Kullback-Leibler (KL) distance between histograms - matlab

```
function [ d ] = hcompare_KL( h1,h2 )
%This routine evaluates the Kullback-Leibler (KL) distance between histograms.
%             Input:      h1, h2 - histograms
%             Output:    d – the distance between the histograms.
%             Method:    KL is defined as:
%             Note, KL is not symmetric, so compute both sides.
%             Take care not to divide by zero or log zero: disregard entries of the sum

temp = sum(h1 .* log(h1 ./ h2));
temp( isinf(temp) ) = 0; % this resloves where h1(i) == 0
d1 = sum(temp);

temp = sum(h2 .* log(h2 ./ h1)); % other direction of compare since it's not symetric
temp( isinf(temp) ) = 0;
d2 = sum(temp);

d = d1 + d2;

end
```

my problem is that whenever h1(i) or h2(i) == 0 i'm getting inf which is as expected. however in the KL distance i'm suppose to return 0 whenever they h1 or h2 ==0 how can i do that without using a loop ?

matlab image-processing

edited Nov 14 '12 at 6:43

asked Nov 13 '12 at 22:46

Androidy
973 1 6 34

1 It is really difficult to help you if you don't ask better questions. I can't find your mistake either if I don't know what the program should do in the first place. Please suggest a sample input, tell us what output you expect, and what goes wrong. Does the function throw an error? Does the function not return what you want? I have downvoted your question, though I'm happy to revise my vote if the question improves. – Jonas Nov 13 '12 at 23:09

hi @Jonas thanks for the daily answers as u can see i'm learning as we go. let me specify my question later, sorry and thank you – Androidy Nov 13 '12 at 23:13

@jonas i have edited my question can u please look at it, let's assume we have h1=[0:9] and h2=[1:10] as input i will get an error when i have 0 as input.. log(0) – Androidy Nov 14 '12 at 14:42

Now it's a clearer question. Hope my answer helps. – Jonas Nov 14 '12 at 15:32

1 Answer

To avoid having issues when any of the counts is 0, I suggest you create an index that marks the "good" data points:

```
%# you may want to do some input testing, such as whether h1 and h2 are
%# of the same size

%# preassign the output
d = zeros(size(h1));

%# create an index of the "good" data points
```

```
goodIdx = h1>0 & h2>0; %# bin counts <0 are not good, either

d1 = sum(h1(goodIdx) .* log(h1(goodIdx) . /h2(goodIdx)));
d2 = sum(h2(goodIdx) .* log(h2(goodIdx) . /h1(goodIdx)));

%# overwrite d only where we have actual data
%# the rest remains zero
d(goodIdx) = d1 + d2;
```

answered Nov 14 '12 at 15:32

Jonas
51k 3 64 94

yes exactly what i did i used h1(find(h1==0))=1, thanks – Androidy Nov 14 '12 at 16:05

Not the answer you're looking for? Browse other questions tagged matlab image-processing or ask your own question.