
Subject: Curve Fitting Question

Posted by [cgguido](#) on Tue, 13 Nov 2012 00:23:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

How about Otsu's method? Meant for images but works for any set of numbers.

http://www.cis.rit.edu/~cnspci/media/software/otsu_threshold.pro

If your numbers aren't integers, I'd multiply by 1000, round, run otsu, divide by 1000. That's assuming 3 sig figs are enough...

Subject: Otsu Threshold for Bimodal Data

Posted by [David Fanning](#) on Wed, 21 Nov 2012 17:51:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

Gianguido Cianci writes in response to my question about finding an automatic way to locate the threshold that will separate a bimodal data set into two populations:

- > How about Otsu's method? Meant for images but works for any set of numbers.
- >
- > http://www.cis.rit.edu/~cnspci/media/software/otsu_threshold.pro
- >
- > If your numbers aren't integers, I'd multiply by 1000, round, run otsu, divide by 1000. That's assuming 3 sig figs are enough...

The OTSU_THRESHOLD program Gianguido mentions above makes a number of undocumented assumptions that can cause problems in the real-world use of the program. In my first attempt at implementing this algorithm I inadvertently copied all those dangerous assumptions into my own cgOtsu_Threshold program.

In particular, this Otsu_Threshold program assumes you are working with byte type images and that your binsize is 1. It returns erroneous thresholds if you are working with real data outside the range 0 to 255 or using bin sizes other than 1.

Since I work with both these conditions often in effect, I needed a more robust algorithm. Today I found a great web page that allowed me to rebuild the algorithm from the ground floor up:

<http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>

The new algorithm replicates the results of OTSU_Threshold

for byte data with a binsize of 1, but it also works with real data of whatever data range and with different bin sizes.

You can find the old program with a new algorithm here:

http://www.idlcoyote.com/programs/cgotsu_threshold.pro

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Otsu Threshold for Bimodal Data

Posted by [David Fanning](#) on Sat, 24 Nov 2012 17:20:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

- > The OTSU_THRESHOLD program Gianguido mentions above makes
- > a number of undocumented assumptions that can cause problems
- > in the real-world use of the program. In my first attempt
- > at implementing this algorithm I inadvertently copied all
- > those dangerous assumptions into my own cgOtsu_Threshold
- > program.
- >
- > In particular, this Otsu_Threshold program assumes you
- > are working with byte type images and that your binsize
- > is 1. It returns erroneous thresholds if you are working with
- > real data outside the range 0 to 255 or using bin sizes
- > other than 1.
- >
- > Since I work with both these conditions often in effect,
- > I needed a more robust algorithm. Today I found a great
- > web page that allowed me to rebuild the algorithm from
- > the ground floor up.

The new algorithm turned out to be not quite as robust as I expected it to be. But, good news, this turned out to be due to a faulty assumption on my part and not to a failure of the algorithm. In particular, the program was having problems dealing with integer data.

My fault entirely. I assumed an equation I was using with

the threshold values was converting the threshold values to floats. It was not. Once this problem was solved, the algorithm is again acting like a champ!

http://www.idlcoyote.com/programs/cgotsu_threshold.pro

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thue. ("Perhaps thou speakest truth.")
