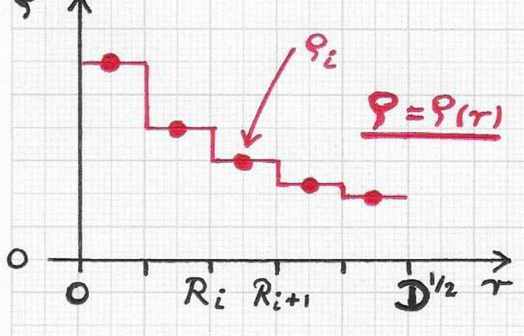
OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks: ... Best approximation of data via the least-squares method applied to a chosen model function can sometimes lead to results that are not acceptable. For example, the described model function $\varrho(\mathbf{u}) = A \exp(-B \sum_i u_i^2)$, once the values of A and B are known, is not necessarily positive everywhere. One can introduce and use additional constraints for the general model function $\varrho(\mathbf{u})$ or simply "re-set" or clamp the model function's value to ε, $\varepsilon \ll 10^{-10}$, whenever the calculated value of $\varrho(\mathbf{u})$ is smaller than ε. From a practical perspective, it is therefore highly desirable not to have to worry about this issue. One simple solution approach for this problem is the use of a piecewise-constant or piecewise-linear function $\varrho(\mathbf{u})$ that can, a priori, never become negative when all data/values to be approximated or interpolated by ϱ are not negative. The left figure

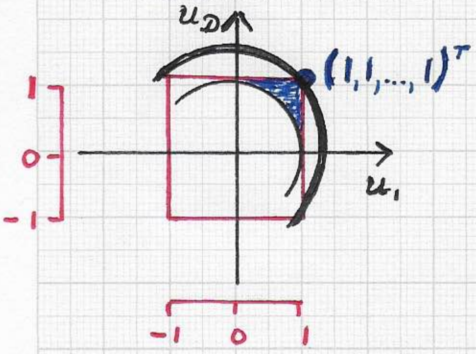


is based on the technique described on pp. 8–10 (4/24-25/2023) for the purely radius-based estimation of non-negative ϱ_i-values of probability density for intervals $[R_i, R_{i+1})$. This technique assumes that ϱ in D-dimensional "normalized" $\mathbf{u}$-space is univariate, depending on radius r ($r^2 = u_1^2 + \ldots + u_D^2$) only.

...

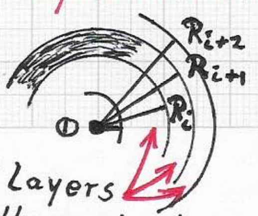
OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...



The left figure serves the purpose of pointing out that the maximal r-value to be used for the determination of discrete point densities in u-space is the length of the positional vector of one of the corner vertices of the "normalized" minimal hyper-cube $[-1,1]^D$, e.g., $(1,1,...,1)^T$.

This maximal r-value is $r = (1+1+...+1)^{1/2} = \sqrt{D}$. Thus, for the purpose of constructing the univariate probability density function $\varrho(r)$, we use as input the r-values $R_0 = 0$, $R_1 = \Delta, ..., R_i = i\Delta, ..., R_N = N\Delta$, where $\Delta = \sqrt{D}/N$ and $i = 0...N$; further, we calculate/know the estimated discrete density value ϱ_i with the interval $[R_i, R_{i+1})$, $i = 0...N$. Alternatively, one might prefer to subdivide the hyper-volume of a D-ball with radius $\sqrt{D}$ into hyper-subvolumes that have equal magnitude and are bounded by two concentric hyper-spheres; in this case one must determine the radii $R_0, R_1, ..., R_N$ that induce constant hyper-subvolumes for every region in D-dimensional space between the two hyper-spheres with radii R_i and R_{i+1}, $R_i < R_{i+1}$. The formulae for the hyper-volume of a D-ball are given on p. 9 (4/25/2023). The left figure sketches the N hyper-subvolume layers constituting a D-ball with radius $\sqrt{D}$ for which we must compute R_i-values.



N Layers with constant hyper-subvolumes

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ... The hyper-volume of the D-ball with maximal radius $\sqrt{D}$ is V. The value of V is $V = const \cdot (\sqrt{D})^D = const \cdot D^{D/2}$. We must determine the radii R_i, $i = 0 \ldots N$, of D-balls with hyper-volumes that have values given as $^0/_N\, V$, $^1/_N\, V, \ldots, {}^N/_N\, V$. Thus, the R_i-values are defined by the equation

$$V_i = const \cdot (R_i)^D = {}^i/_N\, V = {}^i/_N \cdot const \cdot D^{D/2}.$$

Solving the equation for R_i, one obtains

$$const \cdot (R_i)^D = {}^i/_N \cdot const \cdot D^{D/2} \Leftrightarrow R_i^{\,D} = {}^i/_N\, D^{D/2} \Leftrightarrow R_i = \sqrt[D]{{}^i/_N} \cdot \sqrt{D}.$$

Using the expressions for R_i and R_{i+1}, one can compute the hyper-subvolume of the subvolume bounded by the two hyper-spheres with radii R_i, R_{i+1}:

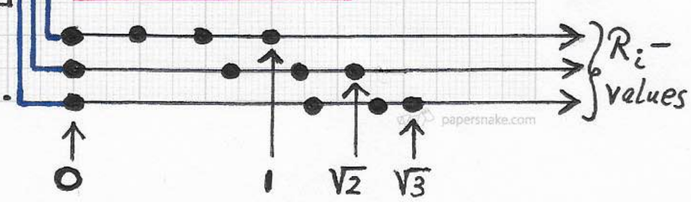
$$\begin{aligned} \text{Layer Volume} = V_{i+1} - V_i &= const\left(\left(\tfrac{i+1}{N}\right)^{\frac{1}{D}} \sqrt{D}\right)^D - const\left(\left(\tfrac{i}{N}\right)^{\frac{1}{D}} \sqrt{D}\right)^D \\ &= const\left(\tfrac{i+1}{N} D^{D/2} - \tfrac{i}{N} D^{D/2}\right) \\ &= {}^1/_N \cdot const \cdot D^{D/2} = {}^1/_N\, V. \end{aligned}$$

Therefore, each layer has the same hyper-subvolume, i.e., $^1/_N\, V$, by construction. The table (left) uses the formulae

D \ i	0	1	2	3
1	0	1/3	2/3	1
2	0	0.816	1.155	$\sqrt{2}$
3	0	1.201	1.513	$\sqrt{3}$

R_i-values for $D = 1, 2, 3$ and $N = 3$ ($i = 0 \ldots 3$) needed for 3 layers.

$R_i = \sqrt[1]{{}^i/_3} \cdot \sqrt{1}$, $D = 1$;
$R_i = \sqrt[2]{{}^i/_3} \cdot \sqrt{2}$, $D = 2$; and
$R_i = \sqrt[3]{{}^i/_3} \cdot \sqrt{3}$, $D = 3$.



...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

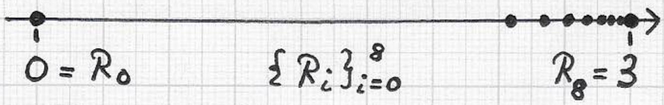
• Laplacian eigenfunctions and neural networks: ...

Example: $D = 9$, $N = 8$

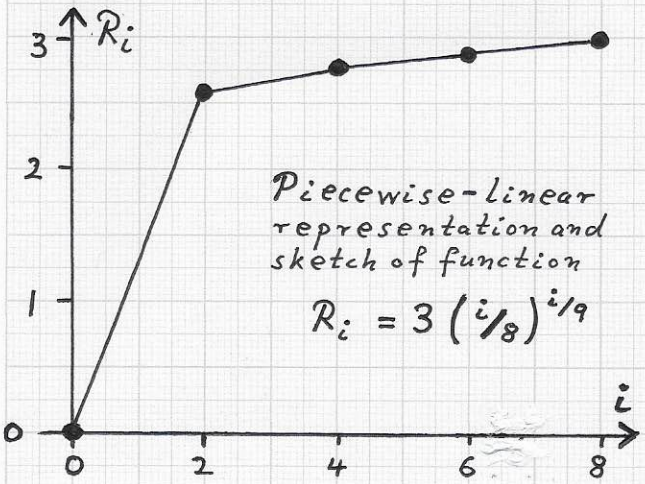
$\Rightarrow R_i = \sqrt{D} \cdot \left(\frac{i}{8}\right)^{\frac{i}{D}} = 3\left(\frac{i}{8}\right)^{\frac{i}{9}}$

The detailed example included here (left) considers the 9-dimensional case.

i	0	1	2	3	4	5	6	7	8
R_i	0	2.38	2.57	2.69	2.78	2.85	2.91	2.96	3



The highly non-linear nature of the function $R_i = (i/8)^{i/9} \cdot 3$ is a direct consequence of the exponent $i/9$.



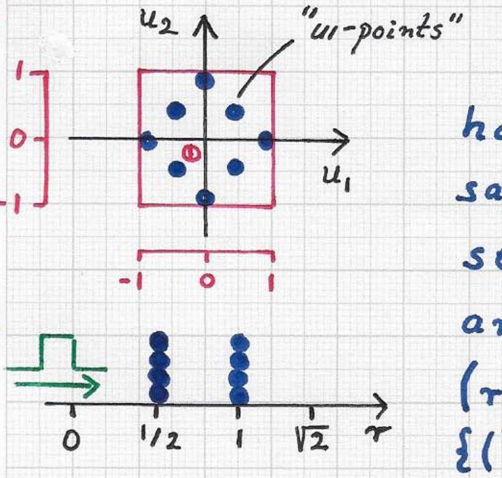
Our goal is the construction of a "simple" and computationally efficient probability density function $\rho(r)$ based on calculated, data-based discrete tuples (R_i, R_{i+1}, ρ_i). The value of ρ_i applies to the interval $[R_i, R_{i+1})$, $i = 0 \ldots (N-1)$. IT IS NOT OBVIOUS WHETHER ONE SHOULD USE A CONSTANT INTERVAL LENGTH, i.e., $R_{i+1} - R_i = \text{const}$, OR A CONSTANT LAYER HYPER-VOLUME, i.e., $V_i = \text{const}$, FOR THE CONSTRUCTION OF $\rho(r)$.

• **Note.** Yet another possibility for the construction of $\rho(r)$ is the calculation and use of a discrete histogram of the (radial) distance values of all "ui-points" from the center of the hyper-cube $[-1,1]^D$.

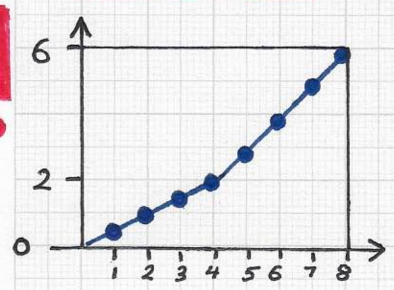
...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

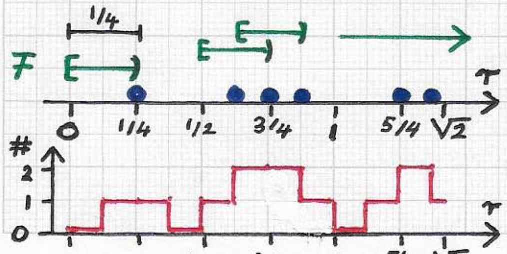
• Laplacian eigenfunctions and neural networks: ...



We consider a simple example for $D=2$, with 8 points in the $[-1,1]^2$ ui-domain. Four points have coordinates leading to the same (radial) distance **1** – the set $\{(1,0)^T, (0,1)^T, (-1,0)^T, (0,-1)^T\}$ – and four points lead to the same (radial) distance $1/2$ – the set $\{(\sqrt{2}/4, \sqrt{2}/4)^T, (-\sqrt{2}/4, \sqrt{2}/4)^T, (-\sqrt{2}/4, -\sqrt{2}/4)^T, (\sqrt{2}/4, -\sqrt{2}/4)^T\}$, see top-left figure, where the bottom sketch shows the resulting discrete histogram. The actual indices of the "ui-points" are not needed in this context.



The cumulative distance values one obtains when traversing the histogram's r-axis from left to right via a "quasi-convolution filter mask F" are shown in the left figure. The sequence of cumulative distance values is: $1/2, 2/2, 3/2, 4/2, 3, 4, 5, 6$. (These values can be normalized by dividing them by 6.)



Convolution. The left sketches show an example for $D=2$ where a window, an interval F of length $1/4$, is moved from left to right through the r-domain. One counts the number of '•' in F and assigns it as $\#(r)$ to F's midpoint. ...