Stat 992: Lecture 15
Diffusion Smoothing I.

Moo K. Chung mchung@stat.wisc.edu

December 23, 2003

Problem 11. Come up with other method for proving iterated kernel smoothing result. One approach is to compute the convolution directly.
Solution by Tulaya Limpiti. Note that

$$\begin{aligned} K_\sigma * K_\sigma(x) &= \mathcal{F}^{-1}[(\mathcal{F}K_\sigma)^2](x) \\ &= \mathcal{F}^{-1}K(\sqrt{2\pi}\sqrt{2}\sigma x) = K_{\sqrt{2}\sigma}(x). \end{aligned}$$

Problem 12. Come up with a similar identity like equation (1) for anisotropic kernel smoothing with bandwidth matrix H.
Solution by Shubing Wang. Note that for n-dimensional i.i.d. random vector $X_i \sim N(0, HH')$, $X_1 + X_2 \sim N(0, 2HH')$. The probability density of $X_1 + X_2$ is $K_H * K_H$. So we must have $K_H * K_H = K_{\sqrt{2}H}$. The general iterative kernel smoothing for anisotropic case follows similarly.

1. *Diffusion smoother in one dimension.* Diffusion equation was introduced in the previous lectures to prove some properties of kernel smoothers. Mathematically kernel and diffusion smoothers are identical but they differ in numerical implementation and if one use large bandwidth in kernel smoothing, there is the problem of boundary effect and kernel smoothing researchers usually came up with the methods for correcting boundary effects. However, if one uses diffusion smoother, the correction for boundary effect can be incorporated into the boundary value problem (BVP) of partial differential equation. Let us implement one dimensional version first. One dimensional isotropic heat equation is defined as

$$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2}$$

with initial data $f(x,t) = Y(x)$. It can be solved by the finite difference method so that

$$f(x, t_{j+1}) = f(x, t_j) + \delta t \widehat{\frac{\partial^2 f}{\partial x^2}}(x, t_j)$$

where $\widehat{\frac{\partial^2 f}{\partial x^2}}$ is an estimate for the second derivative and the iteration step size $\delta t = t_{j+1} - t_j$ is fixed.

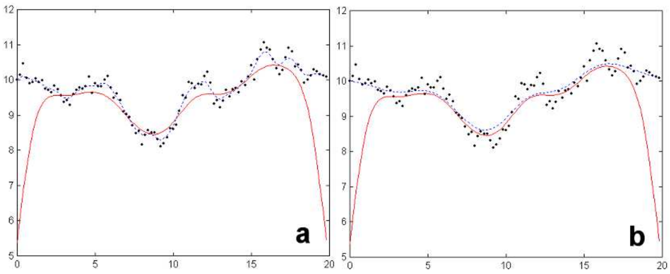

Figure 1: Solid lines are kernel smoothing estimates while the doted lines are diffusion smoothing estimate. **a.** $\sigma = 1$ with 25 iterations. **b.** $\sigma = 1$ with 50 iterations. At the boundary regions, kernel smoothing fails while diffusion smoothing has no such problem due to the boundary value condition.

It is possible to change this value depending on the smoothness of data, i.e. spatially adaptive filter design, but we will not discuss it at this moment.

There are a couple of ways of estimating the second order derivatives. One simple way is to take the finite difference scheme twice so that

$$\widehat{\frac{\partial f}{\partial x}}(x, t_j) = \frac{f(x + \delta x, t_j) - f(x, t_j)}{\delta x}.$$

and

$$\begin{aligned} \widehat{\frac{\partial^2 f}{\partial x^2}}(x, t_j) &= \frac{\frac{f(x+\delta x, t_j) - f(x, t_j)}{\delta x} - \frac{f(x, t_j) - f(x - \delta x, t_j)}{\delta x}}{\delta x} \\ &= \big[f(x - \delta x, t_j) - 2f(x, t_j) + f(x - \delta x, t_j)\big]/\delta x^2 \end{aligned}$$

For the simplicity of argument, we assume data to be observed in discrete grid with equal distance apart but this restriction is not necessary for either kernel or diffusion smoothing. If we index the neighboring spatial positions of x by $x_{i-1} = x - \delta x$, $x_i = x$ and $x_{i+1} = x + \delta x$, we can see that the process of diffusing signal is basically iterated weighted averaging, which is essentially the property of kernel smoothing.

We force the following boundary condition $f(x_1, 0) = Y(x_1)$ and $f(x_m, 0) = Y(x_m)$. A dif-

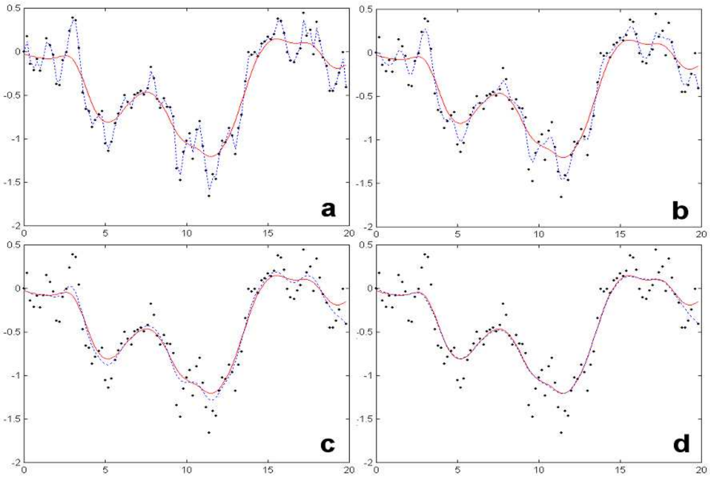

Figure 2: Comparison of kernel and diffusion smoothing. It can be seen that the diffusion smoothing (dotted lines) converges to kernel smoothing as the number of iterations increase (5, 25 and 50 iterations respectively).

ferent boundary condition can be applied depending on the type of problem dealing with.

Afterward the finite difference scheme is iterated many times with very small δt. If δt is too small, the computation is slow while if it is large, we will have the problem of divergence. So the problem is finding the largest δt that grantee the convergence.

2. *Discrete maximum principle.* Since the diffusion smoothing and kernel smoothing are equivalent, The diffused signal $f(x_i, t_{j+1})$ must be bounded by the minimum and the maximum of signal. So

$$
\begin{aligned}
f(x_i, t_{j+1}) &= f(x, t_j) + \delta t \widehat{\frac{\partial^2 f}{\partial x^2}}(x, t_j) \\
&\leq \max\left[f(x_{i-1}, t_j), f(x_i, t_j), f(x_{i+1}, t_j)\right]
\end{aligned}
$$

and also we can bound it by below. Hence

$$
\delta t \leq \max\Big[\Big|\frac{f(x_{i-1}, t_j) - f(x_i, t_j)}{\widehat{\frac{\partial^2 f}{\partial x^2}}}\Big|, \Big|\frac{f(x_{i+1}, t_j) - f(x_i, t_j)}{\widehat{\frac{\partial^2 f}{\partial x^2}}}\Big|\Big].
$$

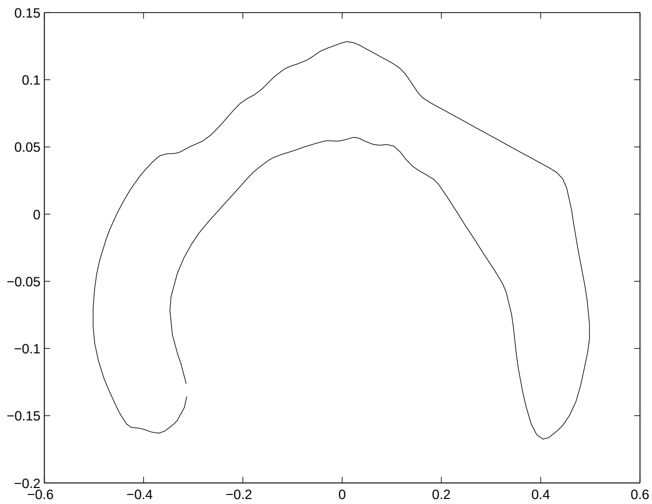

Figure 3: Corpus callosum boundary. See Problem 25.

Problem 25. Download data from
http://www.stat.wisc.edu/
∼mchung/teaching/data/CCcurve.data
Estimate the curvature using only up to 5 adjacent vertices at a time and apply diffusion smoothing. You will get extreamly noise curvature function. Smooth it via diffusion. Experiment with different bandwidth and pick one that smooth best.

```
load CCcurve.data;
x=CCcurve(:,1); y=CCcurve(:,2);
plot(x,y)
```