a spike is close to a circle) and is in the unit of the input DEM (e.g., meter). If you provide the point file but do not specify the "size" field, the constant user-specified default size will be applied to every user-specified spike. If you do not provide a point Shapefile, this option will be disabled.

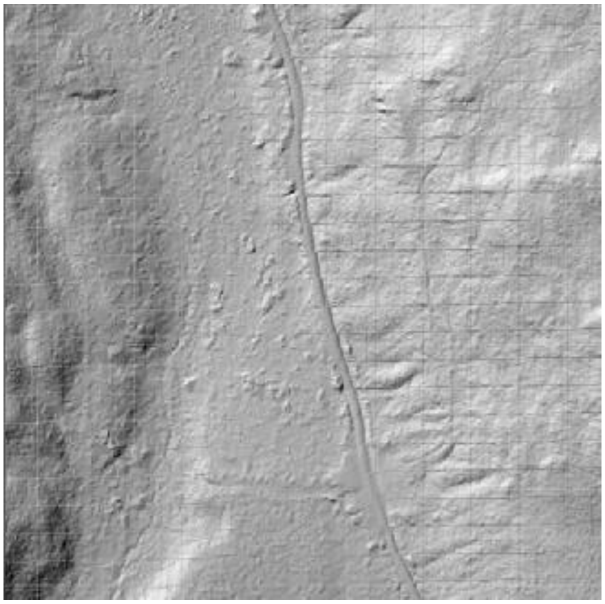
Default Size of Spike (Diameter): This is the constant value for the spike size under the automatic mode or the user provides a point shapefile but no size field. You should specify this value according to the situation in your specific mapping area. The size is represented by diameter (the program assumes that the shape of a spike is close to a circle) and is in the unit of the input DEM (e.g., meter).

Distance Decay Factor: When using the surrounding values to derive new values for the spike area, the program will apply the distance decay rule, i.e., the cell that is close to the spike area will be more important in the derivation. The greater the distance decay factor, the more important the distance, i.e., a close cell will become even more important. Meanwhile, the greater this factor value, the less smooth the resulting new surface.

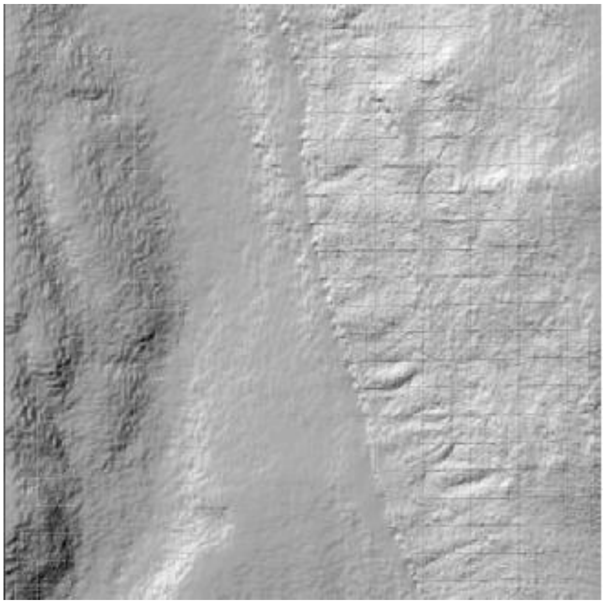
Only remove spikes lower than (in the unit of DEM): This option allows you to specify that you want to remove spikes in areas that are below a certain elevation.

Number of Iterations: This specifies the number of times the process repeats. Usually, repeating the process a number of times improves the result. The default number set for the program is 10. This option only applies to the automatic process.

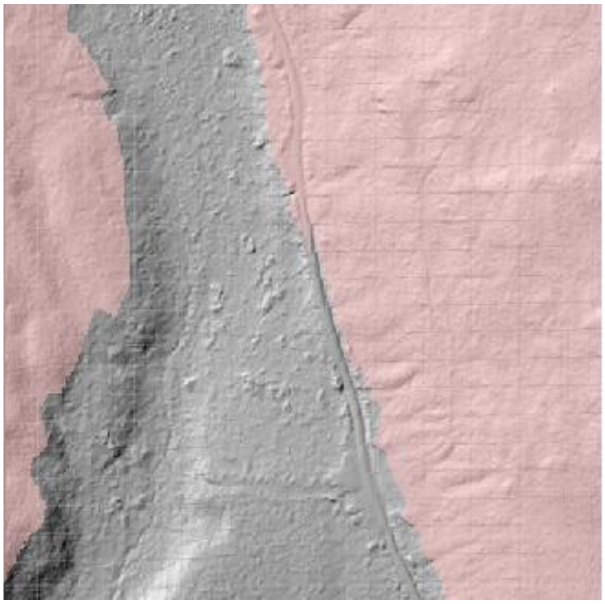
The graphics below illustrate the effect of this tool:



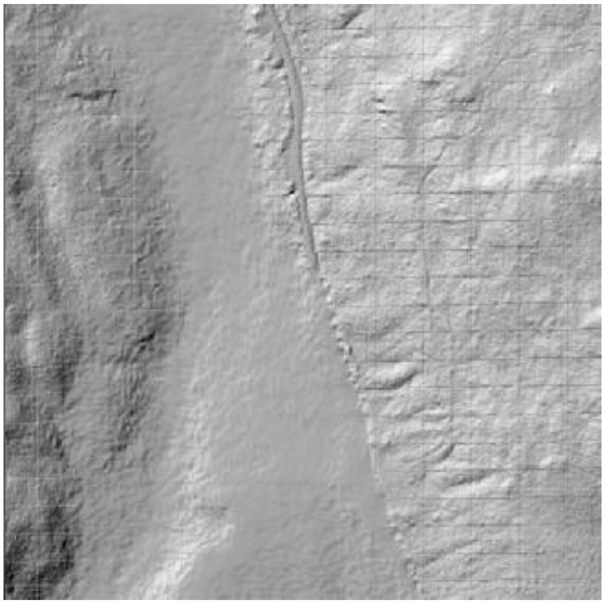
Orignial DEM (hillshade image)



Spikes removed (by default setting)



Original DEM (reddish areas have elevation >= 635 m)



Spikes removed (specified only for areas < 635 m)

3.1.3. Removing Linear Artifacts

This tool is designed to remove linear bumps, such as roads, from a high resolution DEM.

The program cannot automatically identify linear bumps. You need to provide a polyline Shapefile to tell the program where such linear features are.

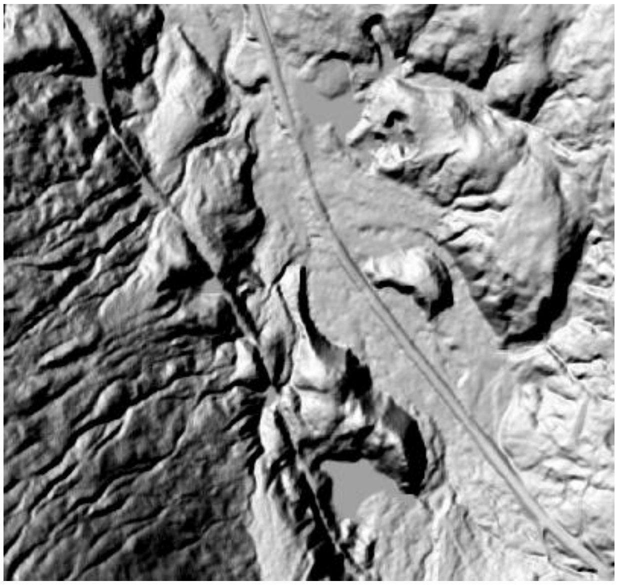
The removing process contains two steps. First, the program derives new values for the cells that are on the bumps; the new values are derived from the cells that near the linear bumps. Second, the derived values are smoothed by averaging the values in its vicinity.

Polyline Feature Class**:** A polyline Shapefile that contains the centerlines of the linear bumps.

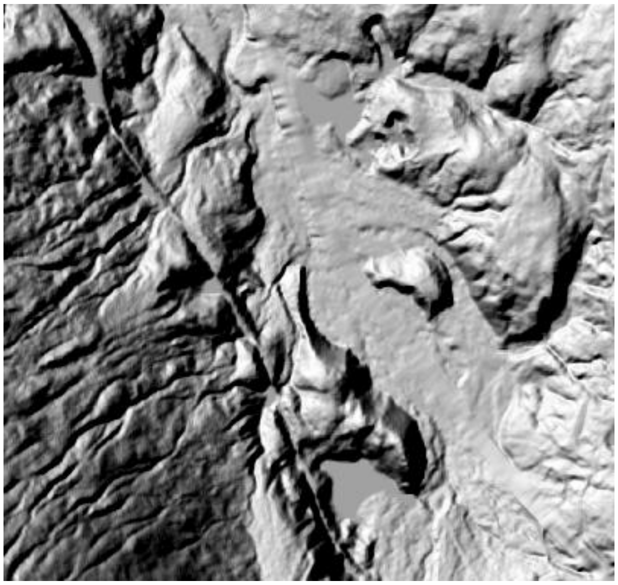
Buffer Distance**:** This distance is in the unit of the input DEM (e.g., meter). The program assumes that the linear bump can be represented by a "belt" that can be created by applying buff operation to the centerline. This parameter specifies the buffer distance for this operation. The buffer distance is determined by the width of the linear bump. **Note**: The cells on the buffer boundary will be used to infer the new values for the cells within the buffer (i.e., derive new values for those *bump* cells. This should be considered when you specify the buffer distance.

Degree of smoothing: This is the number of iterations of the smoothing operation, i.e., the smoothing operation will repeat this number of times. The larger the number, the smoother the result.

The graphics bellow illustrate the effect of this tool:



The original DEM (hillshade image)



The road in the DEM has been removed.

3.2. Surface

The tools in this group derive terrain attributes characterizing land surface. They are all so-called *focal* analysis that works with a *moving window*. The basic operation is to use the cells in a predefined window (e.g., a 3x3 window) to calculate value for the cell at the center of the window. This operation is repeated for each cell in the raster, i.e., the moving window *scans* the entire raster.

3.2.1. Surface Derivatives

This is a toolbox for calculating some of the most commonly used terrain attributes, including slope gradient, aspect, and various types of curvatures. Mathematically these attributes can be considered as the first- or second-order derivatives of a continuous terrain surface.

3.2.1.1. Terrain Attributes

Gradient**:** Slope gradient, measuring the steepness at a location. The output is in percentage (45 degrees = 100%).

Aspect**:** Slope aspect, giving the direction the slope facet is facing. The output is in degrees, starting from north and increasing in the clockwise direction (east = 90, south = 180, west = 270).

Profile curvature**:** Measures the shape of the slope surface in the vertical (up-and-down) direction. Positive values indicate convex shapes and negative values indicate concave shapes.

Planform curvature**:** Measures the shape of the slope surface in the horizontal (cross-slope) direction. Positive values indicate convex shapes and negative values indicate concave shapes.

Tangent curvature**:** Measures the shape of the slope surface in the direction that is perpendicular to the surface at the given location. Negative values indicate convex shapes and positive values indicate concave shapes.

Min curvature**:** Gives the minimum curvature at a location. Negative values indicate convex shapes and positive values indicate concave shapes.

Max curvature**:** Gives the maximum curvature at a location. Negative values indicate convex shapes and positive values indicate concave shapes.

Curvature**:** Measures the overall shape at a location. Positive values indicate convex shapes and negative values indicate concave shapes.

Note: For the curvature layers generated by the tools in ArcToolbox or Spatial Analyst of ArcGIS, the interpretation of the sign (positive or negative) might be different.

3.2.1.2. Algorithms

Evans-Young**:** Based on the idea that the terrain surface can be modeled by a quadratic polynomial.

Horn**:** Similar to Evans-Young method, but in the specific (finite-element) calculation gives the cells in the cardinal directions heavier weights.

Zevenbergen-Thorne**:** Based on the idea that the terrain surface can be modeled by a Lagrange polynomial.

Shi**:** A modified Zevenbergen-Thorne method. It first applies the Zevenbergen-Thorne method to the four cardinal cells and four diagonal cells separately, and then uses the average of the results from the two operations as the final result.

Note: For all the curvatures, the "Horn" option is equivalent to the "Evans-Young" option, and the "Shi" option is equivalent to the "Zevenbergen-Thorne" option. The tool in ArcToolbox uses the "Zevenbergen-Thorne" algorithm.

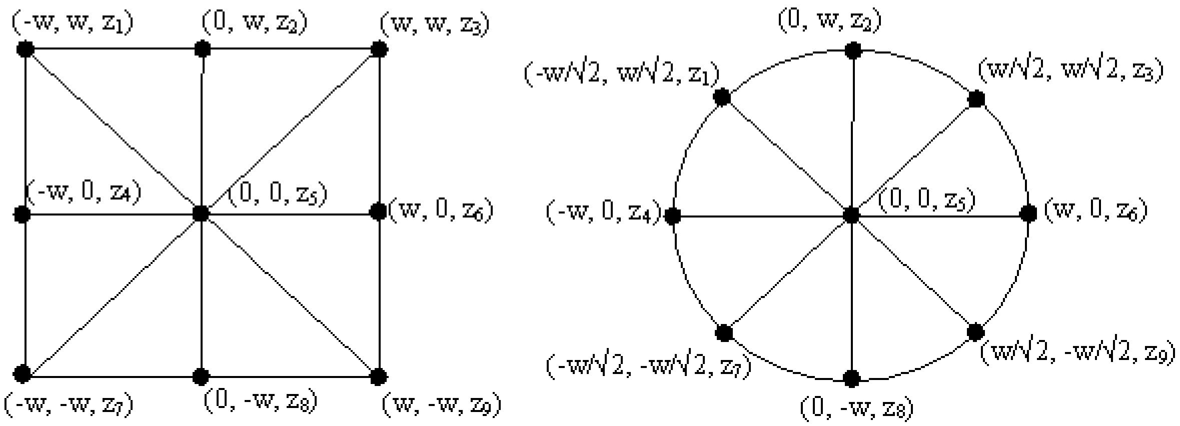
3.2.1.3. Neighborhood Size

Traditionally, the neighborhood size for these terrain attribute algorithms is determined by the cell size, since the calculation for a given cell is based on the 3×3 contiguous cells in its neighborhood. For example, with a 30m DEM, the width of a traditional 3×3 neighborhood is 20m (measured between the centers of cells). ArcSIE allows the user to define an arbitrary size for the neighborhood, independent of the cell size. The values of those elevation values on the neighborhood edge (needed for calculating terrain attributes) are estimated through bilinear interpolation.

Larger neighborhood sizes create "smoother" results and may generally reduce local "noise" in the surface, but may also dampen the signal from strong breaklines, such as escarpment edges or floodplain boundaries. Specifying a neighborhood larger than the traditional neighborhood may be desired when working with a high-resolution DEM, such as a LiDAR-based DEM. Also, larger neighborhood sizes for curvature may be more effective for identifying features such as head slopes and nose slopes.

3.2.1.4. Neighborhood Shape

The traditional neighborhood for calculating terrain attributes from a raster DEM is determined by the 3×3 contiguous cells and therefore is a square, which leads to directional bias. ArcSIE implements a circular neighborhood to mitigate this problem. The elevation values on the edge of the circular neighborhood (for calculating terrain attributes) are estimated through bilinear interpretation. The two kinds of neighborhoods are illustrated in the graphic in next page.



a. square neighborhood

b. circular neighborhood

References:

Shi, X., Zhu, A-X., Burt, J., Choi, W., Wang, R-X., Pei, T., and Li, B-L., 2007, An Experiment with Circular Neighborhood in the Calculation of Slope Gradient from DEM, *Photogrammetric Engineering & Remote Sensing*, **73**(2): 143-154.

Shary, P.A., 1995. Land surface in gravity points classification by complete system of curvatures, *Mathematical Geology*, 27:373-390.

3.2.2. Ruggedness

This tool implements the Topographic Ruggedness Index (TRI) developed by Riley, et al. (1999). TRI is the rooted mean squared difference between the elevation of a cell and the elevations of the cells in its neighborhood:

$$TRI_0 = \sqrt{\frac{\sum(z_0 - z_i)^2}{n}}$$

where TRI_0 is TRI value at cell *o*; z_0 is the elevation of cell *o*; z_i is the elevation of a cell in cell *o*'s neighborhood; n is the total number of cells in cell *o*'s neighborhood.

The parameters on the interface are explained as follows:

Radius of Neighborhood (No. of cells): Expanded from the conventional TRI calculation, which uses a simple 3×3 neighborhood, ArcSIE allows you to specify the size of the neighborhood.

Neighborhood Shape: Expanded from the conventional TRI calculation, which uses a simple 3×3 neighborhood, ArcSIE allows you to choose to use a circular neighborhood, which is more reasonable in defining neighborhood for a cell.

TRI measures local variance in elevation. The authors of this index suggest a classification as follows, based on the output TRI values:

TRI	***Class code***	***Description***
0 – 80 m	1	a level terrain surface
81 – 116 m	2	nearly level surface
117 – 161 m	3	a slightly rugged surface
162 – 239 m	4	an intermediately rugged surface
240 – 497 m	5	a moderately rugged
498 – 958 m	6	a highly rugged
959 – 5000 m	7	an extremely rugged surface

Reference:

Riley SJ, DeGloria SD, and Elliott R, 1999, A terrain ruggedness index that quantifies topographic heterogeneity. *Intermountain Journal of Science* 5:23–27.

3.3. Hydrology

The tools in this group create watershed-based terrain attributes.

3.3.1. Flow Direction

Uses the D8 algorithm (assumes that water only flows into the single neighboring cell that is in the steepest direction) to calculate flow direction. The code representing the direction at each cell is illustrated as follows (same as ArcGIS' code for flow direction):

32	64	128
16	0	1
8	4	2

3.2.2. Flow Accumulation

This tool sums up the areas of the cells whose water flows into the cell under consideration, and assigns that total area (i.e., upstream drainage area) to the cell under consideration.

The area unit of the output layer is the same as that of the input layer (e.g., square meters). This is different from ArcGIS' flowaccumulation function, whose output values are number of cells.

ArcSIE implements both uni-path and multipath algorithms.

***Uni-path** algorithm*: This algorithm takes the flow direction layer as the input and assumes water in a cell only flows into one of its neighboring cells. This is equivalent to the Flowaccumulation function of ArcGIS.

***Multipath** algorithm*: This algorithm assumes that water in a cell flows into all lower neighboring cells, and the distribution of water among lower cells is determined by the slope gradients between those cells and the center cell. The distribution of water is represented by the proportions of the area of the center cell allocated to its neighboring lower cells, and the upstream drainage area of each cell is calculated accordingly. This tool takes the DEM as the input.

3.3.3. Wetness Index

This index is also called compound topographic index (CTI).

Wetness index is calculated as

$$w = \ln(\text{Flow Accumulation}/\text{Slope Gradient})$$

ArcSIE implements both uni-path and multi-path algorithms. See discussion under 3.5 Flow Accumulation.

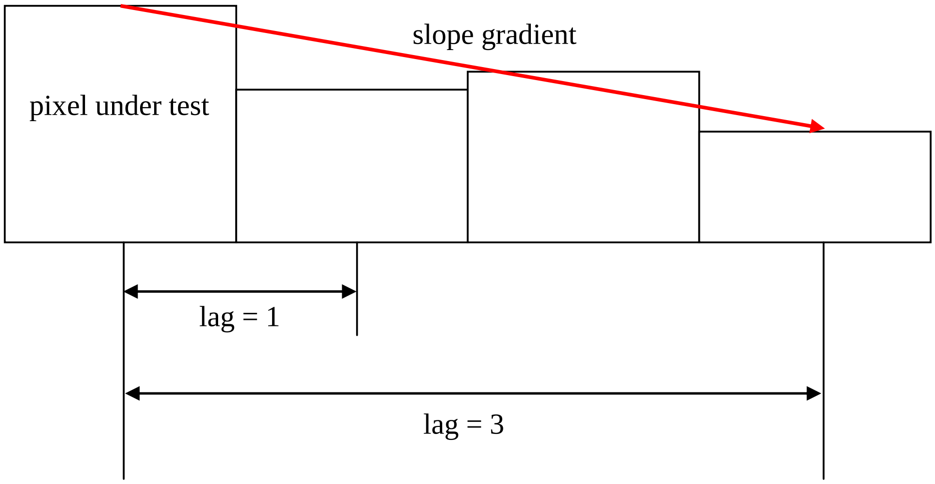
Uni-path** algorithm:** This tool takes a flow accumulation layer calculated using the uni-path algorithm and a slope gradient layer as inputs.

Multi-path** algorithm:** This tool takes three inputs:

- *Flow accumulation* layer calculated using the multi-path algorithm;

- *DEM*: Used to calculate slope gradient for each individual direction in which there is flow from the cell under consideration. This differs from the uni-path algorithm, which uses a pre-calculated slope gradient layer for the wetness calculation.

- *Lag in calculating slope*: The horizontal distance used in the slope calculation (see an illustration in next page). The default value is 1 cell, which calculates slope gradient between two adjacent cells (i.e., the conventional method). For a high-resolution DEM, you may want to specify a larger lag to reduce the impact of local details or artifacts.



***Multi-path: Smoothed** algorithm*: This method addresses the "dry valley" problem in the raw wetness index layer (for both uni-path and multi-path results): No matter how wide a valley is in reality, high values for wetness index will concentrate along a thin line. This problem is caused by the sensitivity of the conventional wetness index algorithms to slight local elevation variations. The smoothing algorithm takes five inputs:

- *Flow accumulation* layer calculated using the multi-path algorithm;
- *DEM*: Used to calculate slope gradient for each individual direction in which there is flow from the cell under consideration. This differs from the uni-path algorithm, which uses a pre-calculated slope gradient layer for the wetness calculation.
- *Lag in calculating slope*: The horizontal distance used in the slope calculation (see an illustration in previous page). The default value is 1 cell, which calculates slope gradient between two adjacent cells (i.e., the conventional method). For a high-resolution DEM, you may want to specify a larger lag to reduce the impact of local details or artifacts.
- *Vertical Smoothing Range*: If the difference in elevation between two cells is within this range, the two cells are considered to have the same elevation. Larger values will create smoother surfaces. The default value for this parameter is 0.5 map unit (e.g., meter).
- *Horizontal Smoothing Range*: The maximum distance between two cells for their elevations to be compared. Larger values will create smoother surfaces. The default value is 10 cells.

Note: For the two multi-path wetness index algorithms, the DEM is used to characterize local variation, and therefore it is not necessary to use a *filled* DEM.

3.3.4. Catchment

This tool identifies and labels all the basin areas and catchments that have outlets at the edge of the image (DEM). Each such catchment will be labeled with a unique id. The catchments are identified using the D8 flow direction algorithm. All the cells whose water flows into the same basin bottom cell or outlet cell will be labeled by the same catchment id.

3.4. Slope Positions

The tools in this group identify relative slope positions using DEM. The algorithms underlying these tools were developed by different authors for different regions and purposes. The outputs from different tools may not be comparable.

3.4.1. Ridgelines

This tool identifies and labels cells that are on ridgelines.

3.4.1.1. Algorithms

Peuker and Douglas: This algorithm identifies cells on ridgelines using two steps: first, create a binary image with the same dimension and resolution as the DEM and set the value of each cell in this image to be 1; and second, use a 3×3 moving window to scan the DEM: in this window, find the cell with the lowest elevation value and change the value of the corresponding cell in the binary image to 0. After one sweep with the window on the DEM, the cellls in the binary image still with value 1 mark the ridgelines.

O'Callaghan and Mark: This algorithm calculates the upslope accumulation area for each cell. Those cells whose upslope accumulation area values are zero represent the ridgelines.

Skidmore: In a 3×3 moving window, a cell will be labeled as a ridgeline cell if it has two opposite neighbors with *lower* elevations, and at least one of the other two neighbors (orthogonal) is lower.

3.4.1.2. Ancillary Rules for Topographic Control

Under **Topographic Controls**, you can specify that ridgeline cells must meet certain elevation and/or slope gradient criteria. If you specify a slope gradient value < 1000, you must provide a slope gradient layer. If you specify a slope gradient value >= 1000, the program assumes that you do not want to apply any control about slope gradient, and therefore does not require an input slope gradient layer.

3.4.2. Broad and Narrow Ridgelines

This tool classifies the ridgeline cells identified by the Ridgeline tool (Sec. 3.4.1) into "broad" and "narrow" ridgelines.

The program uses slope gradient values to determine if the cells around those ridgeline cells are "flat". If the slope gradient value of a cell is smaller than the specified threshold, the cell will be considered "flat". Based on this checking, the programs identifies a contiguous "flat area", if any, around the ridgeline cell under consideration.

If the area of the "flat area" is smaller than the specified threshold, the ridgeline cell under consideration will be classified as "narrow ridge".

Otherwise, the program tests the width of the "flat area" at the ridgeline cell under consideration. It tests width in 4 directions: W-E, N-S, NW-SE, and NE-SW. If the width is smaller than the specified "lower limit", the ridgeline cell is classified as "narrow ridge"; if the width is between the "lower limit" and "upper limit", the cell is classified as "broad ridge". If the width is above the "upper limit", the ridgeline cell is considered to be too "broad" even for a broad ridgeline.

3.4.3. Streamlines

This tool derives streamlines from a DEM.

3.4.3.1. Algorithms

O'Callaghan and Mark: This algorithm first calculates the upslope accumulation area for each cell, and then marks as “streamline” those cells whose UDA (upper drainage area, another name for flow accumulation) values are greater than a specified threshold. This algorithm can produce a connected, realistic drainage network, and also provides the flexibility for the user to decide the detail level of the network by changing the threshold. See Sec. 3.11.2 for a discussion of the coding methods available for this algorithm.

Skidmore: This algorithm uses a 3×3 window to scan the DEM. A cell can be defined as a stream if it has two opposite neighbors with higher elevations and the other two neighbors (orthogonal) have a lower elevation and a higher elevation, respectively. If one of the neighbors has the same elevation as that of the center cell, the same testing will be outwardly iterated on that neighboring cell, until a cell with a different elevation is found. The program uses two parameters to avoid unexpected results caused by slight elevation variation: If a cell is higher than the cell under consideration, but the difference does not exceed the **Upper Threshold**, it will not be considered higher; If a cell is lower than the cell under consideration, but the difference does not exceed the **Lower Threshold**, it will not be considered lower.

Peucker and Douglas: This algorithm identifies cells on streamlines using two steps: first, create a binary image with the same dimension and resolution as the DEM and set each cell in this image to be 1; and second, use a 3×3 moving window to scan the DEM: in this window, find the cell with the highest elevation value and change the value of the corresponding cell in the binary image to 0. After one sweep with the window on the DEM, the cells in the binary image still with value 1 mark the stream channels.

3.4.3.2. Coding Stream Segments

The stream network derived using the *O'Callaghan and Mark* method can be coded. Some coding systems are for representing the position of each stream segment in the hierarchy of the stream network; and some are for identifying the stream segment itself.

- *Simple Mark*: No coding: All the stream cells will be labeled as 1, and all the other cells are 0.

- *Strahler's order*: Label each segment according to its order in the network. The main branch has the largest order number.

- *Shreve's order*: Label each segment with a code that is the sum of the codes of its in-flow branches. The main branch has the largest code number.

- *Unique code*: Label each segment with a unique code.

3.4.4. Iwahashi-Pike Slope Positions

This tool implements the method developed by Junko Iwahashi and Richard J. Pike (2007) for identifying slope positions from DEM. The method takes three terrain attributes as input, namely slope gradient, local convexity, and surface texture. It uses a set of rules to dissect the mapping area into a number of classes of slope positions (or landscape elements). Depending on the complexity of the landscape, the user can specify to generate 8, 12, or 16 classes. However, since all the parameter values, including the neighborhood size, value thresholds, and process order, are predefined and have been hardcoded in the program, this process is considered to be *unsupervised*, and it is the user's job to interpret the resulting landscape elements and assign them real world meanings.

Below is a detailed explanation of the algorithm. With ArcSIE, however, you only need to input the DEM and slope gradient layers. All the intermediate factors used by the algorithm are automatically created during the process.

Local Convexity

The local convexity used in this method is generated in three steps:

1) Use a 3x3 *Laplacian* kernel defined as follows to filter the DEM:

0	1	0
1	-4	1
0	1	0

4-direction 3x3 *Laplacian* kernel

1	1	1
1	-8	1
1	1	1

8-direction 3x3 *Laplacian* kernel

Essentially, this filtering calculates the difference between the elevation at the center cell and the mean elevation of its neighborhood.

2) After the filtering, if a cell has positive value, i.e., if the elevation at a cell is greater than the mean elevation of its neighborhood, label this cell as being *convex* using value 1. All the other cells are assigned value 0.

3) Smooth the *convexity layer* generated in step 2) using a circular moving window with radius = 10 cells. Each cell in the smoothed layer receives a *convexity* value ranging 0 – 1, which will be used in the following classification.

Surface Texture

The surface texture used in this method is measured as the total number of pits and peaks in the neighborhood of a cell. This measurement is generated in three steps:

1) Calculate the difference between the elevation at a cell and the median elevation of its neighborhood.

2) If the difference at a cell is not 0, label it with 1, indicating that this cell is either a pit (elevation is smaller than the neighborhood median) or peak (elevation is greater than the neighborhood median). Label the others with 0.

3) Smooth the *pit-peak layer* generated in step 2) using a circular moving window with radius = 10 cells. Each cell in the smoothed layer receives a *texture* value ranging 0 – 1, which will be used in the following classification.

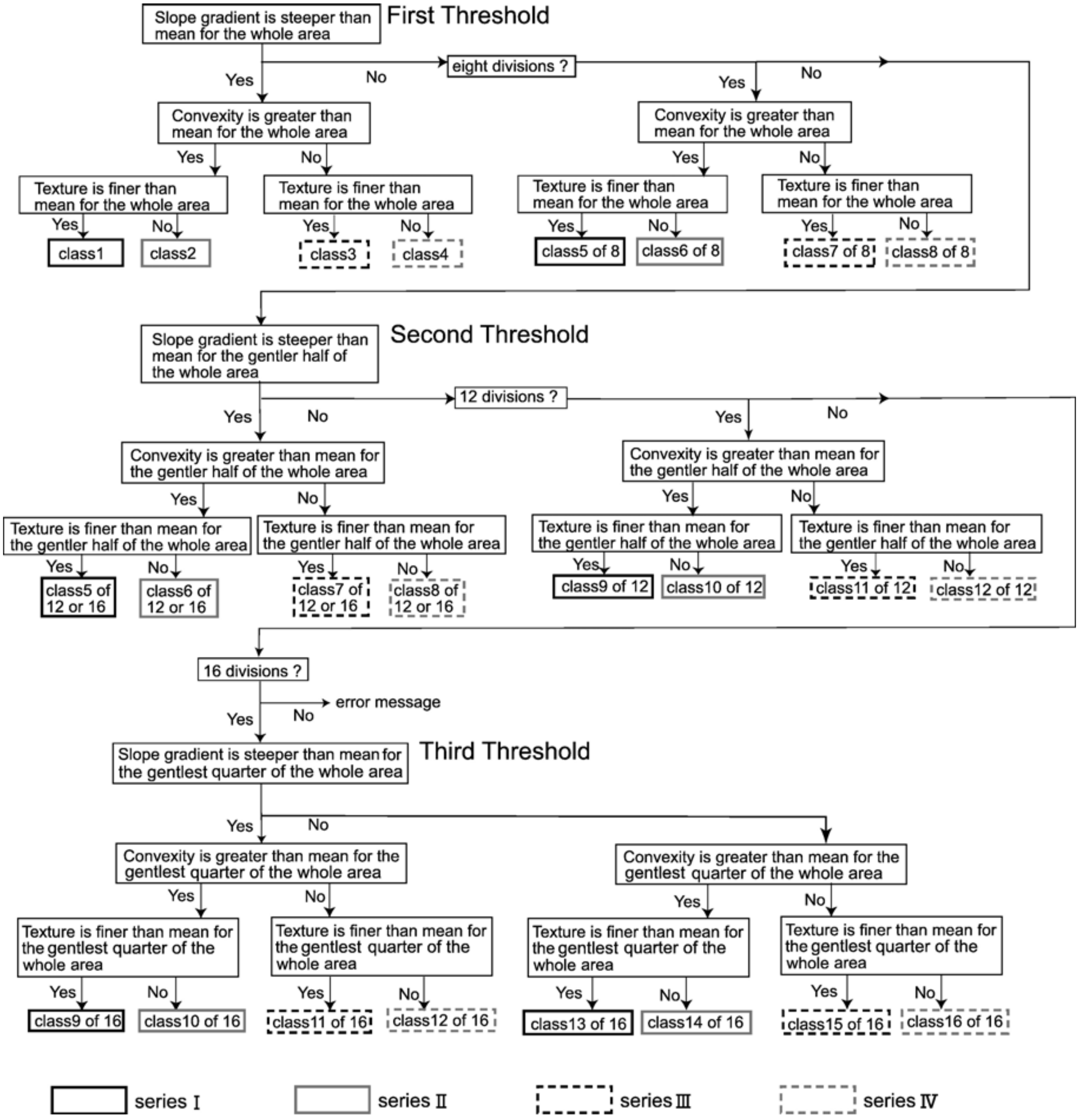
Classification

Using the three input factors, namely slope gradient, local convexity, and surface texture, the program performs a rule-based classification for the landscape. This classification is following two general ideas:

- The thresholds for distinguishing classes are mean values of the three input factors, either the mean of the entire layer, or the mean of a subset, e.g., mean of the values that are greater than the layer mean.

- The order of the three factors in the procedure is slope gradient → convexity → texture.

The rule system for classification is illustrated by the diagram as follows (Junko Iwahashi and Richard J. Pike, 2007):


First Threshold
Slope gradient is steeper than mean for the whole area
Yes
No
eight divisions ?
Yes
No
Convexity is greater than mean for the whole area
Yes
No
Texture is finer than mean for the whole area
Yes
No
class1
class2
class3
class4
class5 of 8
class6 of 8
class7 of 8
class8 of 8
Second Threshold
Slope gradient is steeper than mean for the gentler half of the whole area
12 divisions ?
Convexity is greater than mean for the gentler half of the whole area
Texture is finer than mean for the gentler half of the whole area
class5 of 12 or 16
class6 of 12 or 16
class7 of 12 or 16
class8 of 12 or 16
class9 of 12
class10 of 12
class11 of 12
class12 of 12
16 divisions ?
error message
Third Threshold
Slope gradient is steeper than mean for the gentlest quarter of the whole area
Convexity is greater than mean for the gentlest quarter of the whole area
Texture is finer than mean for the gentlest quarter of the whole area
class9 of 16
class10 of 16
class11 of 16
class12 of 16
class13 of 16
class14 of 16
class15 of 16
class16 of 16
series Ⅰ
series Ⅱ
series Ⅲ
series Ⅳ

The parameters on the user interface are explained as follows:

Input DEM Layer: The input DEM, from which the local convexity and local texture will be generated.

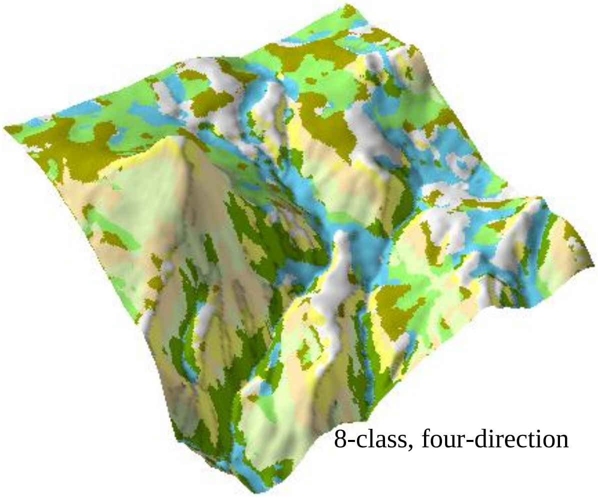
Input Slope Gradient Layer: The slope gradient layer needs to be generated beforehand, and specified here as an input. This provides the flexibility that the user can decide what parameter values to use in calculating slope gradient.

Output Integrated Slope Position Layer: The name for the output layer that contains the resulting slope positions.

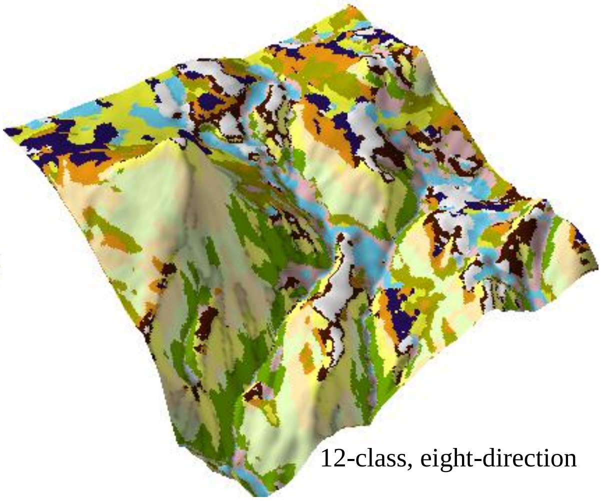
Number of Slope Position Classes: The user has to select one from the three options: 8, 12, and 16.

Laplacian Kernel Type: The user has to select one from the two options: four-direction and eight-direction.

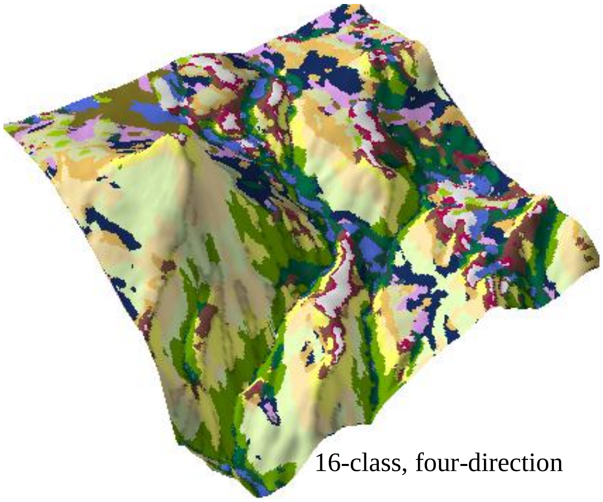
The graphics below illustrate outputs from this tool:



8-class, four-direction



12-class, eight-direction



16-class, four-direction

Reference:

Junko Iwahashi and Richard J. Pike, 2007, Automated classifications of topography from DEMs by an unsupervised nested-means algorithm and a three-part geometric signature. *Geomorphology* 86:409–440.

3.4.5. Zimmerman's Relative Positions

This tool implements a method developed by Niklaus E Zimmermann in late 1990s for deriving relative positions from DEM.

This tool generates continuous measurements of relative slope positions, from ridge top or peak to valley bottom. The continuous values can then be hardened into crisp classification of topographic positions, such as ridge, slope, toe slope, and bottom. Another feature of this tool is that it first runs at various spatial scales, and then hierarchically integrates the results at different scales into a single layer. It is argued that this way can capture terrain features at different scales, and represent them by an integrated layer.

The basic process of this tool is to calculate the difference between the elevation at a cell and the mean elevation of the cell's neighborhood. The assumption is that a large positive difference indicates that the cell is at the ridge top or peak, and a large negative difference indicates that the cell is at the valley bottom. The raw difference is then normalized as follows:

100*(Diff – demMean)/demStd

where Diff is the raw difference; demMean is the mean of the entire DEM, and demStd is the standard deviation of the DEM. Note that this is essentially translating the original difference value to Z Score and then inflating the Z Score by 100.

The neighborhood of a cell is defined by a circular search window. The calculation will be performed on a number of different sizes of the search window, representing different spatial scales. The user can specify the minimum and maximum radii of the search window, as well as the increment.

The integrated layer is generated by starting with the result from the largest window, then adding the values from smaller windows where the (absolute) values of the smaller window exceed the values of the larger window. This operation attempts to preserve important features identified by smaller search windows.

Finally, from the integrated layer the tool will generate a four-class layer for ridge, slope, toe slope, and bottom, using breaks as follows:

Value in the integrated layer	*Value in the four-class layer*	*Meaning*
< -200	4	Valley bottom
-200 -100	3	Toe slope
-100 150	2	Slope
>150	1	Ridge

Using the reclassification tool of ArcGIS, you can define your own breaks and create your own classes.

The parameters on the interface are explained as follows:

Input DEM layer: The DEM layer from which the relative slope positions are to be identified.

Output Integrated Slope Position Layer: The name of the output layer that integrates results from different search windows. This name will be used as the *base name,* based on which other output layers will be named. The layer from a specific search window will be named by attaching the radius to the base name, e.g., if the base name is *zim*, the layer from radius = 1 will be named as *zim_1*. The layer of the four-class layer will be named by attaching "_c" to the based name, e.g., *zim_c*.

Minimum Neighborhood Radius (No. of Cells): The user-specified minimum radius for the search window, measured by the number of cells. For example, if this value is 1, then the radius is 1 cell, and the search window is eventually the 3x3 window.

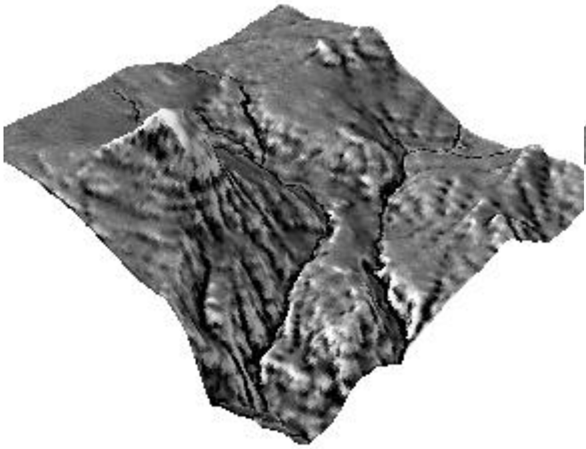
Maximum Neighborhood Radius (No. of Cells): The user-specified maximum radius for the search window.

Radius Increment (No. of Cells): The increment between two consecutive radii for the search windows. For example, if the minimum radius is 1, maximum is 5, and the increment is 1, then there will be five search windows with radius = 1, 2, 3, 4, and 5, each leading to a continuous relative position measurement layer.

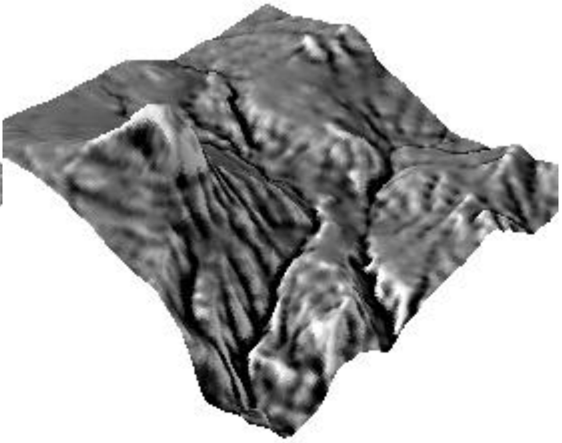
Output result of each neighborhood size: If this box is checked, all the layers from individual search windows with different radii will be output and loaded to ArcMap.

Smooth the integrated layer: If this box is checked, the original values in the integrated layer will be smoothed by a 3x3 moving window.

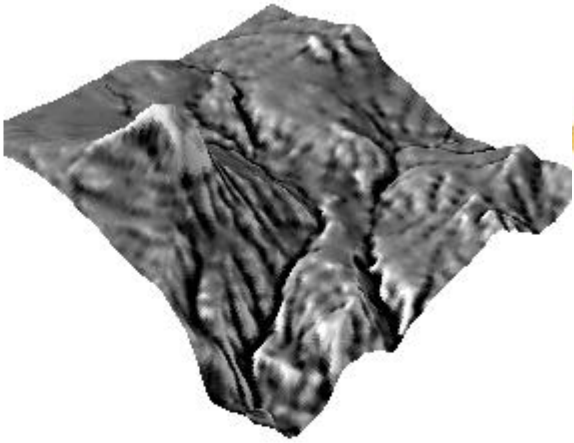
The graphics below illustrate some outputs from this tool.



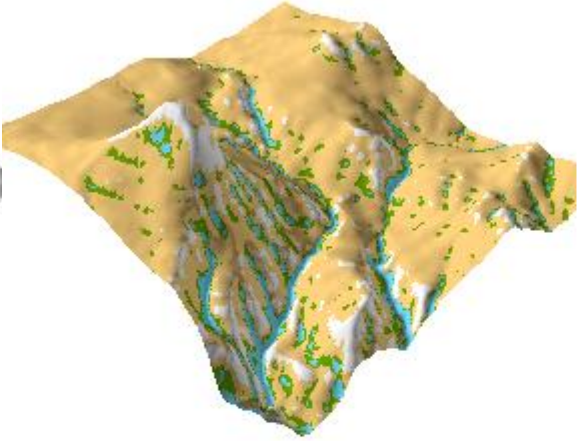
Radius = 1



Radius = 5



Integrated from Radius = 1, 2, … 5

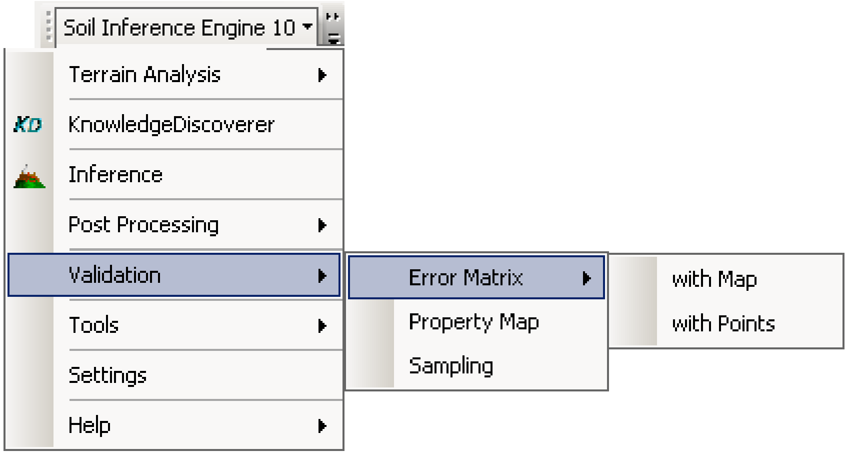


four classes

Reference:
http://www.wsl.ch/staff/niklaus.zimmermann/programs/aml4_1.html

Chapter 4. Validation

ArcSIE provides tools to facilitate validation of the maps generated by the Inference Engine. The submenus for the validation tools are shown in the graphic below:



The uses of the outputs from these validation tools, however, may not be limited to validation. For example, basically what the "Property Map" tool does is to *translate* "soil type maps" into "soil property maps", and the resulting maps can be used wherever soil property maps are needed; the "Sampling" tool implements several spatial sampling strategies and can serve the general purpose of determining sampling locations.

4.1. Generating Error Matrix

Error Matrix is a tool for evaluating the accuracy of nominal data. It can be used to compare the soil type names inferred by ArcSIE with the type names assigned by the soil scientist based on field work.

Below is an example of error matrix generated by ArcSIE:

	1	***2***	***3***	***Percent***
1	15	1	6	68
2	4	18	3	72
3	2	6	22	73
Percent	71	72	71	71

The first column contains the soil type IDs (1, 2, and 3) read from the inference result, and the first row contains the soil type IDs (1, 2, and 3) specified by the soil scientist. The numbers in the 2nd row of the matrix indicate that at 15 sample sites, both the inference engine and the soil scientist mapped the soils as type 1; at one site the inference engine mapped the soil as type 1, but the soil scientist mapped the soil as type 2; at six sites the inference engine mapped the soils as type 1, but the soil scientist mapped them as type 3; and overall among a total of 22 (15+1+6) sample sites whose soils have been mapped by the inference engine as type 1, 68% (15/22) of them match what the soil scientist mapped. On the other hand, the last number in the 2nd column indicates that among all 21 (15+4+2) samples that have been mapped by the soil scientist as type 1, the inference agreed with 71% (15/21) of them. The overall accuracy of the inference result is 71% (the number in the last row and the last column).

4.1.1. Generating with Map

This tool generates an error matrix based on a hardened map.

The parameters on the interface are explained as follows:

Sample Points: A Point Shapefile of the sample sites.

Observed Value Field: A Field in the Attribute Table of the Point Shapefile. It contains the IDs of the soil types assigned by the soil scientist and must have an integer data type.

Hardened Map File: A Raster Layer of the hardened map.

Output Error Matrix File: Output 1: An ASCII file containing the error matrix.

Output Soil List File: Output 2: An ASCII file containing a complete list of all the soil types (represented by their IDs) appearing in either the inference results or the soil scientist's specifications, or both.

Neighborhood Size: The search neighborhood when performing the comparison. Since the inference value at the cell that is right on the sample location may accidentally miss the soil scientist's specification (this is likely to happen especially when the cell size is small), you can specify to use the majority value of the *neighborhood* around the sample location rather than the individual cell value. This parameter determines the size of a square neighborhood and measures from edge to edge in number of cells. For example, if you specify Neighborhood Size = 3, then the program will define a 3×3 (cells) neighborhood around the sample location. The default value is 1, which specifies to only use the value of the cell that is right on the sample location.

4.1.2. Generate with Points

This tool (see graphic below) generates an error matrix using two sets of corresponding (paired) values stored in a table.

The parameters on the interface are explained as follows:

Value Table File: A DBF table (e.g., the Attribute Table of a point Shapefile) containing the two sets of (paired) values that will be compared.

Observed Value Field: A Field in the Value Table File. It contains the IDs of the soil types assigned by the soil scientist and must have an integer data type.

Inferred Value Field: A Field in the Value Table File. It contains the IDs of the soil types generated by the computer and must have an integer data type.

Output Error Matrix File: Output 1: A text file containing the error matrix

Output Soil List File: Output 2: A text file containing a complete list of all the soil types (represented by their IDs) appearing in either the inference results or the soil scientist's specifications, or both.

4.2. Generating Property Map

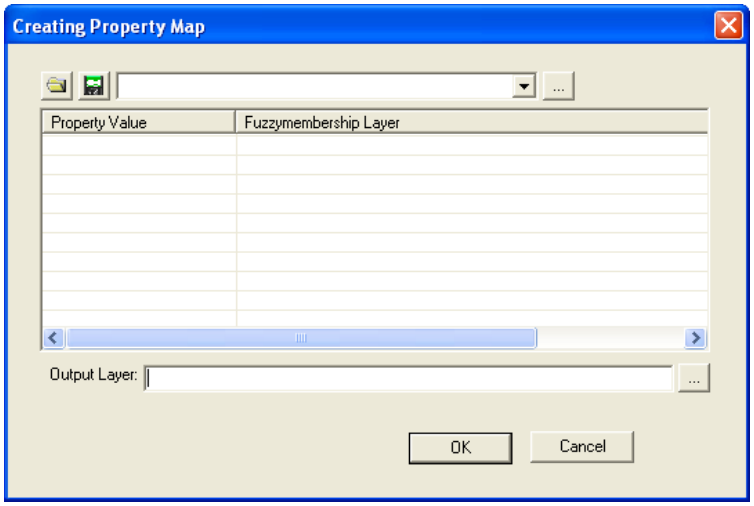
This tool generates a soil property map based on soil type maps. The method used by this tool can be represented by the equation below and the symbols are explained in the following table:

$$H_{ij} = \frac{\sum_{k=1}^{l} s_{ij,k} h_k}{\sum_{k=1}^{l} s_{ij,k}}$$

Symbol	***Meaning***
H_{ij}	the property value at location (*i*, *j*).
$s_{ij,k}$	The final fuzzy membership value at (*i*, *j*) for soil *k* (i.e., the output from function *T*).
l	The total number of soil types prescribed in the soil-landscape model used by the inference engine.
h_k	The typical property value of soil *k*.

The basic idea here is to use the fuzzy membership values of different soil types at a location as the weights to calculate the weighted average of the property values of different soil types.

Choosing the **Property Map** menu item opens the **Creating Property Map** dialog box (see graphic in next page). In this dialog box, you create a list of input fuzzy membership maps, and assign the "typical" property value for each map (Remember: each map represents a soil type). The default typical property value is 0. You can change it to the value you want to use for that soil type.



You can use the dropdown list to add fuzzy membership maps from the current ArcMap project, or use the browse button to load them from the disk.

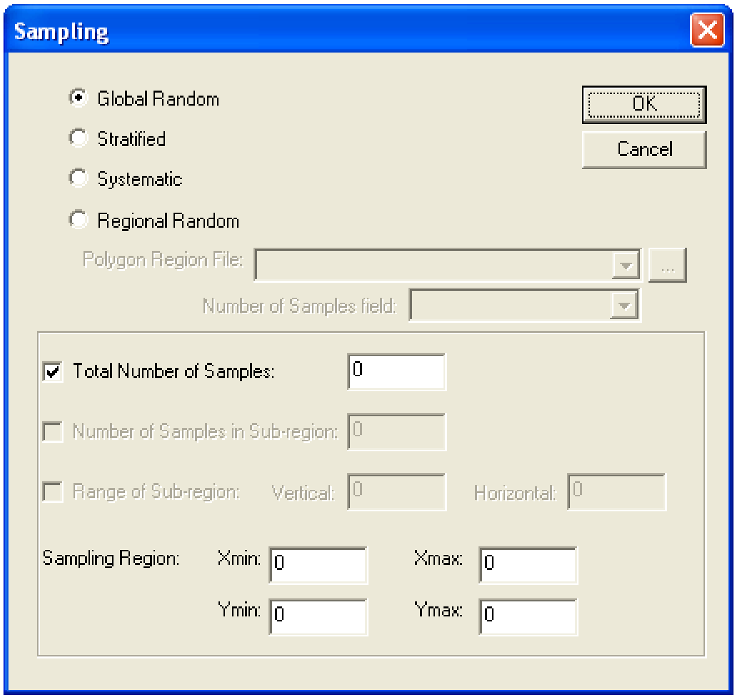
When you finish creating the property-layer list, you can save them to a lookup table file (.lkt) using the save button.

You can load a lookup table from the disk using the load button.

The output is a raster layer containing the property value at each location.

4.3. Determining Sampling Sites

This tool (see graphic below) provides four different ways to determine sampling sites.



● Global Random
○ Stratified
○ Systematic
○ Regional Random

Polygon Region File:

Number of Samples field:

☑ Total Number of Samples: 0

☐ Number of Samples in Sub-region: 0

☐ Range of Sub-region: Vertical: 0 Horizontal: 0

Sampling Region: Xmin: 0 Xmax: 0

Ymin: 0 Ymax: 0

4.3.1. Global Random

This method randomly picks locations within a rectangular region. The user must specify:

- *Total Number of Samples*

- Coordinates for defining the *Sampling Region*.

4.3.2. Stratified

This method first divides the entire rectangular sampling region into regular rectangular "Sub-regions", and then randomly picks locations within each sub-region. The user must specify:

- *Total Number of Samples*

- One of the two ways to determine the number of samples in each sub-region:

  - *Number of Samples in Sub-region*. The program will then automatically calculate the number of sub-regions and their sizes accordingly. If the remaining of

  $$\frac{\text{total number of samples}}{\text{number of sub-region samples}}$$

  is not zero, more sample sites than the specified total number will be generated to ensure every sub-region has equal number of samples.

  - *Range of Sub-region*. Enter the *Vertical* and *Horizontal* sizes in map units (e.g., meters). The program will then automatically calculate the number of samples in each sub-region. If the specified total number of samples cannot be equally allocated to the sub-regions, more sample sites than the specified total number will be generated to ensure every sub-region has an equal number of samples.

- the coordinates for defining the *Sampling Region*.

4.3.3. Systematic

This method allocates samples following a regular grid. The user must specify:

- *Total Number of Samples*

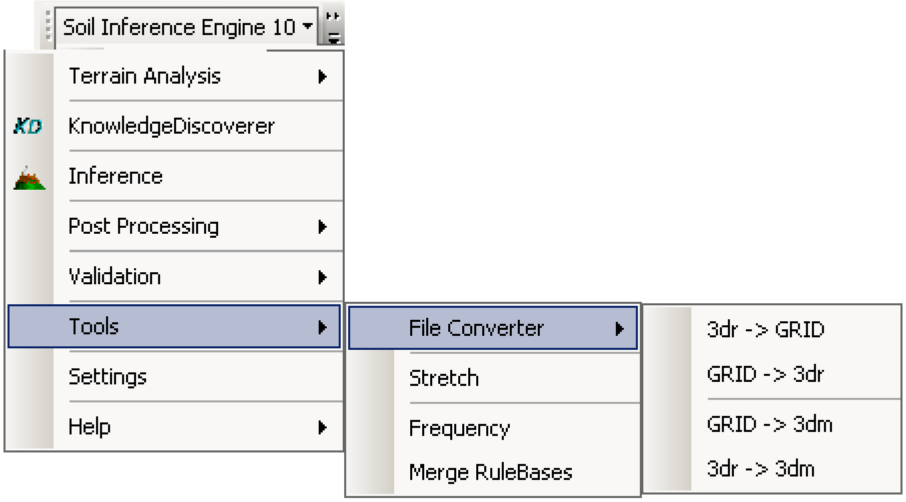
- Coordinates for defining the *Sampling Region*.

4.3.4. Regional Random

This method picks random locations within polygons in a Shapefile specified by the user. The user must specify a field in the attribute table of the Shapefile that contains the number of samples to be put into each polygon.

Chapter 5. Tools

This menu is a collection of tools for some special purposes. The submenus of the tools are illustrated as below:



5.1. File Converters

These tools convert data between formats of ArcGIS (Grid) and 3dMapper (3dm and 3dr).

When converting a 3dr file to an ArcInfo Grid, you can specify to attach the spatial reference (i.e., coordinate system) of the current ArcMap mxd to the new Grid, but it is your responsibility to ensure that the 3dr file is indeed in that coordinate system.

5.2. Stretching

This tool changes the value range of a raster layer to a user-specified range.

The original values will be stretched linearly.

By checking on the *Customize* button, you can specify a particular range in the original values to stretch.

5.3. Frequency

This tool reports number of cells for each unique value in the input raster layer. The output is a text file containing two columns. The first column lists all unique values in the raster, and the second column lists their corresponding number of cells. An example is as follows:

```
0       3993
1       683
2       28286
3       31403
4       30135
```

For an integer raster, this tool outputs the exact same information as that in the attribute table of the raster. This tool also works on a floating point raster, for which attribute table is not available from ArcGIS.

Chapter 6. Knowledge Discoverer

The Knowledge Discoverer (KD) was created primarily for facilitating the map updating process in soil survey. A soil scientist can use it to "discover" the knowledge implicitly represented by an existing soil map and revise the discovered knowledge. As a tool, KD can also be used to conduct *casebase*-to-*rulebase* conversion and data exploration.

Similar to the tools for *cases* in the Inference Engine, a primary function of KD is to overlay vector features over raster layers, and use the cell values that are associated with a vector feature to generate mathematic functions (represented by curves) about that vector feature; it provides tools for the user to revise a curve and save it to a *rulebase*. For example, with a DEM, KD can identify those cells enclosed by a polygon in an existing soil map, and use their elevation values to build a curve about the distribution of elevation within that polygon. The curve would be considered as a representation of the knowledge about the relationship between elevation and the soil represented by that polygon. KD provides this "discovered" knowledge to the soil scientist, who then can revise the knowledge according to new knowledge and/or new data, and save it into a *rulebase*. Later, the soil scientist can use the revised knowledge to generate an *updated* map.

The input vector features can be points, lines, or polygons. For a polygon, the cells that are enclosed by the polygon will be used to calculate statistics and build the curve; for a line, the cells that are passed through by the line will be used to calculate statistics and build the curve; and for a point feature, the cell that the point falls into will be used to build the curve (no statistics to calculate for a single cell). For the rest of this chapter, we will use polygon to explain and illustrate how KD works. However, these explanations and illustrations can also be applied to line and point wherever proper.

The input raster layers are assumed to represent environmental factors that characterize the soil formation environment in the mapping area.

KD offers several ways to create the soil-environment function curves, trying to capture the "core concept" in the knowledge represented by the polygon in an existing soil map.

KD provides the same tools as those in the Inference Engine for examining and editing a curve.

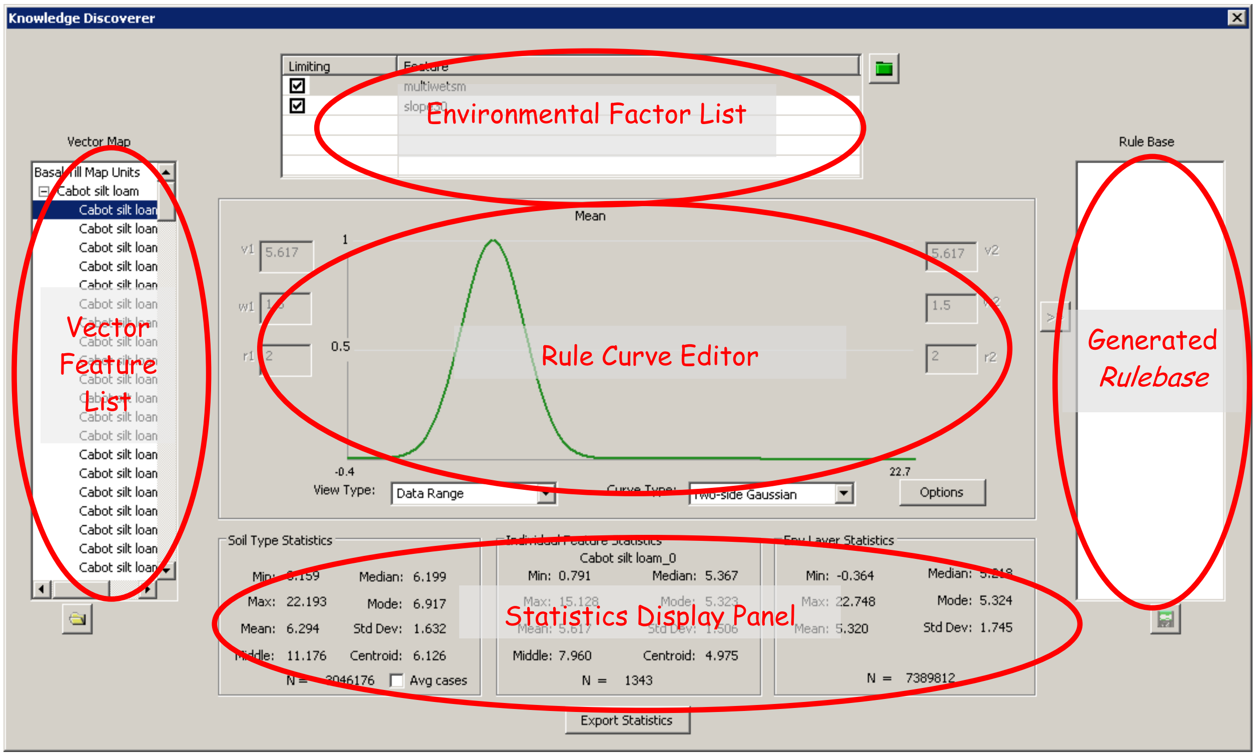
In addition to the tools for *cases* in the Inference Engine, which focus on individual *cases*, KD offers extra functionality to generalize information from individual features. For example, KD can display curves from different polygons for the user to compare and identify general pattern. It also provides tools to mathematically integrate information from different polygons, e.g., merging multiple curves into one.

Besides discovering knowledge, you can also use KD to convert a *casebase* into a *rulebase*, i.e., extracting the fuzzy membership function part of a *casebase* and save it separately. The resulting *rulebase* can then be applied to a different mapping area if appropriate.

KD calculates descriptive statistics for the cell values that are associated with the polygons, which is a useful function for knowledge validation, data exploration, or even sampling design.

6.6. Launching the Knowledge Discoverer

In the Soil Inference Engine menu, clicking *Knowledge Discoverer* opens the KD dialog box:



v1 5.617 w1 1.5 r1 2

v2 5.617 w2 1.5 r2 2

View Type: Data Range

Curve Type: Two-side Gaussian

Soil Type Statistics

Min: 0.159 Median: 6.199
Max: 22.193 Mode: 6.917
Mean: 6.294 Std Dev: 1.632
Middle: 11.176 Centroid: 6.126
N = 2046176 □ Avg cases

Individual Feature Statistics

Cabot silt loam_0
Min: 0.791 Median: 5.367
Max: 15.128 Mode: 5.323
Mean: 5.617 Std Dev: 1.506
Middle: 7.960 Centroid: 4.975
N = 1343

Env Layer Statistics

Min: -0.364 Median: 5.218
Max: 22.748 Mode: 5.324
Mean: 5.320 Std Dev: 1.745
N = 7389812

6.7. Preparing the *Environmental Database*

KD works between knowledge represented by polygons and environmental data in the form of raster layer. Therefore, for KD to work, environmental data layers must be loaded to build the *environmental database*, just like that with the Inference Engine.

6.7.1. Load environmental data layers

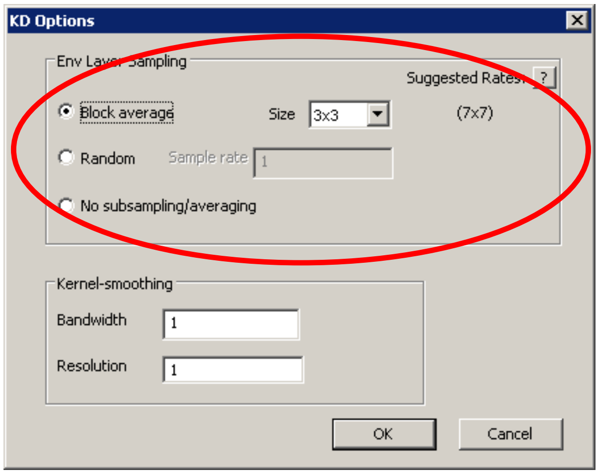
To build the *environmental database*, click 🟩 to launch the Environmental Data Layer Editor dialog box. See Preparing *environmental database* for the usage of the Environmental Data Layer Editor.

All the raster layers of the environmental data must have the same dimensions (number of columns and number of rows), same resolution (cellsize), and same coordinate system. In other words the raster/grid layers much be snapped on top of each other. The spatial setting set by the Spatial Setting button does not have effect on the operations in KD.

6.7.2. Sample the environmental data

When the raster layers of the environmental data are large and/or the number of polygons is large, the calculation for creating the curves can be intensive and slow. To speed up the operation, you can choose to use sampled environmental data instead of the original data.

In the Environmental Data Layer Editor, when you load the first raster layer, the program will check the size of the layer and make suggestion about sampling. If the program considers that the raster layer is large and sampling is necessary, the KD Option dialog box will pop up (below).



● Block average Size 3x3 (7x7)

○ Random Sample rate 1

○ No subsampling/averaging

Bandwidth 1

Resolution 1

The upper part of this dialog box is about sampling. It shows the currently used sampling strategy and rate, as well as the rate suggested by the program. The specification in this dialog box will be applied to all the raster layers in the *environmental database*. After the *environmental database* is created, you can click Options to open this dialog box and change the sampling setting at any time.

6.7.2.1. Options for sampling strategy

- *Block average*: This strategy aggregates values of the original cells into larger *blocks*. The method of aggregation is simple averaging. It essentially reduces the resolution of the raster layer, and the resulting low-resolution raster will be used in the later calculation of statistics and other operations. The primary advantage of this strategy is that it ensures that the outputs are identical as long as the sampling setting remains the same. For example, two operations using 3×3 setting to process a raster layer will result in the same calculated statistics. KD provides five options for defining the block, including 3×3, 5×5, 7×7, 9×9, and 11×11.
- *Random*: This sampling strategy randomly selects cells from the original raster according to the specified sampling rate, and the selected cells will be used in in the later calculation of statistics and other operations. While random sampling is a real and more formal sampling, a problem with it is that two samplings may result in different statistics.
- *No subsampling/averaging*: This indicates no sampling should be performed, and the original raster will be used in the later calculation of statistics and other operations.

6.7.2.2. Default sampling rate

According to the size of the raster layer, KD makes suggestion about sampling rate. The *suggested* rate is determined as follows:

$$k = \log_{10}(\text{number of columns} \times \text{number of rows})$$

Round k to integer. For the *random sampling* option, the suggested sampling rate will be:

$$r = 1/10^{(k\text{-}5)}$$

For example, if the size of the raster is 5000×5000, k will be 7, and the sampling rate will be 1/100.

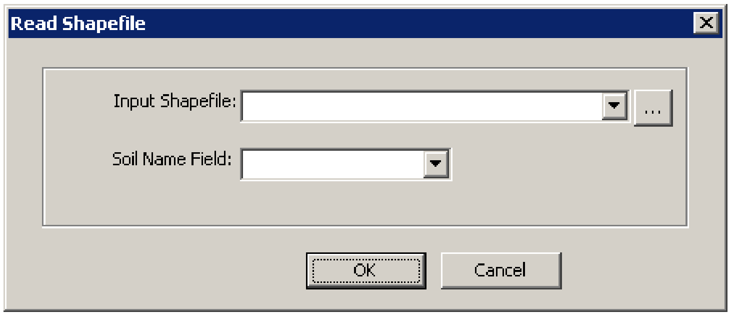
For the *block average* option, the program will choose a block size whose sampling rate is in the same *order* as that calculated using the method above. The following table provides specifics:

Calculated Rate	***Suggested Block Size***
1/10	3×3
1/100	7x7
1/1000	11×11

The program will not suggest 5×5 and 9×9, but you can specify to use either of them as you wish.

6.8. Loading Vector Features

A polygon layer loaded to KD is supposed to be an existing soil map for the mapping area, and the goal of using KD is to *update* this map with the updated knowledge and/or environmental data.

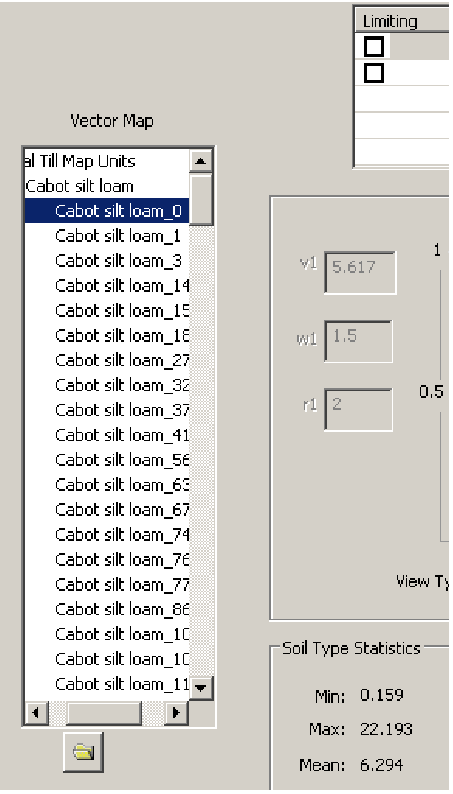

Read Shapefile

Input Shapefile:

Soil Name Field:

Click 📂 under the "Vector Feature List" pane to open a dialog box as follows, from which you specify the Shapefile to load and the field that contains *soil type* name (left).

The attribute table of the Shapefile must have a text field for *soil type* names. KD automatically lists all the text fields in the Shapefile for you to choose. If no text field is found, a warning message will appear and the vector layer will not be loaded.). For example, the user can join the "mapunit" table from the tabular SSURGO that has the "muname" which provides information about the *soil type* names. This table can be joined to the soil polygon shape file via "mukey". The loaded polygons will appear in the feature list pane (below).



al Till Map Units
Cabot silt loam
Cabot silt loam_0
Cabot silt loam_1
Cabot silt loam_3
Cabot silt loam_14
Cabot silt loam_15
Cabot silt loam_18
Cabot silt loam_27
Cabot silt loam_32
Cabot silt loam_37
Cabot silt loam_41
Cabot silt loam_56
Cabot silt loam_63
Cabot silt loam_67
Cabot silt loam_74
Cabot silt loam_76
Cabot silt loam_77
Cabot silt loam_86
Cabot silt loam_10
Cabot silt loam_10
Cabot silt loam_11

v1 5.617

w1 1.5

r1 2

Soil Type Statistics

Min: 0.159

Max: 22.193

Mean: 6.294

All the vector features that have the same *soil type* name are considered to represent the same soil and will be grouped together in the feature list. This list has the exactly same structure as that of a casebase.

In the feature list, each polygon has a unique name in the form of *soil type_FID*, e.g. *Cabot silt loam_1*. The FID in the name is the FID in the attribute table of the Shapefile, which establishes correspondence between the curves in KD and the polygons in ArcMap.

For each polygon, KD calculates statistics of the cells in each environmental layer that are enclosed by the polygon, and builds curves based on the statistics. Depending on the number of the polygons, size of the raster layers, and the sampling rate, the calculation of statistics may take long to complete.

When block-averaging sampling is used, the formed *blocks* (eventually larger cells) will replace the original cells in calculating statistics and building curves. When random sampling is used, the sample set of cells will be used to calculate statistics and build curve.

6.9. View the Curve and Statistics

Like the Inference Engine, KD uses curves to visualize soil-environment relationships and edit soil-environment models. Specifically for KD, the curves may represent the knowledge contained in the existing soil maps, and provide a graphic means for the soil scientist to update the knowledge.

6.9.1. View individual curve

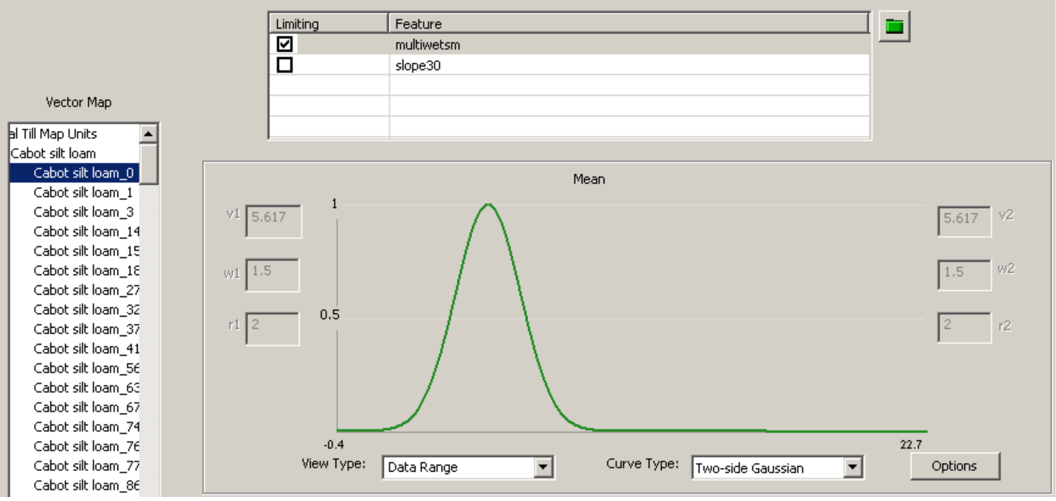
Select a polygon in the feature list, and check on an environmental factor, the corresponding curve will appear in the curve editing area.

KD can automatically build two types of curves: two-side Gaussian optimality curve and kernel-smoothed frequency curve. Once the two-side Gaussian curve is built, you can modify it to other forms, including S-shaped, Z-shaped, nominal, ordinal, and cyclic.

6.9.1.1. Two-side Gaussian optimality curve

6.9.1.1.1. Overview

This type of curve is defined with the same parameters as those used in the Inference Engine, including *v*, *w*, and *r* on both sides of the curve:



Limiting	Feature
☑	multiwetsm
□	slope30

v1 5.617
w1 1.5
r1 2
v2 5.617
w2 1.5
r2 2
View Type: Data Range
Curve Type: Two-side Gaussian

For knowledge discovering purposes, KD provides special options for setting initial top value (*v*) of the curve. The top value represents the most optimal condition of the environmental factor for the formation or existence of the given soil.

6.9.1.1.2. Options for the top value (v)

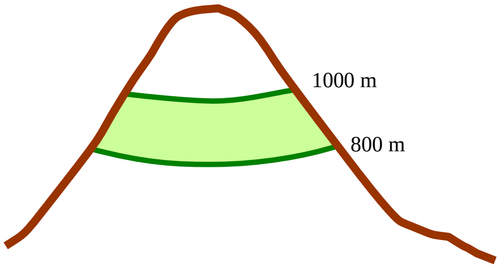
Mean (average) of all the cells enclosed by the polygon might be the first choice for the top value (*v*). However, mean was not necessarily the most optimal environmental value in the soil scientist's mind when he or she was delineating the soil polygon. For example, the soil scientist may consider the value at the geometric center of the polygon to be the most optimal; or the soil scientist may have a value range in mind defined by a minimum and a maximum, and thus the middle value of the range can better represent the most optimal value.

Generally, once the soil scientist has formalized a soil-environmental model, he or she may have three ways to implement it, either implicitly or explicitly, during the delineation of a polygon:

- The *numerical* way: The soil scientist may have a value range in mind for the environmental factor, defined by a minimum and a maximum. For example, he or she may consider that the slope gradient 0-8% is the range for soil X. When delineating the polygon, he or she tries to enclose those areas that fit into this range. In this case, the middle value of the range might be a good value to characterize the most optimal condition in the soil-environment model. Alternatively, the soil scientist may have a single value or fairly narrow value range for the environmental factor in the model. In this latter case, the most frequently occurring value might be a good choice.
- The *geographic* way: The soil scientist may see typical areas or locations for the established soil-landscape model, and delineate polygons around those areas or locations. In this case, the value at the geometric centroid of the polygon might be a good choice.
- The *statistical* way: The soil scientist delineates the polygon and he or she knows or tries to achieve that certain statistical properties of the environmental values within the polygon, e.g., mean or median, meet the requirements of the soil-environment model.

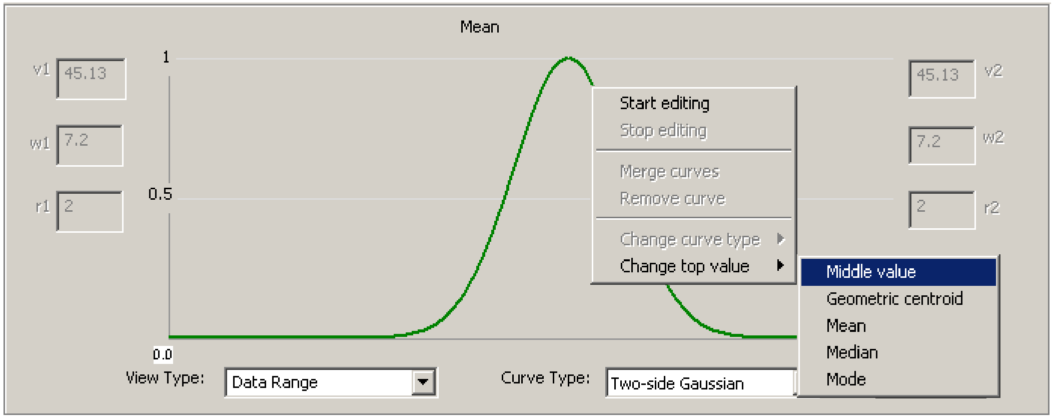
These different approaches may be consistent in some situations, and may be not in others. For example, if the values within a polygon have a normal distribution, the numerical way and the statistical way should lead to very similar outputs. However, if the frequency distribution is complicated, different methods may generate quite different outputs. Different options in the same approach also lead to different outputs (e.g., mean is usually different from median).

What exact approach was used by the soil scientist who created the existing soil map is usually not well documented (which is a well-known issue with the traditional soil mapping based on tacit knowledge). In addition, it is likely that the soil scientist applied more than one approach to a single soil map or even a single polygon. You may have to make reasonable guesses about how the soil scientist implemented the soil-environment model when delineating the polygon, based on your own knowledge and experience, and by exploring and evaluating different options.



The green polygon in the left graphic, representing a cross-section of a landform, is likely to have been defined by a range of elevation, and thus the middle value 900 m should be the best choice for the top value of its curve. Any other method may lead to a biased representation of the knowledge.

KD provides several options for the top value to represent different approaches. To change the top value, right-click on the curve, and in the popped-up menu, select *Change top value* to open the submenu of top value options to make selection (below):



v1 45.13
w1 7.2
r1 2
v2 45.13
w2 7.2
r2 2
View Type: Data Range
Curve Type: Two-side Gaussian

- ***Middle value***: The value that is right in the middle between the minimum and maximum of all the cell values that are enclosed by the current polygon. In other words, it is the average of the minimum and maximum values. Note this is not the median value.
- ***Geometric centroid***: The value of the cell that is at the geometric centroid of the current polygon. Geometric centroid may be a representative location for a polygon with low perimeter/area ratio (close to a circle). It may be less representative for a polygon with elongated or very complex shape. If the cell at this location has nodata, the *Middle value* will be used for this option so that the curve can still be created.
- ***Mean***: The simple average of all the cell values that are enclosed by the current polygon.
- ***Median***: The median value of all the cell values that are enclosed by the current polygon.

- ***Mode***: The mode value, i.e., the value that has the highest frequency, of all the cell values that are enclosed by the current polygon.

Once you choose an option for the top value, both *v1* and *v2* will be set with the chosen value.

6.9.1.1.3. Other parameters

The other parameters for defining the Gaussian curve, including the two sets of *w* and *r*, have exactly the same meanings as *w* and *r* in the Inference Engine.

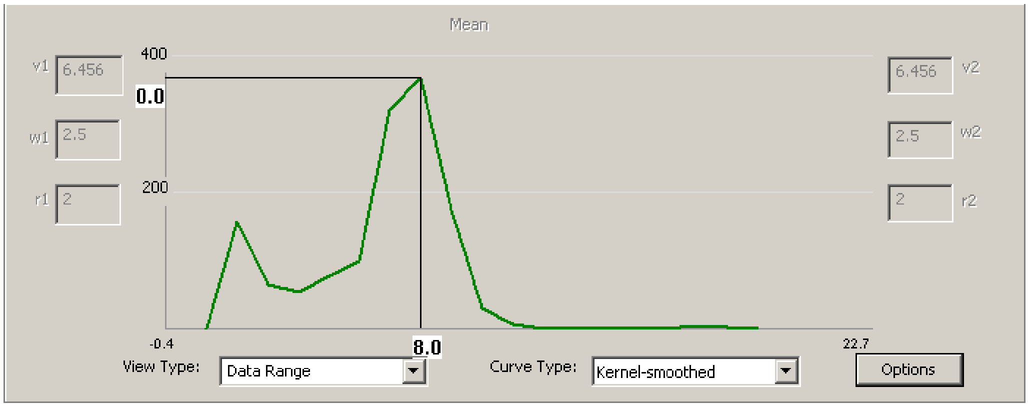
6.9.1.2. Frequency curve

6.9.1.2.1. Overview

Besides Gaussian curve, KD implements frequency curve as another way to discover and represent the knowledge of soil-landscape relationships implied by the polygons in the existing soil map. A frequency curve shows the frequencies of different cell values within a given polygon.

In KD, the assumption underlying the use of frequency curve is that the more a value of an environmental factor occurs within a polygon, the more optimal the value is for the formation of the soil represented by the polygon. For example, if in a polygon of soil X we see that the cells with slope gradient value 10% are more than the cells with slope gradient value 35%, we assume that the soil scientist who drew the polygon was considering that 10% slope gradient is more optimal than 35% slope gradient for soil X.

In the left graphic, the horizontal axis represents the value of the currently selected environmental factor, and the vertical axis represents frequency, measured by number of cells. For an environmental factor with continuous values, the number of cells with a unique value can be very small, and thus directly using the number of cells for each unique value may not result in a meaningful frequency curve. The way to get around this problem is to use a range instead of a single value when counting the number of cells, which is called the *kernel method*.



v1 6.456
w1 2.5
r1 2
v2 6.456
w2 2.5
r2 2
View Type: Data Range
Curve Type: Kernel-smoothed

To find out the corresponding values at each point on the frequency curve, right-click on the point and the values will appear. The graphic above shows that in the given polygon, there are about 350 cells having values in the range around 8.0 for the currently selected environmental factor.

Compared with Gaussian curve, which is a highly modeled and thus simplified representation of the environmental situation in a polygon, a frequency curve can provide more detailed characterization. For example, from the frequency curve in the above graphic, we see that the environmental factor has two value peaks, although one is higher than the other; also, the frequency distribution of the values is not symmetric. These features cannot be represented by a Gaussian curve.

However, a drawback of frequency curve is that it is a nonparametric approach, which means that it is not using a simple mathematical model (e.g., a Gaussian function) to model the information. In other words, it cannot be directly translated into a *rule*, like with a Gaussian curve. For the same reason, a frequency curve is not editable, i.e., adjusting parameters *v*, *w*, and *r* does not have effect on it. A frequency curves created by KD is mainly for you to visually inspect the details of the environmental condition associated with a polygon, so as to make a better evaluation and decision about the discovered knowledge.

For example, if within a polygon there are multiple peaks in the values of an environmental factor, you may consider these possibilities:

- This particular environmental factor may not have a tangible relationship with this particular soil. In other words, the factor is not a good predictor for the soil.
- This polygon does not well follow the established relationship between the environmental factor and the soil. In other words, the quality of this polygon is questionable.
- The relationship between the factor and the soil is complicated and cannot be well represented by a simple model. The rule formed based on the Gaussian curve may not have a good quality and multiple *instances* should be considered. You may also want to consider using case-based reasoning. This is particularly important when one considers multiple component soil map units such as complexes and associations, or pure soil map units with minor inclusions. In such cases, the multi-peak frequency curve could be an indication of the presence of one or more components/soil series within the map unit polygon with distinctive relationships with the terrain/environmental factors. The user is encouraged to read about the soil map unit composition and in particular the soil component-landscape relationships, often described in the published soil surveys to find out about these relationships.

6.9.1.2.2. How a frequency curve is built

A frequency curve is created using the *kernel* method. The technical procedure is as follows:

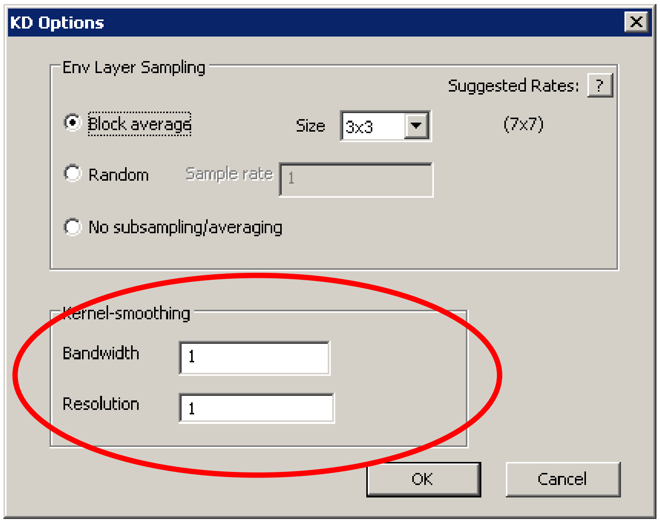
a. Sort all the cell values of the currently selected environmental factor enclosed by the currently selected polygon.
b. Starting from the minimum value, e.g., -0.4, count how many cells fall into a numerical range around this value, and use that number of cells as the frequency value for -0.4. This range is called *kernel* in statistics, and the parameter ***bandwidth*** determines how large the kernel is. For example, if bandwidth = 0.8, then the range is -1.2 ~ 0.4 (i.e., -0.4-0.8 ~ -0.4+0.8); if the number of cells whose values are within [-1.2, 0.4] is 134, then 134 is the frequency value for -0.4. The kernel is set around the specified value, so for the minimum value, its kernel actually does not have the left side, because there is no value less than the minimum value. This kind of bias or error is called the *edge effect*.
c. After determining frequency for the minimum value, the program moves to the next value and performs the same operation, i.e., the kernel is moving forward. How far to move is determined by the parameter ***resolution***. For example, if resolution = 0.1, then the next value is -0.4 + 0.1 = -0.3, so the kernel is to be set at -0.3 and frequency is to be calculated for [-1.1, 0.5]. The kernel keeps moving until the maximum value is reached.

6.9.1.2.3. Adjusting frequency curve

By adjusting *bandwidth* and *resolution*, you can adjust how information is to be presented by the frequency curve. In this adjustment, you are trying to achieve two balances:

- *Balance between specifics and general trend*. This is achieved by adjusting *bandwidth*. Increasing bandwidth is a way of generalization, which will conceal specifics and present more general trend. Visually, increasing bandwidth results in smoother curves. A too large bandwidth, however, may conceal too much specific information and lose the point of creating a frequency curve. To the extreme, if the bandwidth is as large as the range from the minimum to the maximum, the frequency curve will be a simple level line, indicating all cells are the same. On the other hand, if bandwidth is too small, the curve will present too much specific information and the general trend hardly emerges, which also loses the point of creating the frequency curve.
- *Balance between details and computation burden*. This is achieved by adjusting *resolution*. Specifying a large value for *resolution* speeds up the operation, but a too large value for resolution may result in a too coarse curve that may miss the precise value of a critical threshold or turning point you want to capture.

There are no predefined rules of thumb for specifying optimal values for bandwidth and resolution. The adjustment is basically a trial-and-error process. After all, frequency curve itself is a means for exploration and investigation. The goal in the operations with frequency curve is not to create the "best" frequency curve, but to use frequency curve to better learn and understand the environmental conditions within the polygon, and most importantly how they relate to soil distribution in the landscape.



KD Options

Env Layer Sampling

Suggested Rates: ?

● Block average Size 3x3 (7x7)

○ Random Sample rate 1

○ No subsampling/averaging

Kernel-smoothing

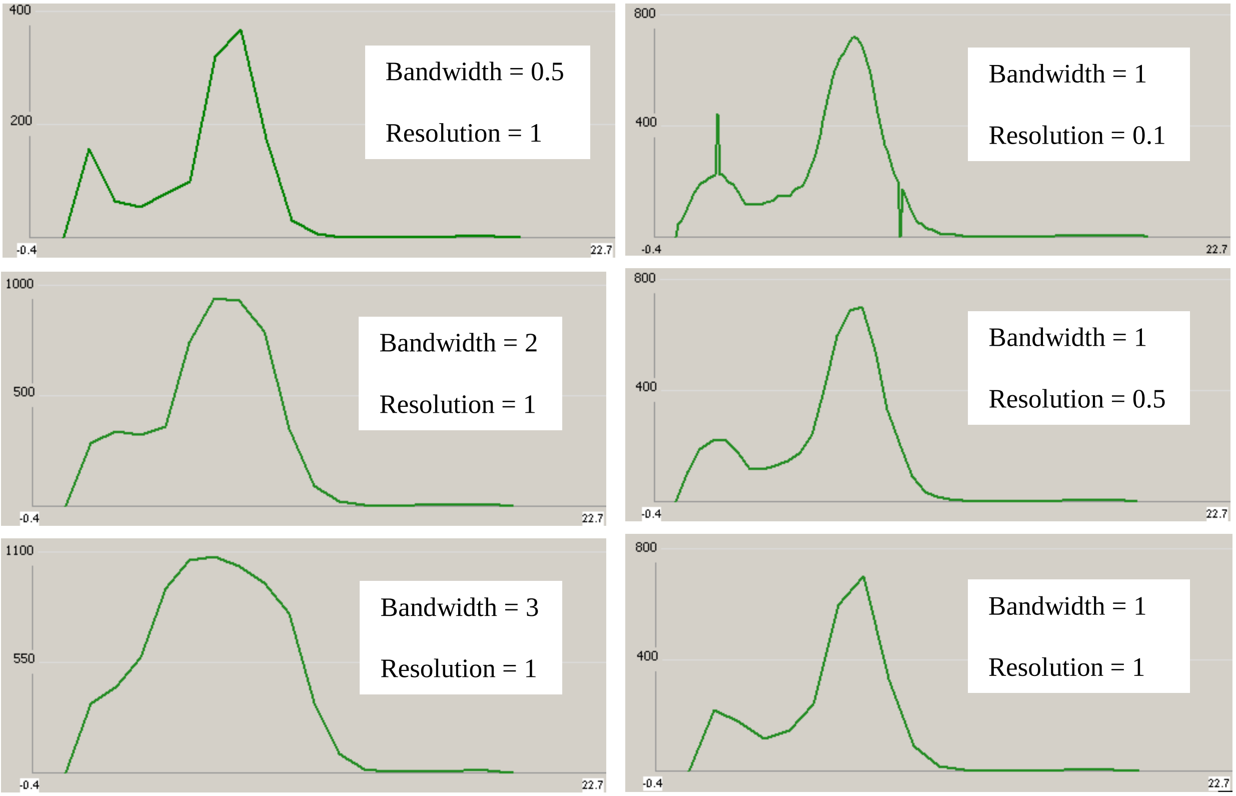
Bandwidth 1

Resolution 1

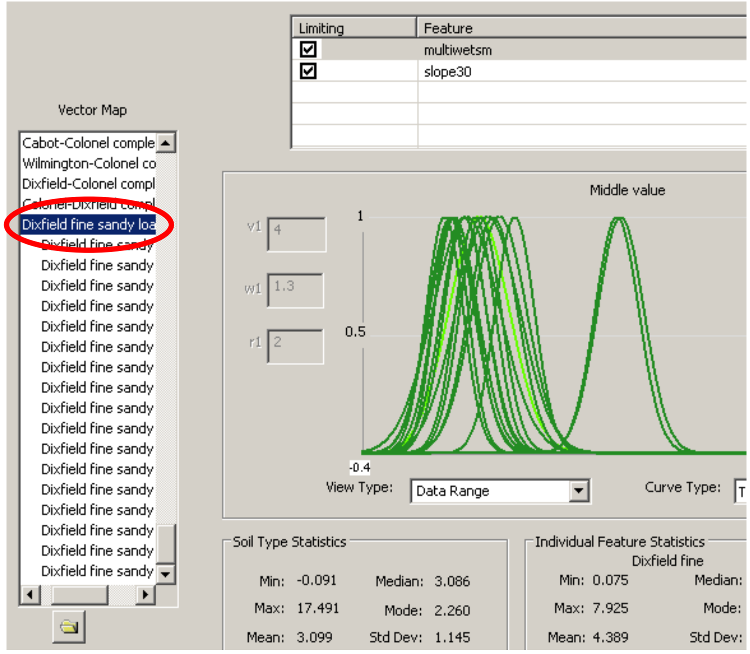
To change the values for the two parameters, click Options to open the Option dialog box. The lower part of this dialog box is for setting parameters for frequency curve (left).

Bandwidth must be equal or greater than 0.5*resolution. Otherwise, there will be gaps, i.e., some cell values are not covered by the frequency curve. On the other hand, apparently the range (kernel) at each value has overlaps with its neighbors. Therefore the numbers of cells on the frequency curve do not sum up to the total number of cells enclosed by the polygon.

In the graphics below, the three on the left show the effect of bandwidth, and the three on the right show the effect of resolution.



6.9.2. View all curves of a *soil type*



Limiting	Feature
☑	multiwetsm
☑	slope30

Vector Map

- Cabot-Colonel comple
- Wilmington-Colonel co
- Dixfield-Colonel compl
- Colonel-Dixfield compl
- Dixfield fine sandy loa
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy
  - Dixfield fine sandy

v1: 4
w1: 1.3
r1: 2

View Type: Data Range
Curve Type: T

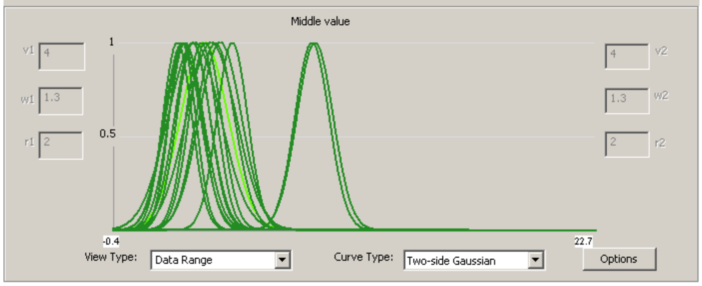
Soil Type Statistics

Min: -0.091 Median: 3.086
Max: 17.491 Mode: 2.260
Mean: 3.099 Std Dev: 1.145

Individual Feature Statistics
Dixfield fine

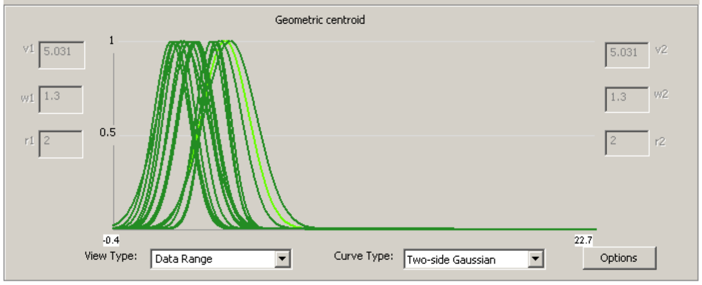
Min: 0.075 Median:
Max: 7.925 Mode:
Mean: 4.389 Std Dev:

You may want the curves of all the polygons belonging to the same *soil type* to be displayed together, so that you can quickly see how they are in common and if there are outliers. This can be achieved by clicking on the *soil type* instead of an individual polygon (left).


Middle value

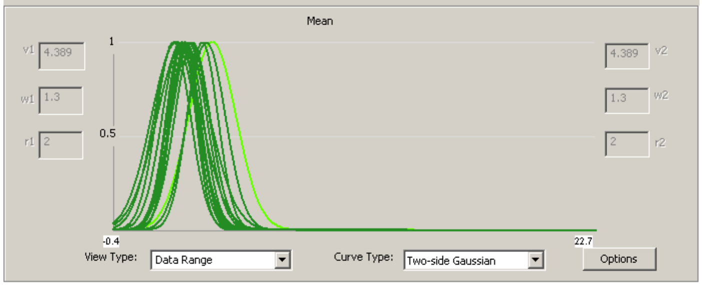
v1 4
w1 1.3
r1 2
v2 4
w2 1.3
r2 2

View Type: Data Range
Curve Type: Two-side Gaussian


Geometric centroid

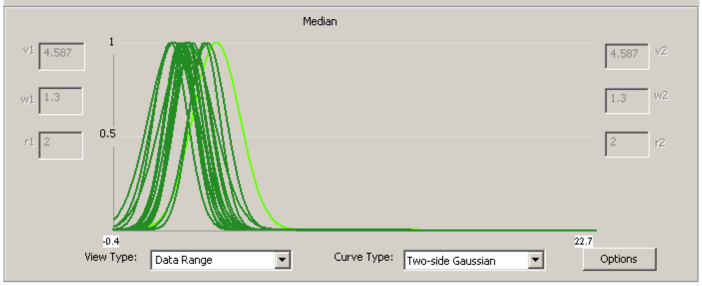
v1 5.031
w1 1.3
r1 2
v2 5.031
w2 1.3
r2 2

View Type: Data Range
Curve Type: Two-side Gaussian


Mean

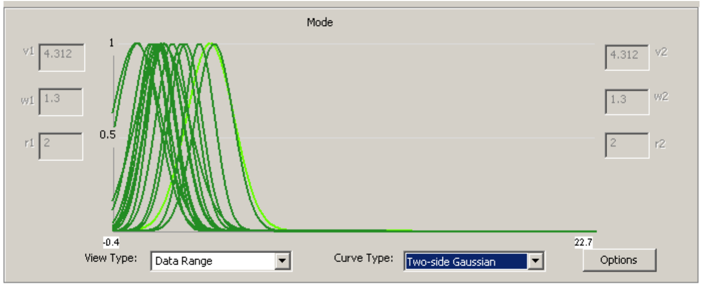
v1 4.389
w1 1.3
r1 2
v2 4.389
w2 1.3
r2 2

View Type: Data Range
Curve Type: Two-side Gaussian


Median

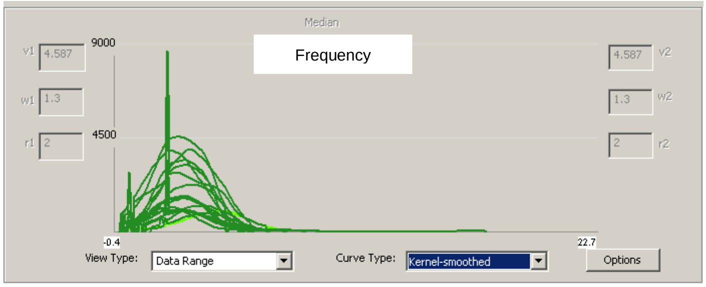
v1 4.587
w1 1.3
r1 2
v2 4.587
w2 1.3
r2 2

View Type: Data Range
Curve Type: Two-side Gaussian


Mode

v1 4.312
w1 1.3
r1 2
v2 4.312
w2 1.3
r2 2

View Type: Data Range
Curve Type: Two-side Gaussian


Median

Frequency

v1 4.587
w1 1.3
r1 2
v2 4.587
w2 1.3
r2 2

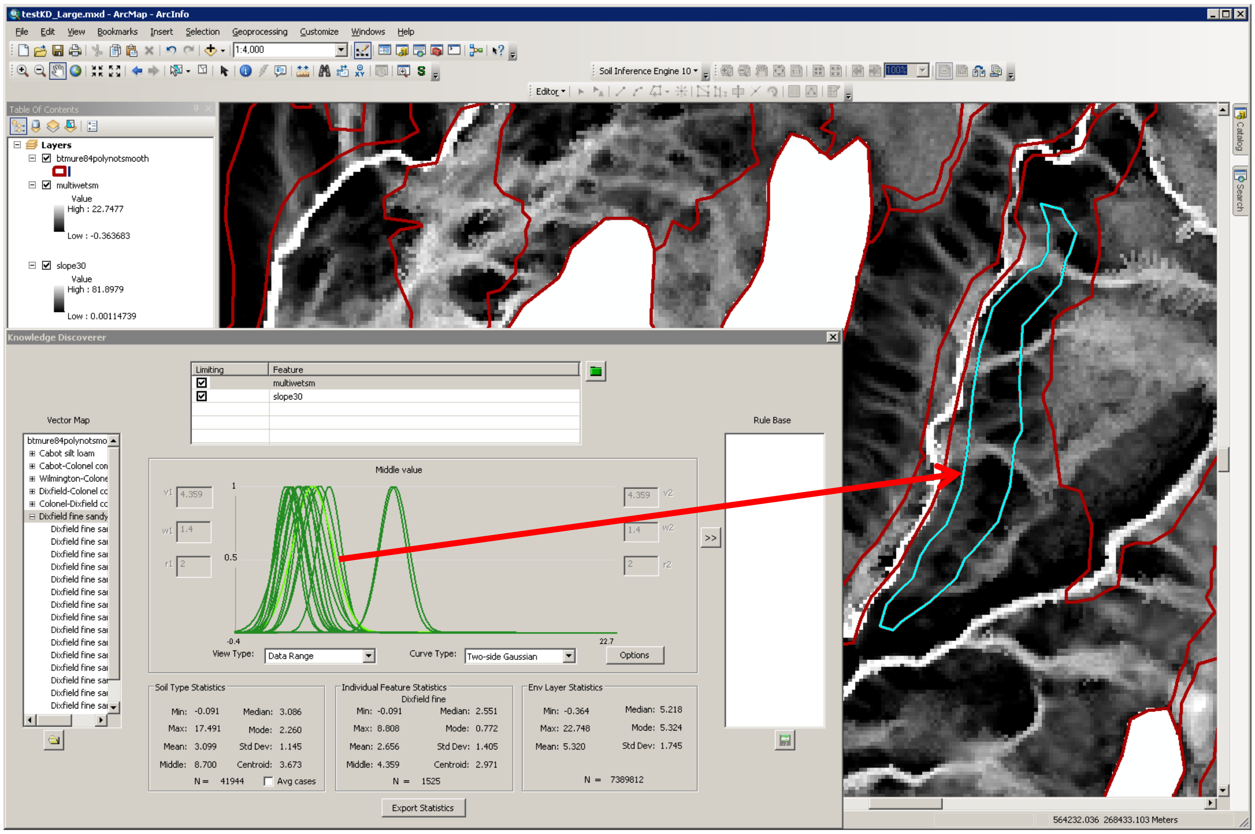
View Type: Data Range
Curve Type: Kernel-smoothed

In this group display, you can find out the top value of a specific curve by clicking on the curve to select it. The selected curve will be highlighted and its top value will be shown in the *v1* and *v2* fields.

The graphics on the left show an example of group display. The curves in these graphics are all for the same *soil type*, but are under different options for the top value and curve type (Gaussian vs. frequency). The selected curves in all these graphics are from the same polygon. It is apparent that the commonality and variance, as well as outliers, of these curves are different under different top value options. For example, the selected curve seems to be the most average one under the middle-value option, but becomes a marginal one under the mean and median options. This again prompts the question: How did the soil scientist implement the soil-environment model when he or she drew this polygon? This example demonstrates a use of the group display.

6.9.3. Corresponding a curve to its polygon in the map

When you select a curve in KD, the corresponding polygon in ArcMap will be automatically selected and highlighted as well. This correspondence between the information in KD and the data in ArcMap allows you to visually inspect the numerical distribution and geographic distribution of the environmental values within a polygon at the same time.



Limiting	Feature
☑	multiwetsm
☑	slope30

v1: 4.359 w1: 1.4 r1: 2
v2: 4.359 w2: 1.4 r2: 2

View Type: Data Range
Curve Type: Two-side Gaussian

Soil Type Statistics
Min: -0.091 Median: 3.086
Max: 17.491 Mode: 2.260
Mean: 3.099 Std Dev: 1.145
Middle: 8.700 Centroid: 3.673
N = 41944 □ Avg cases

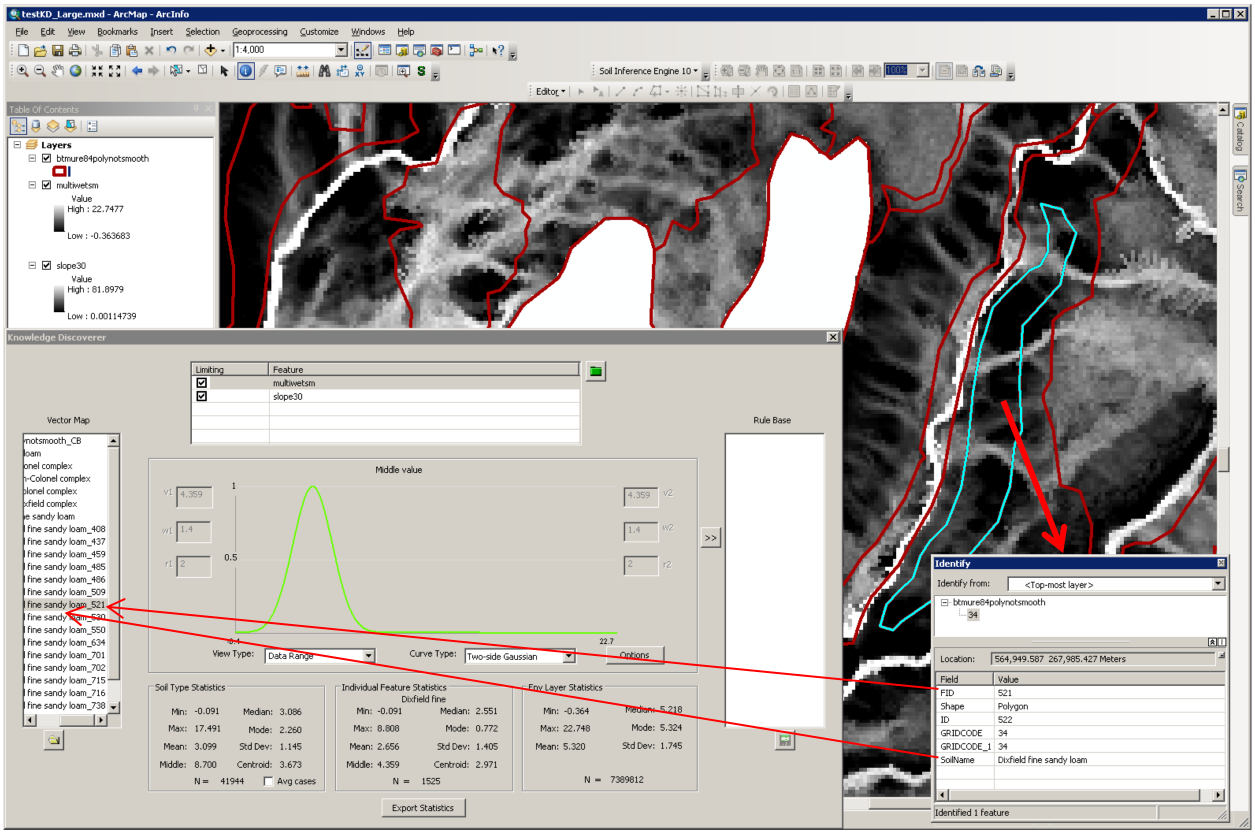
Individual Feature Statistics
Dixfield fine
Min: -0.091 Median: 2.551
Max: 8.808 Mode: 0.772
Mean: 2.656 Std Dev: 1.405
Middle: 4.359 Centroid: 2.971
N = 1525

Env Layer Statistics
Min: -0.364 Median: 5.218
Max: 22.748 Mode: 5.324
Mean: 5.320 Std Dev: 1.745
N = 7389812

In the current version of KD, this correspondence is one-way only, i.e., if you select a polygon in ArcMap, KD is not able to automatically identify its corresponding curve. However, you can manually build the ArcMap→KD connection:

a. In ArcMap, find out the *soil name* and FID of the given polygon (e.g. using the ⓘ tool of ArcMap).
b. Use the *soil name* and FID to identify the corresponding polygon in KD by checking the polygon name under the proper *soil type*.

This procedure is illustrated by the graphic in next page.


testKD_Large.mxd - ArcMap - ArcInfo
File Edit View Bookmarks Insert Selection Geoprocessing Customize Windows Help
1:4,000
Soil Inference Engine 10
Editor
Table Of Contents
Catalog
Search

Layers
☑ btmure84polynotsmooth
☑ multiwetsm
Value
High : 22.7477
Low : -0.363683
☑ slope30
Value
High : 81.8979
Low : 0.00114739

Knowledge Discoverer

Limiting	Feature
☑	multiwetsm
☑	slope30

Vector Map
notsmooth_CB
loam
onel complex
-Colonel complex
olonel complex
xfield complex
e sandy loam
l fine sandy loam_408
l fine sandy loam_437
l fine sandy loam_459
l fine sandy loam_485
l fine sandy loam_486
l fine sandy loam_509
l fine sandy loam_521
l fine sandy loam_530
l fine sandy loam_550
l fine sandy loam_634
l fine sandy loam_701
l fine sandy loam_702
l fine sandy loam_715
l fine sandy loam_716
l fine sandy loam_738

Middle value
v1 4.359 v2 4.359
w1 1.4 w2 1.4
r1 2 r2 2

View Type: Data Range
Curve Type: Two-side Gaussian

Soil Type Statistics
Min: -0.091 Median: 3.086
Max: 17.491 Mode: 2.260
Mean: 3.099 Std Dev: 1.145
Middle: 8.700 Centroid: 3.673
N = 41944 □ Avg cases

Individual Feature Statistics
Dixfield fine
Min: -0.091 Median: 2.551
Max: 8.808 Mode: 0.772
Mean: 2.656 Std Dev: 1.405
Middle: 4.359 Centroid: 2.971
N = 1525

Env Layer Statistics
Min: -0.364 Median: 5.218
Max: 22.748 Mode: 5.324
Mean: 5.320 Std Dev: 1.745
N = 7389812

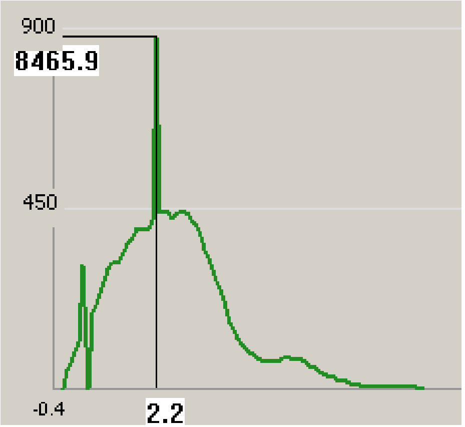
Export Statistics

Identify
Identify from: <Top-most layer>
btmure84polynotsmooth
34
Location: 564,949.587 267,985.427 Meters

Field	Value
FID	521
Shape	Polygon
ID	522
GRIDCODE	34
GRIDCODE_1	34
SoilName	Dixfield fine sandy loam

Identified 1 feature

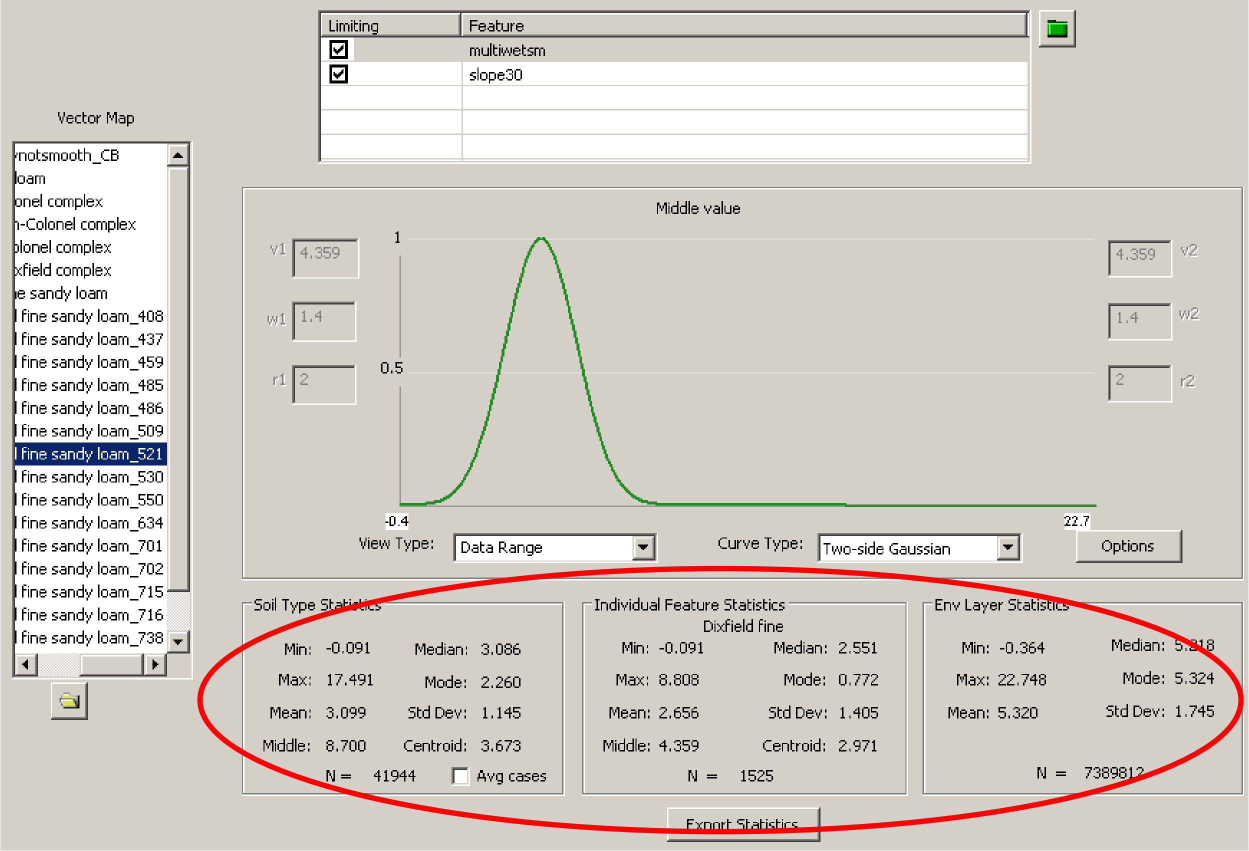
In the example shown above, it appears that the polygon largely falls into low-value areas of the environmental factor, but does not follow the boundary of the value variation very well, and there are several high-value intrusions in the polygon. Compare the map with the corresponding frequency curve (below, created with bandwidth = 0.5 and resolution = 0.1), we see that there is a steep and narrow peak at 2.2 and another small peak at the lower end of the curve. Based on these observations, we may consider that a z-shaped curve might best characterize the relationship between the soil and this environmental factor, which is also supported by the group display where most other curves are on the lower side of the selected curve; also, the output rule should have a *w2* value that ensures 2.2 to haves a high (if not full) optimality value. We may also anticipate that, if we give this particular environmental factor a dominant importance in the soil inference, then in the updated map, which is generated with the revised knowledge, this polygon may shift to the right a little to better meet the variation boundary; the upper end of the original polygon, which falls into a relatively high-value area, may get cut off; and it is likely that this polygon will be split into several smaller ones due to those high-value intrusions.



6.9.4. Descriptive Statistics

6.9.4.1. View the statistics

The Knowledge Discoverer calculates descriptive statistics for the cell values that are associated with the polygons. The statistics are calculated for both individual polygons and the groups of polygons of *soil types*. As a benchmark, it also calculates statistics for the entire raster layer of the environmental factor. These statistics are displayed by three panels:



Limiting	Feature
☑	multiwetsm
☑	slope30

Vector Map

```
notsmooth_CB
loam
onel complex
n-Colonel complex
olonel complex
xfield complex
e sandy loam
l fine sandy loam_408
l fine sandy loam_437
l fine sandy loam_459
l fine sandy loam_485
l fine sandy loam_486
l fine sandy loam_509
l fine sandy loam_521
l fine sandy loam_530
l fine sandy loam_550
l fine sandy loam_634
l fine sandy loam_701
l fine sandy loam_702
l fine sandy loam_715
l fine sandy loam_716
l fine sandy loam_738
```

Middle value

v1 4.359 v2 4.359
w1 1.4 w2 1.4
r1 2 r2 2

View Type: Data Range
Curve Type: Two-side Gaussian

Soil Type Statistics

```
   Min:  -0.091      Median:  3.086
   Max:  17.491        Mode:  2.260
  Mean:  3.099      Std Dev:  1.145
Middle:  8.700     Centroid:  3.673
      N =   41944   □ Avg cases
```

Individual Feature Statistics

```
         Dixfield fine
   Min:  -0.091      Median:  2.551
   Max:  8.808         Mode:  0.772
  Mean:  2.656      Std Dev:  1.405
Middle:  4.359     Centroid:  2.971
          N =    1525
```

Env Layer Statistics

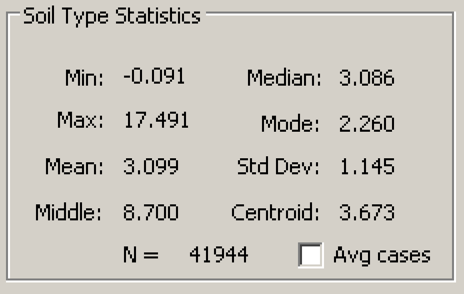
```
  Min:  -0.364       Median:  5.318
  Max:  22.748         Mode:  5.324
 Mean:  5.320       Std Dev:  1.745

               N =   7389812
```

- *Individual Feature Statistics*: This panel contains the statistics of the cell values in the currently selected environmental layer that are enclosed by the currently selected polygon. In the above example, the raster layer is "multiwetsm" (wetness index) and the polygon is "Dixfiled fine sandy loam_521". N is the total number of cells used for calculating the statistics. If sampling is applied, N is the number of sample cells. Note that N does not include the cells whose values are *nodata*.
- *Soil Type Statistics*: This panel contains the statistics of the currently selected environmental layer for the currently selected *soil type*. In the above example, the currently selected raster layer is "multiwetsm" (wetness index) and the currently selected *soil type* is "Dixfield fine sandy loam". The (geographic)

centroid value of a *soil type* is the mean of the centroid values of all its polygons. For the other statistics of *soil type*, KD implements two ways for the calculation:

- *Cell-lumping*: This method lumps all the cells enclosed by the polygons of the currently selected *soil type* and calculate statistics based on those cells. This is the default method. If this method is used, N is the total number of cells used for calculating the statistics. If sampling is applied, N is the number of sample cells. Note that N does not include the cells whose values are *nodata*.
- *Polygon-averaging*: This method first calculates statistics for each individual polygon belonging to the currently selected *soil type*, and then uses the simple average of the polygon-level statistics as the statistics of the *soil type*. To choose this option, check □ Avg cases. If this method is used, N is the number of polygons used for calculating the statistics.

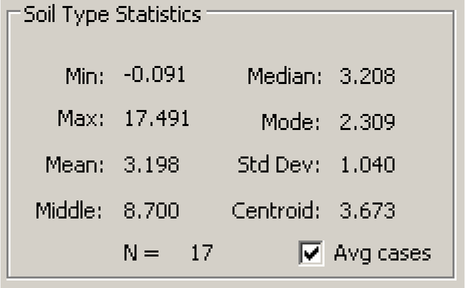
The graphic below shows an example of the difference between these two calculation methods:



Soil Type Statistics

Min:	-0.091	Median:	3.086
Max:	17.491	Mode:	2.260
Mean:	3.099	Std Dev:	1.145
Middle:	8.700	Centroid:	3.673

N = 41944 □ Avg cases



Soil Type Statistics

Min:	-0.091	Median:	3.208
Max:	17.491	Mode:	2.309
Mean:	3.198	Std Dev:	1.040
Middle:	8.700	Centroid:	3.673

N = 17 ☑ Avg cases

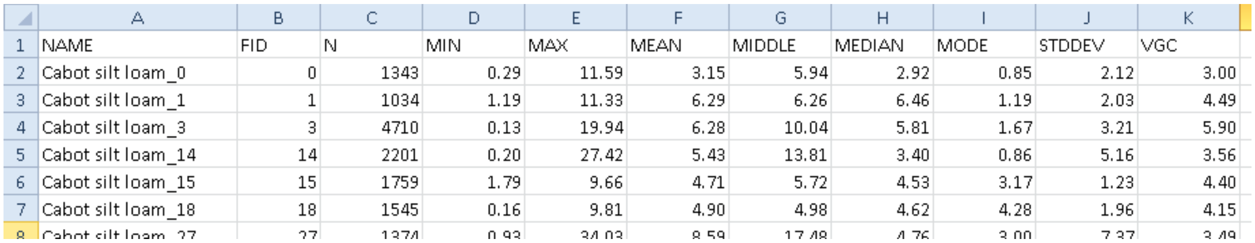
- Env Layer Statistics: This panel contains the statistics of all cells in the currently selected environmental layer.

6.9.4.2. Export the statistics

You can export the statistics into dbf tables, with which further statistical analyses can be conducted. This function is similar to ArcGIS' *zonal statistics*, but provides additional information.

To export the statistics, click Export Statistics. You will be prompted to provide a name for the output files. This function outputs two dbf tables. Suppose the name you provide is "EssexSoilStats", one dbf table will be named as "EssexSoilStats_I.dbf", and the other will be named as "EssexSoilStats_T.dbf".

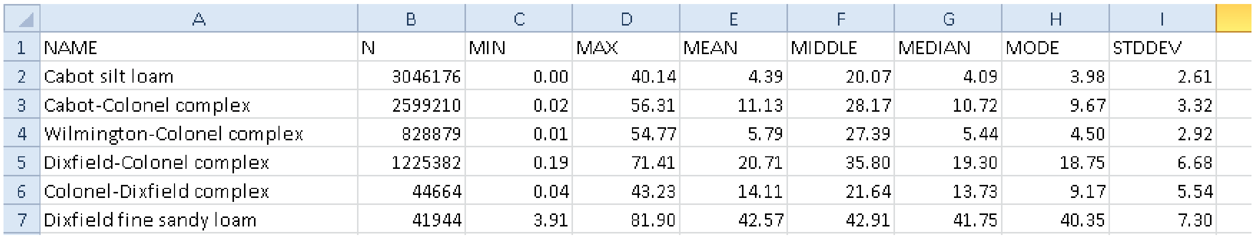
EssexSoilStats_I.dbf contains statistics of each individual polygon:



	A	B	C	D	E	F	G	H	I	J	K
1	NAME	FID	N	MIN	MAX	MEAN	MIDDLE	MEDIAN	MODE	STDDEV	VGC
2	Cabot silt loam_0	0	1343	0.29	11.59	3.15	5.94	2.92	0.85	2.12	3.00
3	Cabot silt loam_1	1	1034	1.19	11.33	6.29	6.26	6.46	1.19	2.03	4.49
4	Cabot silt loam_3	3	4710	0.13	19.94	6.28	10.04	5.81	1.67	3.21	5.90
5	Cabot silt loam_14	14	2201	0.20	27.42	5.43	13.81	3.40	0.86	5.16	3.56
6	Cabot silt loam_15	15	1759	1.79	9.66	4.71	5.72	4.53	3.17	1.23	4.40
7	Cabot silt loam_18	18	1545	0.16	9.81	4.90	4.98	4.62	4.28	1.96	4.15
8	Cabot silt loam_27	27	1374	0.93	34.03	8.59	17.48	4.76	3.00	7.37	3.49

- *Name*: Polygon name in the feature list.
- *FID*: Feature ID of the polygon.
- *N*: Number of cells in the polygon.
- *MIN*: The minimum cell value in the polygon.
- *MAX*: The maximum cell value in the polygon.
- *MEAN*: Mean of the values of all the cells in the polygon.
- *MIDDLE*: The middle value between the min and max values.
- *MEDIAN*: Median of the values of all the cells in the polygon.
- *MODE*: The cell value with the highest frequency in the polygon.
- *STDDEV*: Standard deviation of the values of all the cells in the polygon.
- *VGC*: The cell value at the geometric centroid of the polygon.

EssexSoilsStats_T.dbf contains statistics of *soil type*. The statistics in this table are calculated based on all the cells from all the polygons that belong to the same *soil type*:



	A	B	C	D	E	F	G	H	I
1	NAME	N	MIN	MAX	MEAN	MIDDLE	MEDIAN	MODE	STDDEV
2	Cabot silt loam	3046176	0.00	40.14	4.39	20.07	4.09	3.98	2.61
3	Cabot-Colonel complex	2599210	0.02	56.31	11.13	28.17	10.72	9.67	3.32
4	Wilmington-Colonel complex	828879	0.01	54.77	5.79	27.39	5.44	4.50	2.92
5	Dixfield-Colonel complex	1225382	0.19	71.41	20.71	35.80	19.30	18.75	6.68
6	Colonel-Dixfield complex	44664	0.04	43.23	14.11	21.64	13.73	9.17	5.54
7	Dixfield fine sandy loam	41944	3.91	81.90	42.57	42.91	41.75	40.35	7.30

6.10. Edit the Curve and Output Rule

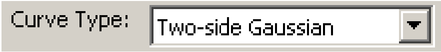
6.10.1. Editing a curve

Just like in the Inference Engine, you can edit the Gaussian curve to make it better represent your understanding of the soil-environment relationship. In other words, you can revise the knowledge discovered from the existing soil map to achieve the goal of *map updating*.

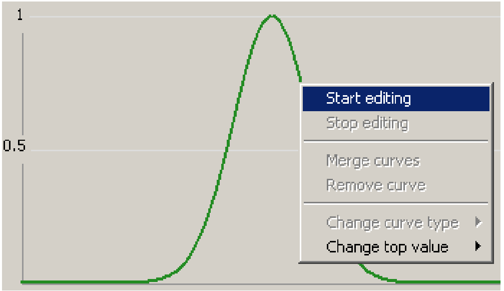
Generally, under two scenarios you might want to revise the discovered knowledge:

- You have more recent knowledge of the soil-environment relationship; and
- You have new and better environmental data that the old knowledge does not account for or cannot fully accommodate.

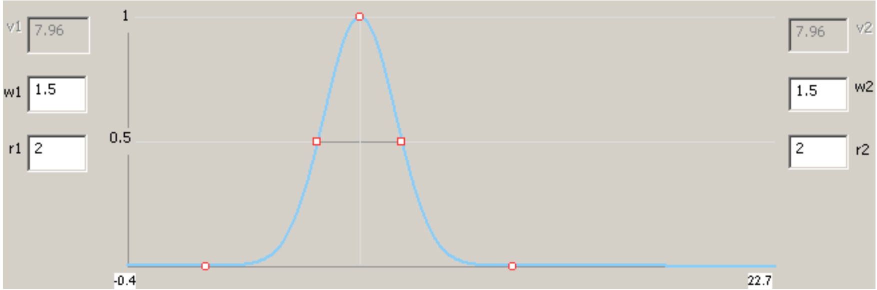
The editing operations are largely the same as those in the Inference Engine. The general procedure is as follows:

c. Make sure you have  Curve Type: Two-side Gaussian. A frequency curves is NOT editable as it is not defined by a simple mathematical function and thus cannot be directly translated into a *rule*.

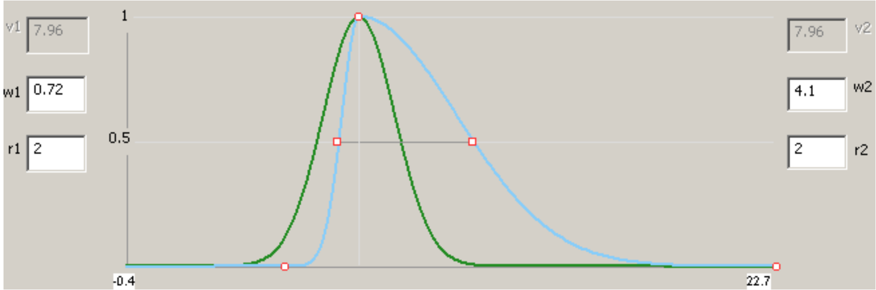
d. Right-click on the curve you want to edit and from the popped-up menu select "Start editing":



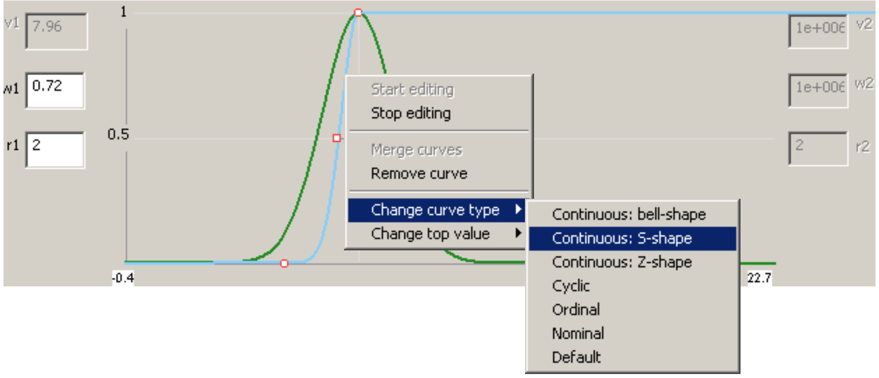
e. Click on the curve. If you choose to edit an original curve (green curve), a highlighted blue curve will appear with the editing handles on it. The blue curve is a copy of the original curve for you to edit. This is for preserving the original curve (the green curve).


v1 7.96
w1 1.5
r1 2
v2 7.96
w2 1.5
r2 2

f. You can change the shape of the curve by adjusting values of the two sets of the parameters (*v*, *w* and *r*). You can also change the type of curve:



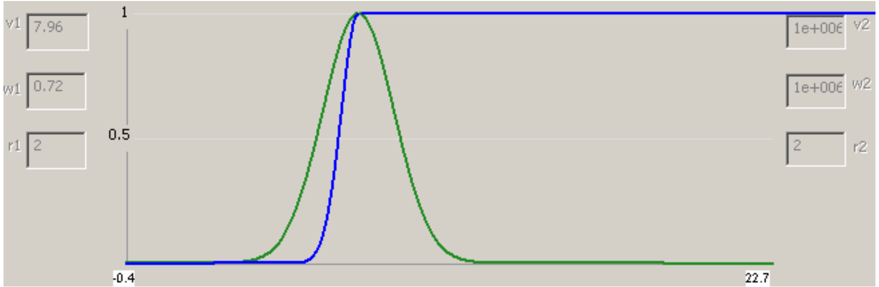
v1 7.96
w1 0.72
r1 2
v2 7.96
w2 4.1
r2 2



v1 7.96
w1 0.72
r1 2
v2 1e+006
w2 1e+006
r2 2

g. You can removed the curve being edited by right clicking on it and choose "Remove curve". Note this only removes the copy that is being edited (the blue one). The original curve (the green one) for the polygon will not be changed.

h. When editing is completed, right click on the curve and select "Stop editing". The edited new curve will stay and be attached to the original curve it is generated from, i.e., it will always be displayed together with the original curve:



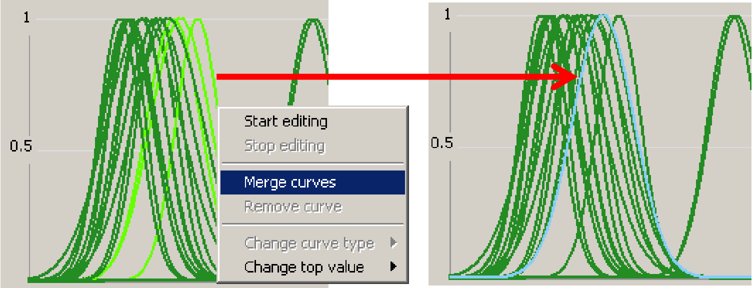
v1 7.96
w1 0.72
r1 2
v2 1e+006
w2 1e+006
r2 2

6.10.2. Merge multiple curves

You can merge multiple original (green) curves into a single curve. The new curve is created using the averages of the parameter values of those participating curves. An edited curve (a blue curve) CANNOT participate in a merging operation.

The procedure of merging curves is as follows:

a. Use Ctrl Key + left mouse button to select the curves you want to merge.
b. Right-click and in the popped-up menu select "Merge curves".



In this example, the three selected curves in the left graphic are merged into the blue curve in the right graphic.

6.10.3. Output curves to *rulebase*

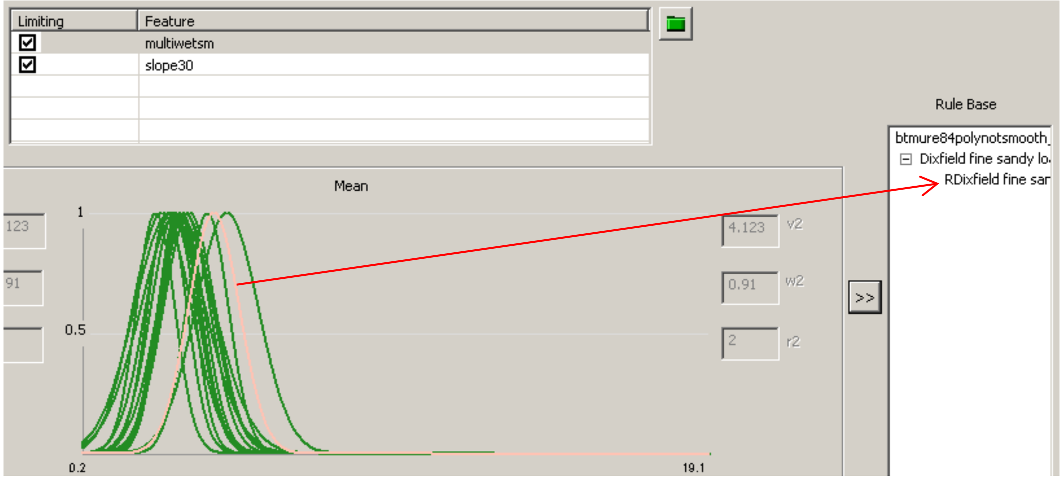
KD can translate Gaussian curves into rules and save them into a *rulebase*. This is the way to save the updated knowledge. You can then use this *rulebase* and the Inference Engine to generate the updated map.

To output a curve to a *rule*, make sure it is not in the editing mode, and then select it by clicking on it and click >>. A new rule will be generated and appears in the *Rulebase* pane.

You can select an original curve (green) or an edited curve (blue) to output. If you choose to output an original curve, you will eventually output a copy of it. In other words, a copy of the original curve will be generated and is associated with the output rule. This is for you to be able to edit it if you need to. After a curve is output, it will turn to red, indicating that its status is "has been output".

You can select a curve to output in either the individual display mode or the group display mode.

The output *rulebase* has the *rulebase->soil type->instance* hierarchical structure. When you output a curve to generate the very first rule, a new *rulebase* will be created, as well as the proper *soil type* and the first *instance* of that *soil type*. The rule will go into that first *instance*. Following that, each time a curve is output, the program checks the *rulebase*: if the *soil type* of this curve has not been created, the *soil type* and the first *instance* of the *soil type* will be created, and the *rule* will go into that first *instance*; if the *soil type* already exists, a new *instance* under that *soil type* will be created to host the new *rule*. The names used in the *rulebase* are taken from their counterparts in the input vector map, except that the default *instance* name starts with an *R* (standing for *rule*) to distinguish it from the original polygon name. You can modify these names in the same way you modify them in the Inference Engine. This is illustrated by the below graphic:



Limiting	Feature
☑	multiwetsm
☑	slope30

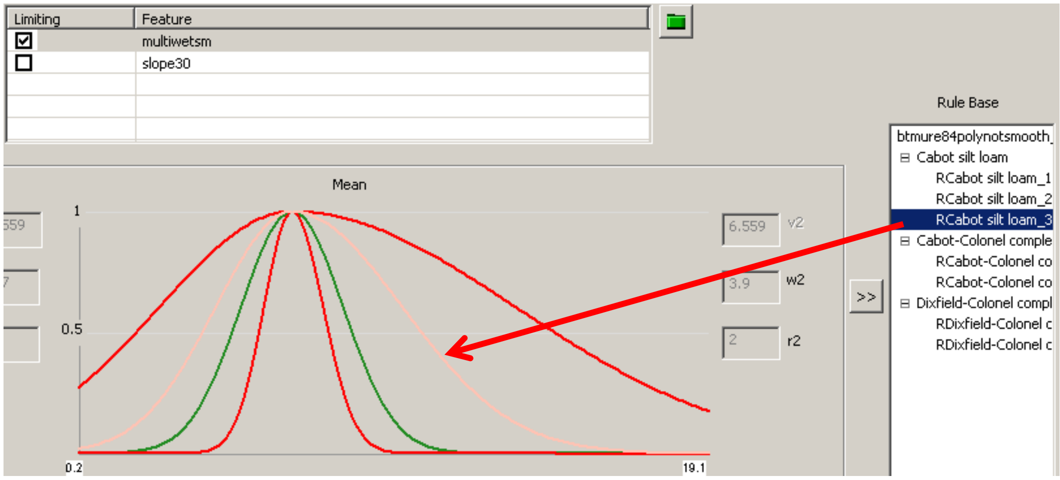
123
91
4.123 v2
0.91 w2
2 r2

Rule Base
btmure84polynotsmooth_
⊟ Dixfield fine sandy lo
RDixfield fine sar

After a curve is output, you can still edit it. The corresponding *rule* in the *rulebase* will be automatically updated with your edits when you "Stop editing".

You cannot output a red curve twice, i.e., a red curve and a *rule* has a one-to-one correspondence. However, based on one single original curve, you can create multiple edited curves and output some or all of them as *rules*.

Clicking on a *rule* (actually an *instance*) brings its corresponding curve to the Curve Editor and makes it selected. In the example illustrated below, clicking on an *instance* selects its corresponding curve, which is one of several curves created based on the same original curve:



Limiting	Feature
☑	multiwetsm
□	slope30

559
7
6.559 v2
3.9 w2
2 r2

Rule Base
btmure84polynotsmooth_
⊟ Cabot silt loam
RCabot silt loam_1
RCabot silt loam_2
RCabot silt loam_3
⊟ Cabot-Colonel comple
RCabot-Colonel co
RCabot-Colonel co
⊟ Dixfield-Colonel compl
RDixfield-Colonel c
RDixfield-Colonel c

When you output a curve to a *rule*, you are actually creating the entire *instance*, which contains *rules* of all the environmental factors in the current *environmental database*. For example, if the current environmental database contains two factors, wetness index and slope gradient, when you output a curve of wetness index, you are outputting its corresponding curve of slope gradient simultaneously into the same *instance*. So, always remember to check *the other* curves in an *instance* to make sure that the formed *instance* meets what you have in mind. KD provides the convenience that you can easily locate *the other* curves and edit them at any time.

You can remove an *instance* in the *rulebase* by right-clicking on it and select "Remove". This does not only remove the *instance* and the *rule(s)* in it, but also its corresponding curve(s). On the other hand, removing a curve by right-clicking on the curve and selecting "Remove curve" also removes its corresponding *instance* in the *rulebase*. When there is no *instance* left under a *soil type*, the *soil type* will be automatically removed. When there is no *soil type* left under the *rulebase*, the entire *rulebase* will be removed.

You can use 💾 to save the *rulebase*. The saved *rulebase* can be directly used by the Inference Engine to create maps.

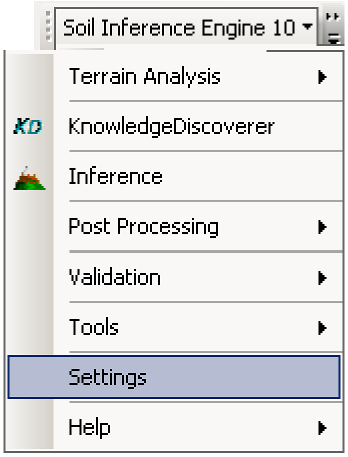
6.10.4. A summary of the meanings of curve colors

In the Curve Editor of the Knowledge discoverer, different colors indicate different status of a curve. This is summarized in the below table:

Color	***Meaning***
Dark green 🟩	Original curve, not selected
Light green 🟩	Original curve, selected.
Dark Blue 🟦	Edited curve, unselected.
Light blue 🟦	Edited curve, selected.
Dark Red 🟥	Output curve, unselected.
Light red 🟥	Output curve, selected.

Chapter 7. Settings

This submenu contains interfaces for adjusting environmental settings for running ArcSIE. Currently, there is only one setting to adjust under this submenu: *Raster I/O Buffer Size*.

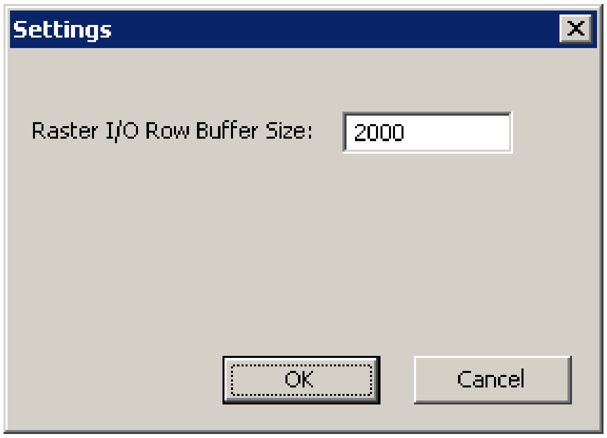


7.1. Change the Raster I/O Row Buffer Size

When a raster layer is very large, or when there are many raster layers to process, the computer's memory may not be able to accommodate them. One way to resolve this problem is to read and process the raster chunk by chunk in a sequent way, and every time only read in and process a number of rows of cells.

Clicking on *Settings* opens the dialog box for adjusting the number of rows to read and process each time, i.e., the size of the chunk.

The example below shows that the user is going to specify the number of rows to be 2000. Under this setting, if the number of columns of the raster is 3000, then each time ArcSIE will read and process 2000 × 3000 cells.



Raster I/O Row Buffer Size: 2000

Whenever possible, you want the computer to take in the entire raster rather than read and process it chunk by chunk. Handling the raster as a whole is much faster and less error-prone in most operations. In other words, whenever possible, you want to set the number of rows on this dialog box to be larger than the number of rows of your raster layer. On most computers, most ArcSIE tools should be able to take in the entire raster if it is not larger than 10000 × 10000.

If your raster layers are large, you can start with specifying a large number of rows. Reduce the number of rows in this dialog box only when you get an "Out of memory" error message.

Appendix A Glossary

case: A representation of the soil scientist's knowledge of the relationship between a soil type and its environment at a specific location. Conceptually, a case is a knowledge composition made up of the information from three spaces: the geographic space, the parametrical space (defined by *environmental features*), and the solution space (taxonomic space). In geographic space, a case corresponds to a location on the ground; In parametrical space, it corresponds to a combination of certain environmental feature values; In solution space, it corresponds to a specific soil type (or, in terms of fuzzy logic, the similarity to this soil type). When creating cases, the soil scientist does not need to specify values in parametrical space, but only needs to pinpoint locations in geographic space. The correspondence between the values in parametrical space and solution space will automatically be built by the inference engine through the correspondence between geographic space and solution space. Technically, in ArcSIE, a case is eventually an instance plus a spatial setting. The spatial setting includes the location information (coordinates) and values of some parameters for performing spatial inference.

casebase: A type of *knowledgebase* that utilizes a collection of *cases* to represent knowledge of local soils. It is created for a certain mapping area and supports the case-based reasoning for mapping the soils in the area. In a *casebase*, *cases* are organized into one or more *case lists*. Each *case list* represents a soil type to be mapped.

instance: A representation of the soil scientist's knowledge of the relationship between a soil type and its environment characterized by topography, geology, climate, vegetation, and other environmental features. It is not explicitly location-specific and is usually applicable to a large portion of the mapping area or even the entire mapping area, thus is usually considered to be the "global" knowledge. Technically, an *instance* contains a set of rules regarding a set of *environmental features*.

Knowledgebase: A structured and formalized collection of the knowledge of the soil-environmental relationships in an area. In ArcSIE, a knowledgebase may be either a *rulebase* or a *casebase*.

line case: *Case* in the form of a line. Suitable for soil types or geomorphic features whose distributions are linear.

point case: *Case* in the form of a point, also called *tacit point*. Each *case* corresponds to a point location on the ground.

polygon case: *Case* in the form of a polygon.

rule: A *rule* is a fuzzy membership function defining the relationship between the values of an *environmental feature* and the optimality values for a soil type.

soil type: A *soil type* in ArcSIE corresponds to a mappable individual; the target of a specific inference model. Technically, it is a collection of *instance*(*s*) or *case(s)*.

rulebase: A type of knowledgebase that utilizes *instances*, which is a group of *rules*, to represent knowledge of local soils. It is created for a specific mapping area and supports the rule-based reasoning for mapping the soils in the area. In a *rulebase, instances* are organized into one or more *instance lists*. Each *instance list* represents a soil type to be mapped.

tacit point: See ***point case***.

Appendix B File Suffixes

.3dm: 3dMapper's data format for an orthophoto-DEM pair. Used by 3dMapper for visualization and terrain attribute calculations.

ArcInfo Grid: ESRI's raster data format. Used by ArcSIE as the native raster data format. All the environmental data layers input to ArcSIE must be in this data format. All the maps produced by ArcSIE are in this format.

.**asc**: The ASCII format for Arc/Info Grid. Used to exchange raster data between 3dMapper/ArcSIE and ArcGIS.

_chk.: *Check file* created by ArcSIE. See 1.3.5.3.1.

_chk.lkt: *Look-up* table file created by ArcSIE. See 1.3.5.3.1. This is a text file.

.dbf: File format of dBASE table. ArcSIE uses dbf tables for storing *rulebases* and *raster casebases*.

.shp: Shapefile of ArcView. ArcSIE uses Shapefile as the data format for vector (*point*, *line*, and *polygon*) *casebase*. ArcSIE creates a new *casebase* using a Shapefile created in a GIS (e.g., 3dMapper, ArcMap, or ArcView). The created *casebase* is still saved as a Shapefile of the same type as that of its source data, but the new Shapefile will contain new fields required by the *casebase* in its attribute table.