
Subject: How to speed up kernel density smoothing for many data points

Posted by [jacobsvensmark](#) on Thu, 10 Oct 2013 12:22:46 GMT

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

Okay so I have a long array of N 2D points: `x = fltarr(N,2)` (as exemplified below). For each point `x(i,*)` I need to input the distance to all other points into my K function: `K(xi-x)`, sum together K for each point, square root and put into my final function `f1(i)`.

I guess I need help both making the input `xi-x` faster, as well as making K run faster. I tried with 'REPLICATE_INPLACE' which helped some, but I am out of ideas....

PRO test_kernel

```
N = 1000000
```

```
r = 2*randomn(seed,N)
```

```
v = 2*randomu(seed,N)
```

```
x = [[r],[v]]
```

```
; Just a smoothing parameter, unimportant...
```

```
hopt = 6.24/(N^(1./6.))*sqrt((stddev(x(*,0))^2+stddev(x(*,1))^2)/2. )
```

```
; Slow loop, where I need help
```

```
f1 = fltarr(N)
```

```
xi = fltarr(N,2)
```

```
for i=0L,N-1 do begin
```

```
    REPLICATE_INPLACE, xi, x(i,0), 1, [1,0]
```

```
    REPLICATE_INPLACE, xi, x(i,1), 1, [0,1]
```

```
    f1(i) = 1./float(N) * total(1./(hopt^2)*K(xi-x,hopt))
```

```
endfor
```

```
END
```

```
FUNCTION K,tvec,h
```

```
  t = vec_norm(tvec)/h
```

```
  aa = where(t ge 1.,n0)
```

```
  bb = where(t lt 1.,nt)
```

```
  RES = fltarr(n_elements(t))
```

```
  if n0 ne 0 then RES(aa) = 0.
```

```
  if nt ne 0 then RES(bb) = 4./!Pi*(1.-t(bb)^2)^3
```

```
  RETURN,res
```

```
END
```

Thanks, Jacob
