Fuzzification – An Arcview Script To Be Used With Fuzzy Logic And Neural Network Applications

Introduction

The fuzzy logic method of spatial analysis requires that the crisp data be scaled into fuzzy membership values, ranging from zero to one. This is a process called fuzzification (Tsoukalas and Uhrig, 1997). Fuzzification is also useful for pre-processing the data for analysis in neural networks. An Avenue script for Arcview called fuzzy.ave that augments the tools in ArcSDM (Kemp and others, 2001) is documented here. Fuzzy.ave implements several algorithms that are in common use in fuzzy-logic applications. The advantage of using an algorithm to transform the crisp measurements into fuzzy membership values is that it makes the transformation repeatable and easy to report. For reporting, it is only necessary to identify the algorithm used and the parameters selected for that algorithm. Additional insights into fuzzy logic and fuzzification can be found in Tsoukalas and Uhrig (1997), Burrough and McDonnell (1998), and Masters (1993).

Fuzzification Algorithms

The algorithms implemented in Fuzzy.ave are the following: Small (Tsoukalas and Uhrig, 1997), Near (Tsoukalas and Uhrig, 1997), Gaussian (Masters, 1993), and Large (Tsoukalas and Uhrig, 1997). These fuzzification algorithms can also be modified, such as very small, with an additional set of algorithms referred to as hedges (Tsoukalas and Uhrig, 1997, Zadeh, 1993).

Fuzzy.ave adds a new attribute containing the attribute to be fuzzified into the active integer grid. The menu for the selection of the parameters is shown in Figure 1. The user selects the algorithm and hedge by typing one of the names in square brackets and the spread and mid values. Spread is a parameter of the fuzzification algorithm that determines how rapidly the fuzzy membership values decrease from one to zero. Mid is the parameter that defines the crisp value. That value will have a membership of 0.5 for small or large fuzzification algorithms or the middle value having the maximum fuzzy membership value for the near and Gaussian fuzzification algorithms. If it is desired to scale the fuzzy membership values from a maximum of less than one, then the fuzzy membership values from these algorithms can always be rescaled, for example multiplication by 0.75 would reduce the maximum fuzzy membership value to 0.75.

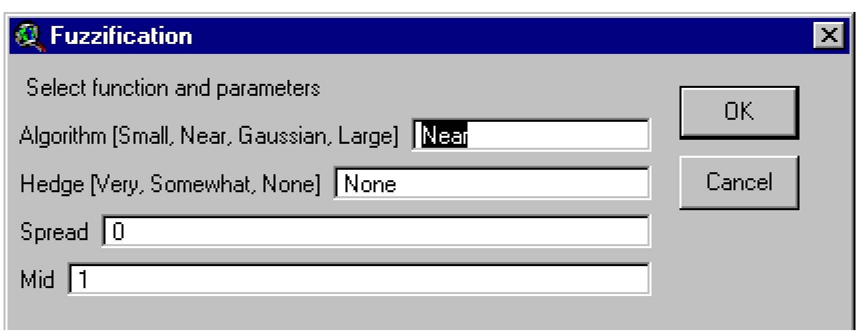

Figure 1: Fuzzification menu for input of fuzzification parameters.

Hedges

The two hedges implemented are *very* and *somewhat* (Tsoukalas and Uhrig, 1997). *Very* is also known as concentration. *Very* is defined as the fuzzy membership function squared. *Somewhat* is also known as dilation or the linguistic term "More or Less". *Somewhat* is the square root of the membership function. The *very* and *somewhat* hedges decrease and increase, respectively, the fuzzy membership functions.

Small and Large

The fuzzification algorithms small and large are used to indicate that small or large values of the crisp set are members of the fuzzy set. The spread and mid parameters are subjectively defined to reflect the expert opinion. Examples of the small and large functions and hedges are shown in Figure 2. The small fuzzification algorithm is defined as

$$\mu(x) = \frac{1}{1 + \left(\frac{x}{f_2}\right)^{f_1}} \qquad \text{(Equation 1: Fuzzy Membership Small)}$$

Where f_1 is the spread of the transition from a membership value of 1 to 0 and f_2 is the midpoint where the membership value is 0.5 (Tsoukalas and Uhrig, 1997).

The large fuzzification algorithm is defined as

$$\mu(x) = \frac{1}{1 + \left(\frac{x}{f_2}\right)^{-f_1}} \qquad \text{(Equation 2: Fuzzy Membership Large)}$$

Where f_1 is the spread of the transition from a membership value of 1 to 0 and f_2 is the midpoint where the membership value is 0.5 (Tsoukalas and Uhrig, 1997).

Note this function works improperly for negative crisp values. To apply these functions to negative numbers, the crisp values need to be transformed to positive numbers before fuzzification.

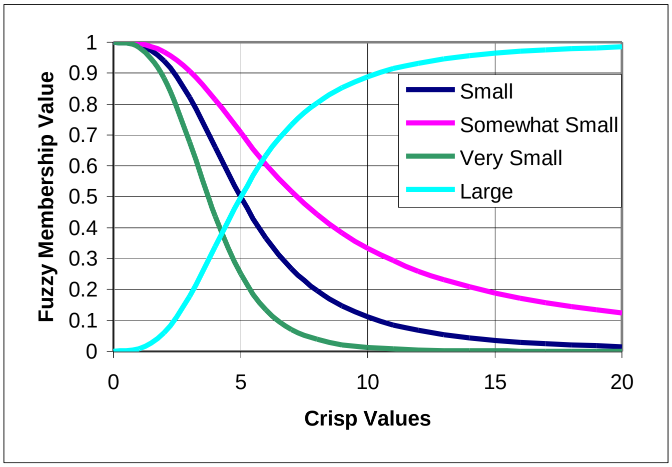

Figure 2: Examples of small and large fuzzification using a mid value of 5 and a spread of 3.

Near

The fuzzification function near is used when some intermediate crisp value is the member of the fuzzy set. The spread and mid parameters are subjectively defined to reflect the expert opinion. An example of the near algorithm is given in Figure 3. The near function is also known as a sinusoidal membership function (Burrough and McDonnell, 1998). The near fuzzification algorithm is defined as

$$\mu(x) = \frac{1}{1 + f_1(x - f_2)^2} \qquad \text{(Equation 3 : Fuzzy Membership Near)}$$

Where f_1 is the spread of the transition from a membership value of 1 to 0 and f_2 is the midpoint where the membership value is 0.5 (Tsoukalas and Uhrig, 1997).

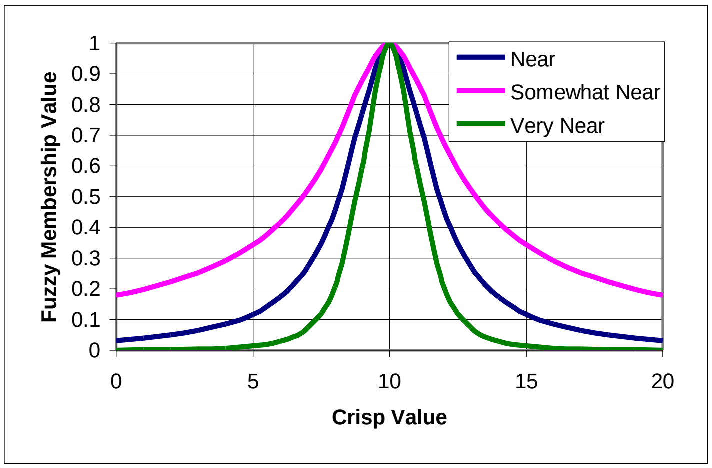

Figure 3: Example of near fuzzification using a mid value of 10 and a spread of 0.3.

Gaussian

The fuzzification function Gaussian is similar to the near function but has a more narrow spread. The near fuzzification algorithm is defined as

$$\mu(x) = e^{-f_1(x - f_2)^2} \qquad \text{(Equation 4: Fuzzy Membership Gaussian)}$$

Where f_1 is the spread of the transition from a membership value of 1 to 0 and f_2 is the midpoint where the membership value is 0.5 (Tsoukalas and Uhrig, 1997).

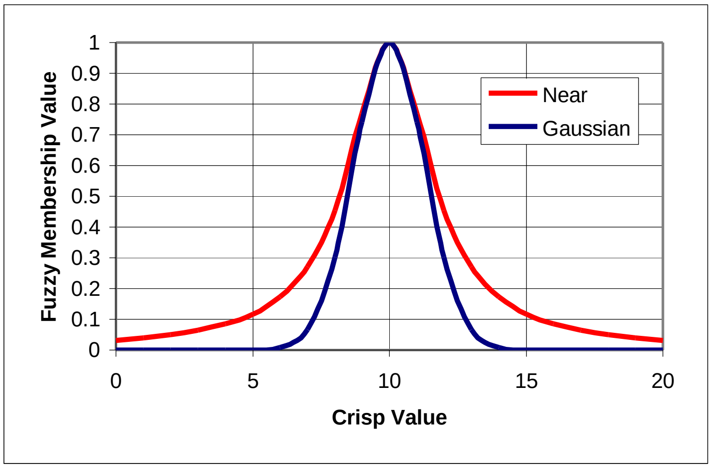

Figure 4: Example of the Gaussian fuzzification function compared to the near function using a spread of 0.3 and a mid value of 10.

Combinations

Combination fuzzification functions can be made by applying multiple fuzzification functions to an integer-grid table and then editing the table to piecewise combine the different functions. An example of such a process using the two near functions is shown in Figure 5

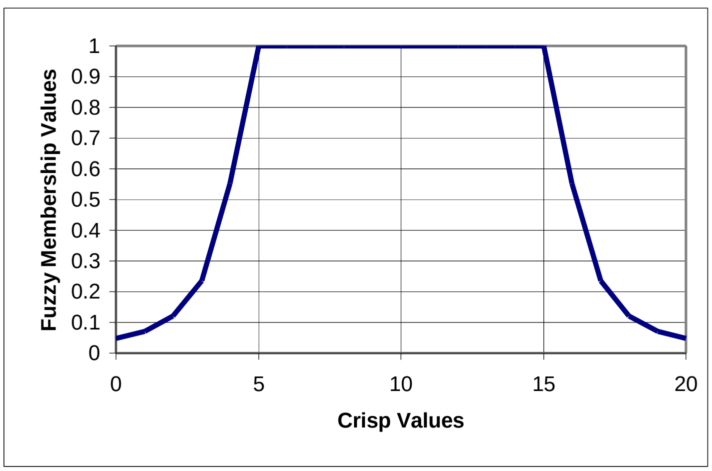

Figure 5: Example of a combination of two near functions with spread of 0.9, a mid value of 5 for less than 5, a mid value of 15 for greater than 15, and a membership value of 1 between 5 and 15.

Spread

The selection of the appropriate spread value is a subjective process that is dependent on the range of the crisp values. A useful way to experiment with different spread values is to use a spreadsheet program with graphs. Then a picture of the effects of different spread functions can be quickly developed. Note, as shown in Figure 6 and Figure 7, as the spread gets smaller the fuzzy memberships approach zero more slowly.

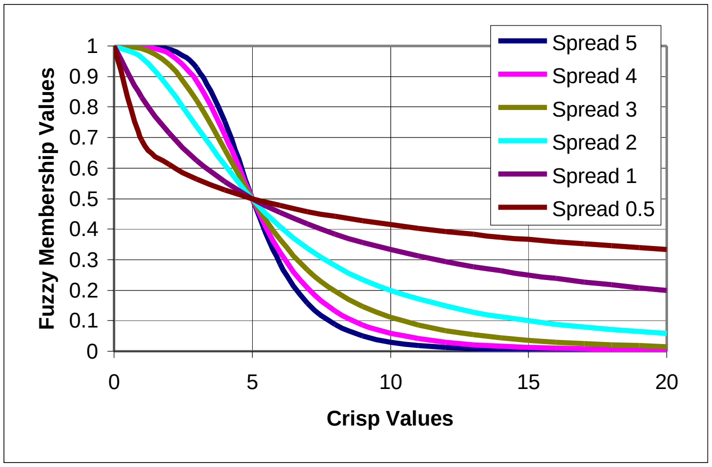

Figure 6: Examples of a range of spreads for a small function with a constant mid value of 5.

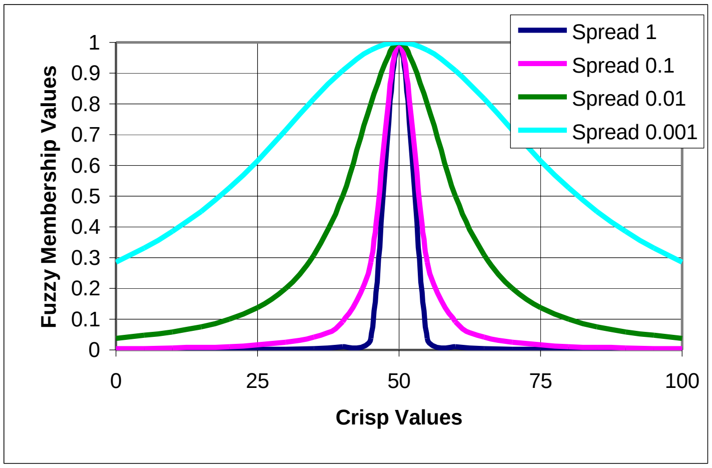

Figure 7: Examples of a range of spreads for a near function with a constant mid value of 50.

References

Burrough, P.A., and McDonnell, R.A., 1998, Principles of geographical information systems: New York, Oxford University Press, 333 p.

Kemp, L., Bonham-Carter, G.F., Raines, G.L., and Looney, C.G., 2001, ArcSDM and DataXplore – spatial data modeler for Arcview and Spatial Analyst: http://ntserv.gis.nrcan.gc.ca/sdm/.

Masters, Timothy, 1993, Practical neural network recipes in C++: New York, Academic Press, p. 285.

Tsoukalas, L.H., and Uhrig, R.E., 1997, Fuzzy and neural approaches in engineering: New York, John Wiley and Sons, Inc., 587 p.

Zadeh, L.A. A computational approach to fuzzy quantifiers in natural languages: Computers and Mathematics, v.9, p. 149-184.