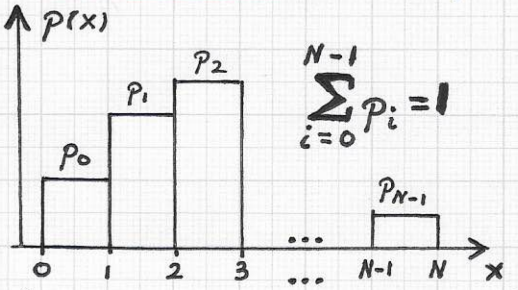
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

• **Note.** Probability – Moments – Generating Functions etc. For our purposes, it is not necessary to discuss these complex and related topics in detail. We must consider that we are merely concerned with point density / distribution functions defined over unit, normalized D-dimensional ball domains; and the independent variable is distance-from-ball-center, i.e., radius. Nevertheless, it is important to realize that invariants, probability, moments, Taylor series, best approximation / least-squares approximation, generating functions etc. are related to our driving problem and are also inter-related topics. Therefore, some of the highest-level ideas and concepts are briefly mentioned at this point, including good references.



In the context of histogram-based characterization of image intensity, involving N discrete integer intensity levels $0, 1, \ldots, N-1$, one can use moments of univariate distributions. The figure illustrates such a distribution with N levels and associated probability values $p_i = p(i)$, $i = 0 \ldots N-1$. Further, one should assume that this piecewise constant function is "normalized," i.e., $\|p(x)\| = (p_0 + p_1 + \ldots + p_{N-1}) = 1$. (Probabilities add to 1.)

• • •

OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

- Laplacian eigenfunctions and neural networks: ... (Reference: Fu, Gonzalez and Lee, "Control, Sensing, Vision and Intelligence - ROBOTICS," McGraw-Hill, sections discussing moments.) The mean (average) is defined as $m = \sum_{i=0}^{N-1} x_i p_i$. After mean subtraction, the j^{th} moment (about the mean) is $\mu_j = \sum_{i=0}^{N-1} (x_i - m)^j p_i$, $j = 1, 2, 3, \ldots$ Formally, one can compute the 0^{th} and 1^{st} moments:

$$\mu_0 = \sum_{i=0}^{N-1} (x_i - m)^0 p_i = \sum_{i=0}^{N-1} = 1 \ ;$$

$$\mu_1 = \sum_{i=0}^{N-1} (x_i - m)^1 p_i = \sum_{i=0}^{N-1} x_i p_i - \sum_{i=0}^{N-1} \left(\sum_{k=0}^{N-1} x_k p_k\right) p_i$$
$$= m - \sum_{i=0}^{N-1} m p_i = m \left(1 - \sum_{i=0}^{N-1} p_i\right) = m \cdot 0 = 0.$$

Again, the second, third and fourth moments, i.e., μ_2 $(= \sigma^2)$, μ_3 and μ_4 define variance, histogram skewness and relative histogram flatness/peakedness ("kurtosis").

Moments in physics (mechanics). Understanding a point as a "physical point" in 3-dimensional space with an associated "point mass" m, for example, the n^{th} moment μ_n is defined as $\mu_n = r^n m$, where r is the distance to the point. The moment is defined in integral form when a mass is distributed in space according to some mass density $\rho(r)$. The n^{th} moment is thus defined as $\mu_n = \int r^n \rho(r)\, dr$. Total mass, center of mass and the moment of inertia can be defined for point masses and mass distributions.

...

OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

- Laplacian eigenfunctions and neural networks:... (In the context of probability theory, one can view "mass density" as equivalent to "probability density function.") The mentioned mass moments are:
  - Total mass: 0^{th} mass moment (μ_0)
  - Mass center: $\mathbb{r} = 1/\mu_0 \sum_i \mathbb{r}_i m_i$ (discrete mass points);
    $\mathbb{r} = 1/\mu_0 \int \mathbb{r}\, \rho(\mathbb{r})\, d^3 r$ (mass distribution)
  - Inertia moment: $I = \sum_i r_i^2 m_i$ (discrete mass points);
    $I = \int r^2 \rho(\mathbb{r})\, d^3 r$ (mass distribution $\rho(\mathbb{r})$)

(Here, the vector-valued notation "$\mathbb{r}$" is used to refer to a point/positional vector in space; "r" refers to distance.)

Generating functions and moments. (Reference: Knuth, "The Art of Computer Programming," Vol. 1, Addison Wesley, 3rd ed., sections 1.2.8, 1.2.9 and 1.2.10.) Donald Knuth considers generating functions in the context of determining a closed-form (function) definition of a (recurrence-based) number sequence and analyzing an algorithm's computational complexity via means of probability. Moments are discussed in this context. Knuth writes a probability distribution as $p_0 + p_1 z + p_2 z^2 + \ldots$ and a moment M_n as $M_n = \sum_k k^n p_k$, and a central moment m_n as $m_n = \sum_k (k - M_1)^n p_k$. "Semi-invariants" $\tilde{\kappa}_n$ were described by Thiele (1903), and Knuth provides this formula:

$$\tilde{\kappa}_n = \sum_{\substack{k_1, k_2, \ldots, k_n \geq 0 \\ k_1 + 2k_2 + \ldots = n}} \frac{(-1)^{k_1+k_2+\ldots+k_n-1}(k_1+k_2+\ldots+k_n-1)!}{k_1!\,1!^{k_1}\,k_2!\,2!^{k_2}\ldots k_n!\,n!^{k_n}} M_1^{k_1} M_2^{k_2} \ldots M_n^{k_n}.$$

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... Specifically, the first of these "semi-invariants" are:

$$\tilde{K}_1 = M_1 \quad ;$$
$$\tilde{K}_2 = M_2 - M_1^2 \quad ;$$
$$\tilde{K}_3 = M_3 - 3M_1M_2 + 2M_1^3 \quad ;$$
$$\tilde{K}_4 = M_4 - 4M_1M_3 + 12M_1^2M_2 - 3M_2^2 - 6M_1^4 \, .$$

(One should compare these formulae to those on pp. 11-12, 4/28-29/2023.)

Best approximation, least squares and inner products.
Best approximation provides us with yet another perspective on probability distributions: The set $\{1, x, x^2, x^3, \dots\}$ defines a basis of all polynomials, and one can use this basis for computing a given function's best polynomial approximation – involving the function's "projections" onto the basis polynomials via inner product computations. The monomials $1, x, x^2, \dots$ are not mutually orthogonal, and one therefore generates orthogonal polynomials to make the computations of coefficients of best polynomial function approximations efficient. For example, one can employ Gram-Schmidt orthonormalization to establish an orthonormal basis $\{p_0(x), p_1(x), \dots, p_n(x)\}$ for the space of polynomials of degree $\leq n$, relative to a specific interval over which $f(x)$ is approximated by $a(x) = \sum_{i=0}^{n} c_i \cdot p_i(x)$. In this case, $c_i = \langle p_i(x), f(x) \rangle = \int_{\text{domain}} p_i(x) f(x)\, dx$.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- **Laplacian eigenfunctions and neural networks: ...** For example, one can call the monomials $f_0(x)=1$, $f_1(x)=x$, $f_2(x)=x^2$, ...

Further, one might use the interval $[0,1]$ as the domain over which orthonormalization must be performed. For the normalization of a function, one usually employs the norm $\|f\| = (\langle f,f\rangle)^{1/2} = (\int f^2\,dx)^{1/2}$.

Orthonormalization of the monomials via the Gram-Schmidt process uses the following computational sequence:

- $f_0(x)=1$; $\tilde{p}_0(x) = f_0(x) = 1$, $\|\tilde{p}_0\| = (\int_0^1 1\,dx)^{1/2} = 1$

  $\Rightarrow p_0(x) = \tilde{p}_0 / \|\tilde{p}_0\| = 1$

- $f_1(x)=x$; $\tilde{p}_1(x) = f_1(x) - \langle f_1(x), p_0(x)\rangle p_0 = x - \left(\int_0^1 x\cdot 1\,dx\right)\cdot p_0$

  $= x - \frac{1}{2}x^2\Big|_0^1 = x - \frac{1}{2}$,

  $\|\tilde{p}_1\| = (\int_0^1 (x-\frac{1}{2})^2\,dx)^{1/2} = \sqrt{3}/6$

  $\Rightarrow p_1(x) = \tilde{p}_1 / \|\tilde{p}_1\| = (2x-1)\sqrt{3}$

- $f_2(x)=x^2$; $\tilde{p}_2(x) = f_2(x) - \langle f_2(x), p_0(x)\rangle\cdot p_0(x)$

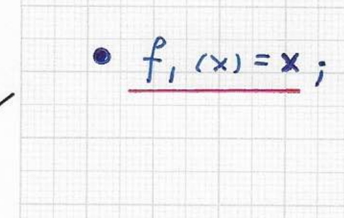
  $- \langle f_2(x), p_1(x)\rangle\cdot p_1(x) = x^2 - x + \frac{1}{6}$

  $\|\tilde{p}_2\| = (\int_0^1 (x^2 - x + 1/6)^2\,dx)^{1/2} = \sqrt{5}/30$

  $\Rightarrow p_2(x) = \tilde{p}_2 / \|\tilde{p}_2\| = (6x^2-6x+1)\sqrt{5}$

Thus, $\{p_0(x), p_1(x), p_2(x)\}$ is an orthonormal polynomial basis.

The orthonormal polynomials $p_i(x)$ can be defined via moments.



$$p_k(x) = c_k \cdot \begin{vmatrix} m_0 & m_1 & m_2 & \dots & m_k \\ m_1 & m_2 & m_3 & \dots & m_{k+1} \\ m_2 & m_3 & m_4 & \dots & m_{k+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{k-1} & m_k & m_{k+1} & \dots & m_{2k-1} \\ 1 & x & x^2 & \dots & x^k \end{vmatrix}, \quad k = 0 \dots n.$$

moments

The polynomials $p_k(x)$ are products of a constant c_k and the determinant included here. Further, $m_k = \int_0^1 x^k\,d\alpha(x) = \int_0^1 x^k\,w(x)\,dx$, where $w(x) \geq 0$ is a weight function.

...