

Stat 992: Lecture 28

Finite element methods.

Moo K. Chung mchung@stat.wisc.edu

December 5, 2003

This lecture is based on **Problem 31.** and part of solutions given by Tulaya Limpiti. Perform diffusion smoothing on the cortical thickness on unit sphere S^2 using Delaunay triangulation.

1. *Review on Spherical Laplacian.* Let X be the parameterization of S^2 via

$$x = \sin \theta \cos \psi, \quad y = \sin \theta \sin \psi, \quad z = \cos \theta.$$

Note that $X : D \rightarrow S^2$ where $(\theta, \psi) \in D = [0, 2\pi] \otimes [0, \pi]$.

```
[a, b, r] = cart2sph(Scoord(1,:), ...
Scoord(2,:), Scoord(3,:));
theta=a+pi; psi=pi/2-b;
figure; trimesh(Ptri, theta, psi, ...
thickness); colorbar
```

Make sure you are diffusing signal on S^2 not on parameter space D . We derived the spherical Laplacian to be

$$\begin{aligned} \Delta_{S^2} &= \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \psi^2} \\ &= \cot \theta \frac{\partial}{\partial \theta} + \frac{\partial^2}{\partial \theta^2} + \frac{1}{\sin^2 \theta} \frac{\partial^2}{\partial \psi^2}. \end{aligned}$$

Then we need to solve $\partial_t f = \Delta_{S^2} f$ with given data $f(p, 0) = g(p)$. In order to estimate the derivatives, we need to find out the connectivity information.

```
%maximum degree computation
n_points=40962
n_tri=81844
degree=zeros(n_points,1);
for j=1:n_tri
    degree(DT(j,:))=degree(DT(j,:))+1;
end
max_degree=max(degree)

%finding connectivity
nbr=zeros(n_points,max_degree);
for i_tri=1:n_tri
```

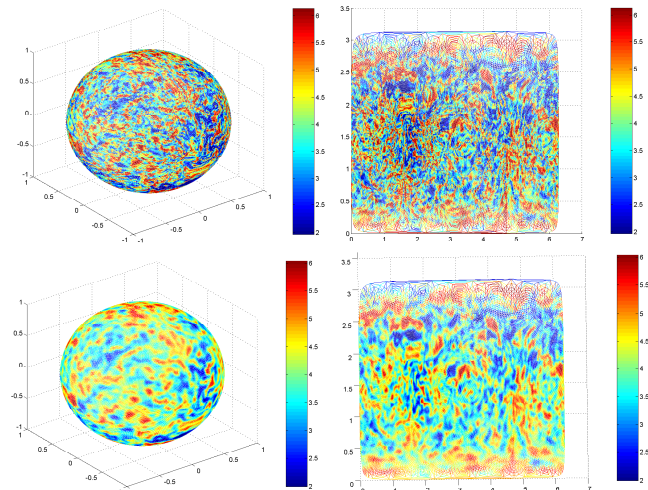


Figure 1: Top left: Cortical thickness measurements on S^2 . Top right: Parameter space D . Bottom: kernel smoothing with $\sigma = 0.01, m = 5$.

```
for j=1:3
    cur_point = DT(i_tri,j);
    for k=1:3
        if (j ~= k)
            nbr_point= DT(i_tri,k);
            if find(nbr(cur_point,:)==nbr_point),
            else
                n_nbr = min(find(...
                    nbr(cur_point,:) == 0));
                nbr(cur_point,n_nbr) = nbr_point;
            end;
        end;
    end;
end;
end;
```

It takes about 15 seconds to run the above codes in the Pentium M machine.

2. *Least squares estimation of Laplacian.* This approach is an improvement over my previous method (Chung, 2001). In the previous method, the Laplace-Beltrami operator was estimated on the

manifolds directly which was influenced by the normal vector estimation. In this parameterization approach, there is no need to estimate the normal vector to the tangent space.

To speed up the least squares estimation, we use backslash or matrix left division `\` in MATLAB. It computes the generalized inverse by Gaussian elimination. It is faster than `pinv`. For given data Y , we fit

$$Y = \beta_1 \delta \theta + \beta_2 \delta \psi + \beta_3 \delta \theta^2 + \beta_4 \delta \theta \delta \psi + \beta_5 \delta \psi^2$$

at each point (θ, ψ) . Then $\partial_\theta Y = \beta_1 + 2\beta_3 \delta \theta + \beta_4 \delta \psi$, $\partial_\theta^2 Y = 2\beta_3$ and $\partial_\psi^2 Y = 2\beta_5$. However, derivative estimation is extremely sensitive to noise. So we have to smooth noise observation Y first. So that we are fitting

$$K_\sigma * Y = \beta_1 \delta \theta + \beta_2 \delta \psi + \beta_3 \delta \theta^2 + \beta_4 \delta \theta \delta \psi + \beta_5 \delta \psi^2$$

for small σ . Parameters can be estimated via the following codes

```
beta=zeros(n_points,5);
for i=1:n_points
    n_nbr = max(find(nbr(i,:)~=0));
    nbr_list=nbr(i,1:n_nbr);
    coord_nbr=[theta(nbr_list);...
    psi(nbr_list)] -kron(ones...
    (1,n_nbr),[theta(i) ;psi(i)]);
    thick_nbr=thickness(nbr_list)...
    -ones(n_nbr,1)*thickness(i);
    x = coord_nbr(1,:)' ;
    y = coord_nbr(2,:)' ;
    % X : design matrix
    X=[x, y, x.^2, x.*y, y.^2];
    beta=(X\thick_nbr)';
end;
```

3. *Barycentric coordinates*. Parameterization gives an automatic way of estimating the Laplacian and in turn it can be used to set up diffusion smoothing. However, in a very complex manifold, simple parameterization may not be given. In that situation, we show how to solve a diffusion equation using the finite element method (FEM). Let N_T be the number of triangles in the triangular mesh S that is the discrete estimation of true surface $\partial\Omega$. We seek an approximate solution F_i in triangle T_i such that the solution $F_i(x, t)$ is continuous across neighboring triangles. The approximate solution F for the whole region is then

$$F(x, t) \doteq \sum_{i=1}^{N_T} F_i(x, t).$$

A slightly different formulation of FEM for the surface flattening problem is given in Angenent *et al.*. Let $p_{i_1}, p_{i_2}, p_{i_3}$ be the vertices of element T_i . In T_i , we interpolate F_i linearly by

$$F_i(x, t) = \sum_{j=1}^3 \xi_{i_j}(x) F(p_{i_j}, t),$$

where nonnegative ξ_{i_k} are given by the *barycentric coordinates*. In the barycentric coordinates, any point $x \in T_i$ is uniquely determined by two conditions:

$$x = \sum_{k=1}^3 \xi_{i_k}(x) p_{i_k}, \quad \sum_{k=1}^3 \xi_{i_k}(x) = 1.$$

If $x \in T_i$, we let $\xi_{i_k} = 0$. Note that $0 \leq \xi_{i_k} \leq 1$. Let G be an arbitrary piecewise linear function on S . Then it can be uniquely written as

$$G(x) = \sum_{i=1}^{N_T} \xi_{i_1}(x) G_{i_1} + \xi_{i_2}(x) G_{i_2} + \xi_{i_3}(x) G_{i_3},$$

where $G_{i_k} = G(p_{i_k})$ are the values of function G evaluated at vertex p_{i_k} of T_i .

Problem 42. Give geometric meaning to the barycentric coordinates. Given a triangle T with vertices $p_1 = (0, 0), p_2(1, 0), p_3 = (0, 1)$ with observations $f(p_i) = i^2$, interpolate $f(p)$ for any $p \in T$ via the barycentric coordinates. Plot your interpolation in MATLAB. It may be useful to use `countour` command.