

Stat 992: Lecture 06

Iterated kernel smoothing.

Moo K. Chung mchung@stat.wisc.edu

February 2, 2004

1. *Iterated kernel smoothing.* From the property of a diffusion smoother, we have

$$K_{m\sigma} = \underbrace{K_\sigma * \cdots * K_\sigma}_{m^2 \text{ times}}. \quad (1)$$

Based on this identity, we implement kernel smoothing numerically. Let $\sigma = 0.4$ and apply iterated kernel smoothing. With small window size 5×5 , Gaussian kernel smoothing with $\sigma = 28$ will chop off the tail regions too much and this is one of the motivation for doing iterated kernel smoothing.

```
w=K(0.4,dx,dy)/sum(sum(K(0.4,dx,dy)))
smooth_Y=Y;
for i=1:250
    smooth_Y=conv2(smooth_Y, w, 'same');
    ISE(i)=sum(sum((mu-smooth_Y).^2));
end;
```

Problem 12. Come up with a similar identity like equation (1) for anisotropic kernel smoothing with bandwidth matrix H .

2. *Integrated squared error.* When $\sigma = 28$ with 490-iterations, Figure 1 shows obvious over smoothing while $\sigma = 0.8$ with 4-iterations are under smoothing. So there must be an optimal bandwidth that estimate the signal best. Most automatic bandwidth selection methods are based on i.i.d. error assumption so the results do not usually work in imaging where image intensities have highly correlated geometric structure. A reasonable way to chose optimal bandwidth that minimizes the fit of the estimation in L_2 norm, i.e.

$$\min_{\sigma} \int [\mu(x) - K_{\sigma} * Y(x)]^2 dx$$

which is called the integrated squared error (ISE). From our simulation the optimal bandwidth is when $\sigma = \sqrt{18} \cdot 0.4 = 1.70$. Unfortunately this gives under smoothing. See J.S. Simonoff's *Smoothing Methods in Statistics* (1996) for optimal kernel smoothers. With i.i.d. error assumptions, you may get rid of outliers easily and run statistical analysis again but in

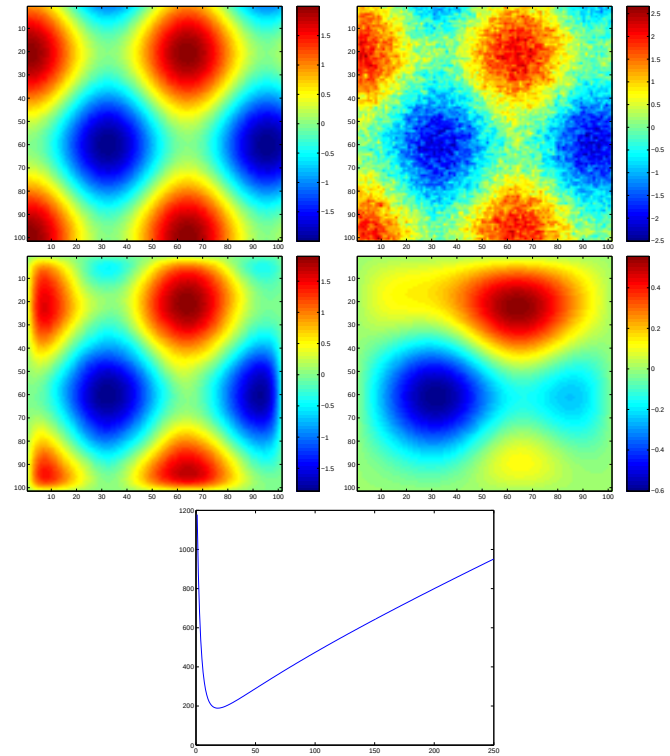


Figure 1: Original unknown signal μ and kernel smoothing estimations corresponding to $\sigma = 0.8(m = 2)$, $\sigma = 4(m = 10)$, $\sigma = 28(m = 70)$. $\sigma = 28$ is obviously the case of over smoothing while $\sigma = 0.8$ is under smoothing. The bottom left image is ISE plotted over the number of iterations.

imaging data we can not just delete outliers since it will destroy the geometry of the image itself.

Problem 13.¹ Check if cross-validation (CV) method can produce a reasonable optimal bandwidth. For example, out of 101×101 image data Y , you may decompose images into training image $Y_L = Y(1:2:101, 1:2:101)$ and validation image $Y_R = Y(2:2:101, 2:2:101)$ and try to minimize the integrated squared error in the validation image using the model fit of the training image. You may try other partitioning schemes. CV is a standard technique in computer vision and statistics. See Ruppert *et al.* (2003) *Semiparametric Regression*.

¹**Project 4.** Investigation of CV in 2D image. Most CV results in the statistical literature that assume i.i.d. error will not work here.

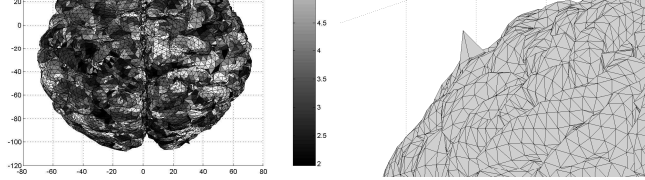


Figure 2: The left image is noisy thickness measure on the surface of brain. The right image shows an image processing artifact that will cause serious problem in statistical analysis later. For example in the estimation of curvature

```
[tri,coord,nbr,normal]=MNigetmesh
                        ('gray.obj');
load thickness.vv
trisurf(tri,coord(1,:),coord(2,:),
        coord(3,:));
```

`tri` gives the list of 3 indices that form a triangle. `coord` gives the coordinates of vertices `nbr` gives the list of adjacent neighbors. `normal` gives unit normal vectors at each vertex. `trisurf` will display the surface mesh in MATLAB. Some graph theoretic concept is useful in describing and implementing kernel smoothing on the mesh. The degree of a vertex is the number of edges that connects to the vertex. Read `MNigetmesh.m` for understanding how one gets it.

```
>> coord(:,1:4)
ans =
    49.9831    44.0817   -54.7131   -40.8825
    30.4634   -51.8008    32.6077   -58.8814
     7.4515    19.1203    17.1390    19.4011
```

```
>> tri(1:5,:)
ans =
         3    10243    10245
    10243     2563    10244
    10245    10243    10244
    10245    10244     2565
    2563    10246    10248
```

```
>> normal(:,1:4)
ans =
    0.4151   -0.1340   -0.9101   -0.2198
   -0.1868    0.7082    0.1541    0.9744
    0.8904   -0.6932    0.3847    0.0482
```

```
>> nbr(1:3,:)
ans =
    15473    15474    16484    28770    28772     0
    30322    30321    30324    31844    38497     0
    10243    10245    11827    19554    20082     0
```

where V, E and F are the number of vertices, edges and faces respectively. In our data $V = 40962$ and $F = 81920$. For a mesh topologically equivalent to a sphere, $\chi(M) = 2$. If we count the number of edges E by counting the number of faces it is $3F$ but we are counting them twice so $3F = 2E$. So simplifying the relation we have

$$2V - F = 4. \quad (2)$$

Equation (2) gives a way to check the topology of mesh. A particular note of interest is that from the *Gauss-Bonnet theorem*, the integral of the Gaussian curvature over M is $2\pi\chi(M)$. So when we integrate the Gaussian curvature κ_G over the brain surface M , we should get

$$\int_M \kappa_G(x) dx = 4\pi.$$

Homework 2. Submit solutions to problems 4-12 by Feb 16 11:00am. You only need to do two thirds of the total problems I pose in the course. For computational problems, please do not just dump me codes. It is your responsibility that you document your codes and proving the correctness of your algorithm.

4. *Topological invariant.* We will only deal with closed surface mesh that is topologically invariant to a sphere. One way to check it is to compute the Euler characteristic $\chi(M)$ of the triangular mesh M which is defined as

$$\chi(M) = V - E + F$$