DEPARTMENT OF COMPUTER SCIENCE (DIKU)
UNIVERSITY OF COPENHAGEN

Estimating the continuous entropy of a discrete set of orientations in R^3

Jon Sporring

Technical Report no. 2018/01

ISSN: 0107-8283

Estimating the continuous entropy of a discrete set of orientations in $\mathbb{R}^3$

Jon Sporring

2018/08/30

1 Theory

Consider vectors $v \in \mathbb{R}^3$ and their representations in Cartesian coordinates $v = (x, y, z)$ and in horizontal astronomical spherical coordinates (θ, ϕ, r). Here, θ is the azimuth and measures the counter-clockwise angle in the $x - y$ plan from the $(1, 0, 0)$, ϕ is the elevation and measures the angle from the $x - y$ plane, and r is the radius and measures the length of the vector. Conversion between the two representations is given by,

$$r = \sqrt{x^2 + y^2 + z^2}, \tag{1}$$

$$\phi = \operatorname{atan2}(z, \sqrt{x^2 + y^2}), \tag{2}$$

$$\theta = \operatorname{atan2}(y, x), \tag{3}$$

where atan2 is the quadrant corrected arctan function [1], and

$$z = r \sin(\phi), \tag{4}$$

$$x = r \cos(\phi) \cos(\theta), \tag{5}$$

$$y = r \cos(\phi) \sin(\theta). \tag{6}$$

Consider a continous distribution p on the unit sphere S^2 for which its continuous entropy exists,

$$H = -\int_{S^2} p \log(p)) ds. \tag{7}$$

where ds is a surface element in Cartesian coordinates, and as measured in 'nits' – the natural information unit. To evaluate the integral we use spherical coordinates, and the vectors

$$t_\theta = (\frac{\partial x}{\partial \theta}, \frac{\partial y}{\partial \theta}, \frac{\partial z}{\partial \theta}), \tag{8}$$

$$t_\phi = (\frac{\partial x}{\partial \phi}, \frac{\partial y}{\partial \phi}, \frac{\partial z}{\partial \phi}), \tag{9}$$

span the tangent plane of the sphere at a point except at the poles. The surface element is found as,

$$ds = \|t_\theta \times t_\phi\|_2 \, d\phi d\theta = r^2 |\cos(\phi)| \, d\phi d\theta, \tag{10}$$

where $\| \cdot \|_2$ is the two norm, $| \cdot |$ is the absolute function and $r = 1$ in the case of the unit sphere. Hence, combining (7), (10), and $r = 1$ we find,

$$H = -\int_{-\pi}^{\pi} \int_{-pi/2}^{pi/2} p(\theta, \phi) \log(p(\theta, \phi)) \cos(\phi) d\phi d\theta. \tag{11}$$

Consider a discrete set of N unit vectors, $V = \{v_i | i = 1 \ldots N\}$, as a realization of a sampling from p. We seek an approximation of the continuous entropy p from the samples. Let the histogram in spherical coordinates be defined as,

$$h_{jk} = |\{v_i | a_j \leq \theta_i < a_{j+1} \wedge b_k \leq \phi_i < b_{k+1}\}| \tag{12}$$

where the bin's edges a_j and b_k are monotonically non-decreasing values in j and k respectively, and $|\cdot|$ is the vector cardinality operator. We use $a_j = -\pi + j\Delta a$ and $b_k = -\pi/2 + k\Delta b$. For N samples and a given bin jk, the average histogram value is,

$$\hat{h}_{jk} = N \int_{a_j}^{a_{j+1}} \int_{b_k}^{b_{k+1}} p(\theta, \phi) \cos(\phi) d\phi d\theta. \tag{13}$$

For small bin sizes we approximate (13) as,

$$\hat{h}_{jk} \approx N p_{jk} A_{jk} \tag{14}$$

where p_{jk} is a value of $p(\theta, \phi)$ in the interval $(\theta, \phi) \in \{a_j, a_{j+1}\} \times \{b_k, b_{k+1}\}$, and A_{jk} is the area of the finite surface element corresponding to the bin,

$$A_{jk} = \int_{a_j}^{a_{j+1}} \int_{b_k}^{b_{k+1}} \cos(\phi) d\phi d\theta \tag{15}$$

$$= (a_j - a_{j+1})(\operatorname{sgn}(\cos(b_k)) \sin(b_k) - \operatorname{sgn}(\cos(b_k)) \sin(b_k)). \tag{16}$$

Here, $\operatorname{sgn}(\cdot)$ is the sign function. To estimate (11) we use a Riemann sum,

$$H \approx -\sum_j \sum_k p_{jk} \log(p_{jk}) A_j k \tag{17}$$

Combining (14) and (17) we arrive at our final result,

$$H \approx -\sum_j \sum_k \frac{\hat{h}_{jk}}{N A_{jk}} \log(\frac{\hat{h}_{jk}}{N A_{jk}}) A_{jk} \tag{18}$$

$$= -\sum_j \sum_k \frac{\hat{h}_{jk}}{N} \log(\frac{\hat{h}_{jk}}{N A_{jk}}). \tag{19}$$

This will be our estimator of the continuous density from a discrete set of samples.

2 Practice

The above theory has been implemented as a Matlab function shown in Figure 1. A uniform distribution on a sphere has value $\frac{1}{4\pi r^2}$, and the true entropy is thus, $\int_{S^2} \frac{1}{4\pi r^2} \log(4\pi r^2) ds = \log(4\pi r^2)$. For the unit sphere, it is 2.531 nits. For 1000 samples of a uniform distribution on the sphere and an angular histogram of 10 by 10 equally spaced bins, the entropy is estimated to be 2.48. Figure 2 shows the samples in both Cartesian and spherical coordinates, the histogram and the bins on the sphere. For accurate results, large sample size and small bin sizes are needed. To evaluate the quality of the estimator, 100 experiments were performed for various bin and sample sizes. The resulting entropy-estimate is shown in Figure 3. We conclude that the estimator appears to converge to the true value of the entropy for uniformly distributed data when the sample size is large and the number of bins is small relative to the sample size.

References

[1] Wikipedia. atan2. https://en.wikipedia.org/wiki/Atan2, July 30 2018.

```matlab
function [entropy, count, c1, c2] = entropyOnSphere(x,n1,n2)
  % ENTROPYONSPHERE estimate the continuous entropy of unit vectors
  %
  % [entropy, count, c1, c2] = entropyOnSphere(x,n1,n2)
  %   entropy - the resulting entropy in natural bits (nits)
  %   count - histogram counts in ab (n1xn2)
  %   c1, c2 - azimuth and elevation axes for the corners of the bins
  %     ((n1+1) and (n2+1) vectors)
  %   x - a 3xn matrix containing 3d unit vectors (ijk)
  %   n1, n2 - the number of bins used in the histogram.
  %
  % The function evaluates a histogram in azimuth-elevation coordinates
  % using cart2sph and estimates the entropy-integral on the unit sphere in
  % the ijk coordinate system.
  %
  % Example:
  %   x = rand(3,1000); x = x./(ones(3,1)*sqrt(sum(x.^2,1)));
  %   entropy = entropyOnSphere(x,10,10)
  %
  % Copyright: Jon Sporring, 2018/08/30

  UNIFORMAREA = false;

  % Estimate histogram in spherical coordinates
  % (a - azimuth, b - elevation, r - radius)
  [a,b,~] = cart2sph(x(2,:),x(1,:),x(3,:)); % ijk->xyz
  % Since c1 and c2 are edges, we add an extra right-edge
  c1 = linspace(-pi,pi,n1+1);
  c2 = linspace(-pi/2,pi/2,n2+1);
  if UNIFORMAREA
    % Sampling uniformly by area, we resample c2
    t = -(sqrt(cos(c2(1:end-1)).^2).*...
      tan(c2(1:end-1))-sqrt(cos(c2(2:end)).^2).*tan(c2(2:end)));
    s = [0,cumsum(t)];
    c2 = interp1(s,linspace(-pi/2,pi/2,length(s)),...
      linspace(0,s(end),length(s)));
  end
  count = hist3([a',b'],'Edges', {c1,c2});

  % Calculate area of surface elements
  area = zeros(length(c1),length(c2));
  area(1:end-1,1:end-1) = (c1(1:end-1)-c1(2:end))'.*...
    (sign(cos(c2(1:end-1))).*sin(c2(1:end-1))...
      -sign(cos(c2(2:end))).*sin(c2(2:end)));

  % Estimate the per area element density and continues entropy:
  % average_histogram_count = total_count * density * bin_area.
  p = zeros(size(count));
  % We ignore very small areas for numerical stability
  ind = count>0 & area>0.0001/(n1*n2);
  p(ind) = count(ind)/sum(count(ind))./area(ind);
  p = p/sum(p(ind).*area(ind));
  entropy = -sum(sum(p(ind).*log(p(ind)).*area(ind)));

  % correct hist3-output
  count = count(1:end-1,1:end-1);
```

Figure 1: Matlab function estimating the continuous entropy on a sphere.

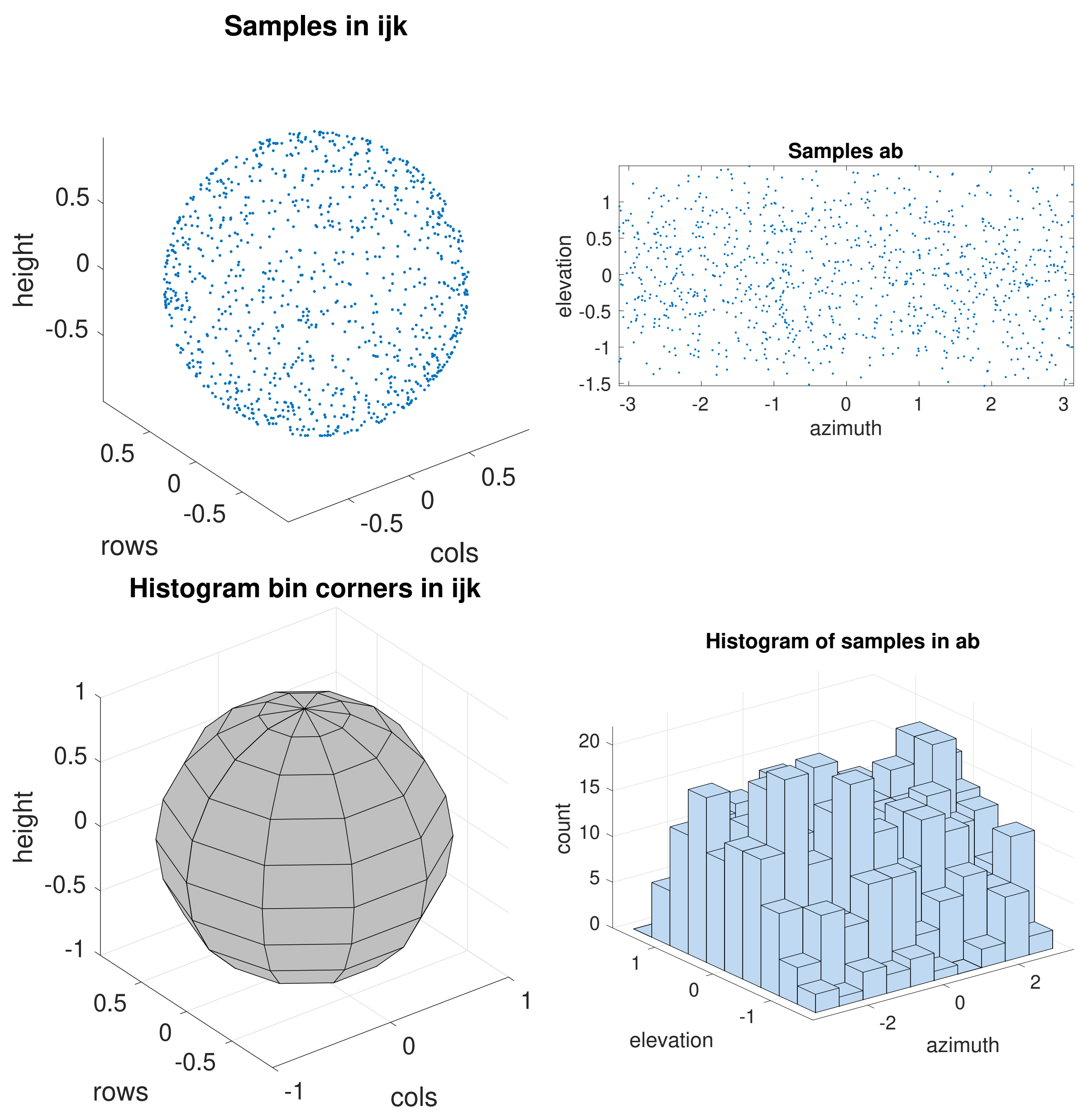

Figure 2: A sample from a uniform distribution.

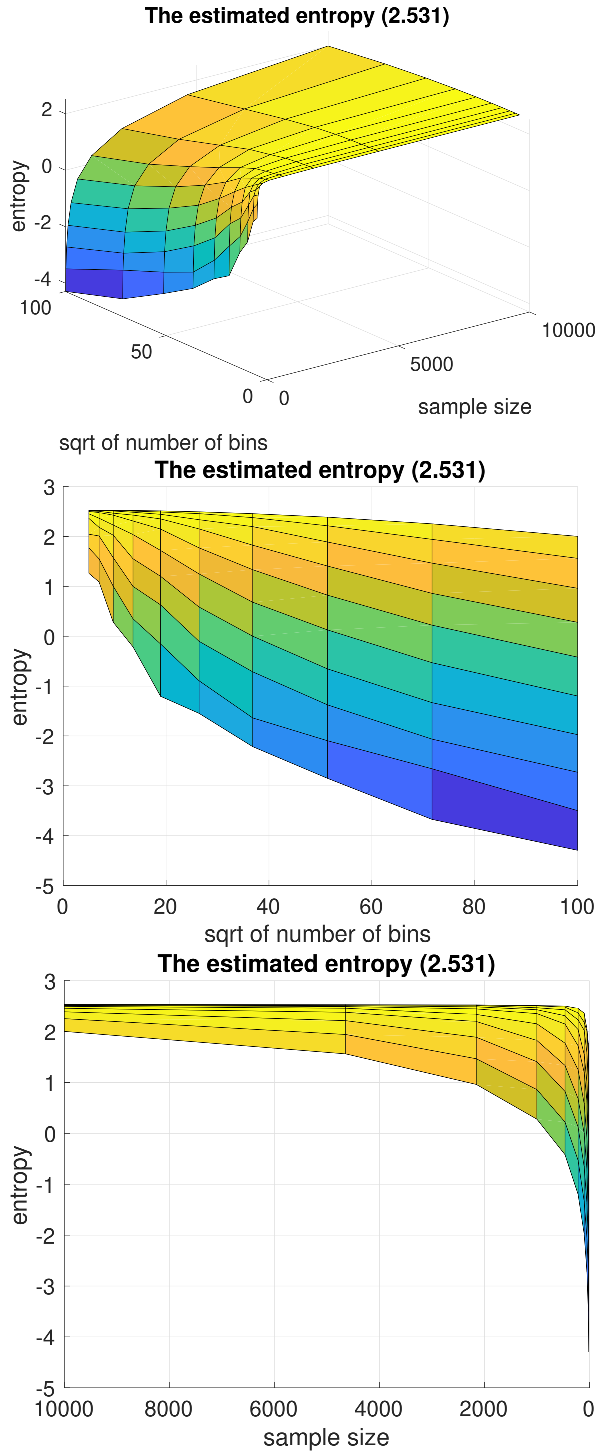

Figure 3: 100 experiments for various sample and bin size. The figures shows 3 different views of the same surface.