Bernd Hamann

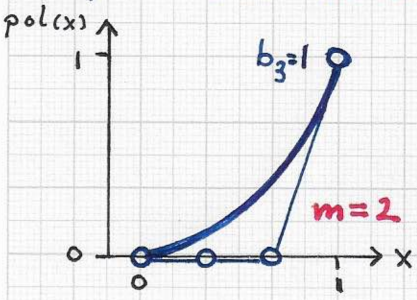
Stratovan

5/9/2022

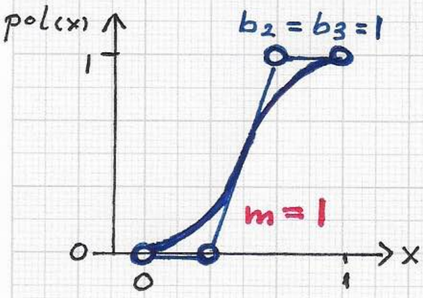
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

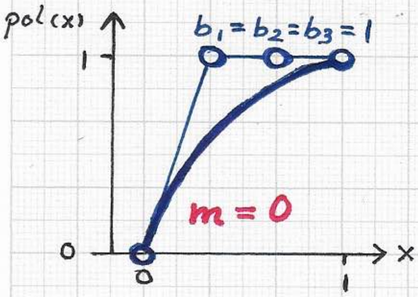
The three possible cubic polynomials in Bernstein-Bézier representation ($n=3$):



$pol(x) = B_3^3(x) = x^3$



$pol(x) = B_2^3(x) + B_3^3(x) = 3x^2 - 2x^3$



$pol(x) = B_1^3(x) + B_2^3(x) + B_3^3(x)$
$= 3x - 3x^2 + x^3$

Three possible tuples $(b_0, b_1, b_2, b_3) = \mathbb{b}$: $(0,0,0,1)$, $(0,0,1,1)$, $(0,1,1,1)$.

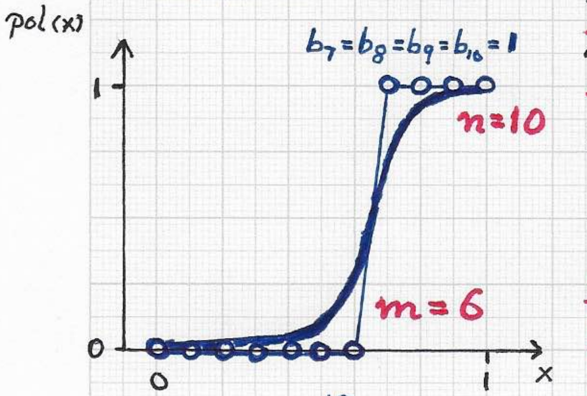
For example, every polynomial in Bernstein-Bézier representation $pol(x) = \sum_{i=0}^{n} b_i B_i^n(x)$, $x \in [0,1]$, with $b_0 = b_1 = \ldots = b_m = 0$ and $b_{m+1} = \ldots = b_{n-1} = b_n = 1$ satisfies the fundamental requirement to be **monotonic** – and therefore suitable as a "RAMP" function. **Thus, the values of m and n are the two degrees of freedom for "RAMP" function design. The only design requirements are that (i) $n \geq 1$, (ii) $b_0 = 0$ and (iii) $b_n = 1$.** Considering the special case of the quintic Hermite polynomial shown on the previous page, the defining values are $m=2$ and $n=5$. Thus, one obtains

$$pol(x) = \sum_{i=0}^{5} b_i B_i^5(x) = \sum_{i=3}^{5} 1 \cdot B_i^5(x)$$
$$= B_3^5(x) + B_4^5(x) + B_5^5(x)$$
$$= \binom{5}{3}(1-x)^2 x^3 + \binom{5}{4}(1-x)x^4 + \binom{5}{5}x^5$$
$$= \ldots = \mathbf{6x^5 - 15x^4 + 10x^3}.$$

This result is indeed the specific quintic Hermite polynomial in monomial form. ...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...



$$pol(x)=\sum_{i=7}^{10} B_i^{10}(x)$$

Example of a degree-10 polynomial in Bernstein-Bézier representation.

The degrees of freedom, m and n, make it possible to design and optimize the shape of the shown prototype "RAMP" function:

- The value of m defines the location of maximal slope of the "RAMP."
- The value of n determines whether the "RAMP" transitions from 0 to 1 rather slowly, gradually or abruptly, in a nearly step function manner.

For example, when varying the value of n from 1 to 10, the total number of possibilities is 55.

The individual decider functions $F_i(p_i)$ are precisely defined by the "expert" for $0 \le p_i \le p_1^i$ (where F_i is 0) and $p_2^i \le p_i \le 1$ (where F_i is 1). Therefore, the behavior of F_i in the interval (p_1^i, p_2^i) is crucial for the optimization of overall classification results. $F_i(p_i)$ is monotonic in (p_1^i, p_2^i), but F_i's behavior in this "RAMP" interval is not constrained a priori by other conditions. THUS, IT IS IMPERATIVE THAT F_i's "RAMP" IS DESIGNED 'NEARLY OPTIMALLY' VIA EXTENSIVE TRAINING, TESTING AND VALIDATION.

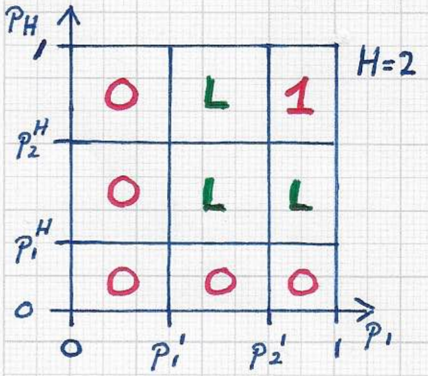
For a fixed value of n, the number of possible **b**-tuples is n; the possible **b**-tuples are $(0,0,\ldots,0,1)$, $\ldots$, $(0,1,\ldots,1,1)$. Further, if one allowed the value of n to vary between 1 and N, the total number of possible **b**-tuples would be

$$1+2+\ldots+N = N(N+1)/2.$$

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...



Example for H = 2 demonstrating the relevance of the design of the decider function in, e.g., multi-linear, type-L cuboids.

The number of type-0 cuboids is $3^2-2^2 = 5$; the number of type-L cuboids is $2^2-1^2 = 3$; and the number of type-1 cuboids is $1^2-0^2 = 1$.

The design, the behavior of the individual univariate decider functions $F_i(p_i)$ in the open interval (p_1^i, p_2^i) and the overall decider function $F(p_1, \ldots, p_H)$ in the type-L cuboids is supremely important for classification performance.

• 'L' indicates basic multi-LINEAR variation over a cuboid; of course, the monotonic "RAMP" function for the interval (p_1^i, p_2^i) can be more general.

• **Note.** The figure (left) is used to remind us that the ratio of the numbers of type-0, type-L (e.g., multi-linear) and type-1 cuboids in the described Cartesian/tensor product scenario is

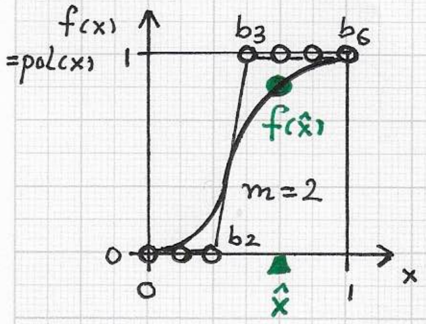
$$(3^H-2^H) : (2^H-1^H) : (1^H-0^H)$$
$$= (3^H-2^H) : (2^H-1) : 1 .$$

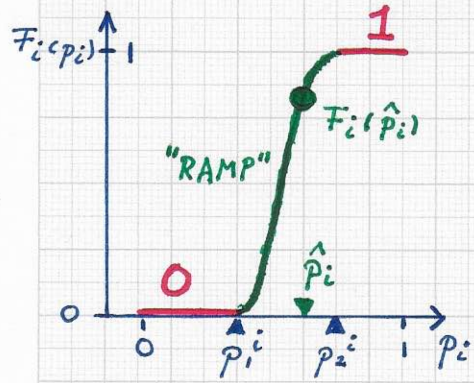
Thus, regardless of the value of the dimension H (= number of scales) THE NUMBER OF CUBOIDS REPRESENTING "BELONGS TO CLASS" IS ALWAYS 1. (This one cuboid is the cuboid $[p_2^1, 1] \times \ldots \times [p_2^H, 1]$.) Since the number of type-L cuboids is 2^H-1, and since the 'classification boundary/interface' between the "BELONGS-TO-CLASS REGION" and "DOES-NOT-BELONG-TO-CLASS REGION" lies in the type-L cuboids, THE DESIGN OF THE UNIVARIATE DECIDER FUNCTIONS $F_i(p_i)$ IS CRUCIALLY IMPORTANT FOR CLASSIFICATION.

...

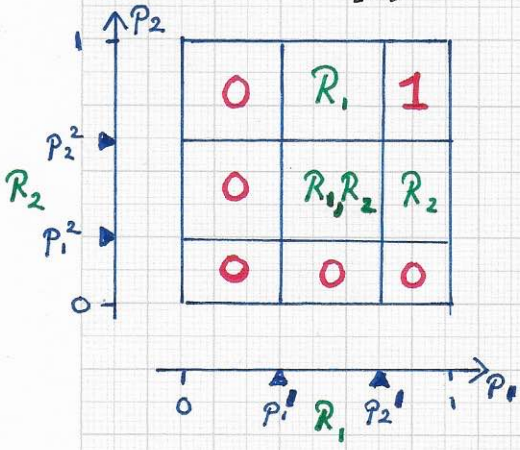
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...





Linear parameter transformation necessary for computation of $F_i(\hat{p}_i)$. Here, $pol(x) = \sum_{i=3}^{6} B_i^6(x)$.



2D scenario showing how and which "RAMP" functions are 'combined' in the Cartesian product domain.

Parameter transformation. The figure (left) illustrates a linear parameter transformation that one should perform when evaluating a univariate decider function $F_i(p_i)$ for a value $\hat{p}_i$ in the p_i-domain, inside the "RAMP" interval (p_1^i, p_2^i). The 'prototype' or 'normalized' "RAMP" function is a monotonic function defined over the interval $[0,1]$ – e.g., the Bernstein-Bézier polynomial $pol(x)$ in the top figure. When computing $F_i(\hat{p}_i)$, one calculates

$$\hat{x} = (\hat{p}_i - p_1^i) / (p_2^i - p_1^i) = \delta_i / \Delta_i \quad \text{and}$$

$$F_i(\hat{p}_i) = f(\hat{x}) = pol(\hat{x}).$$

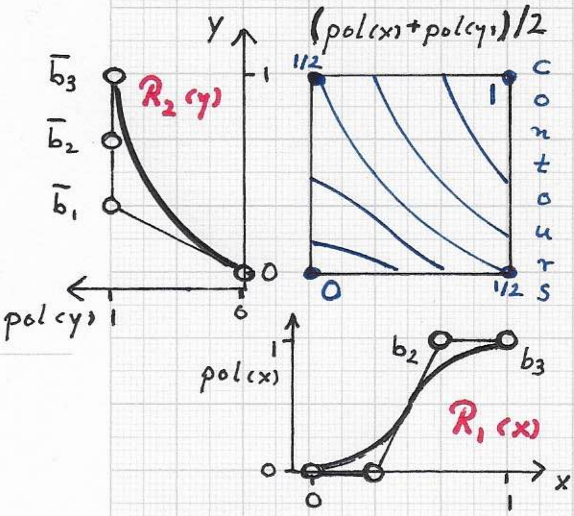
The figure (left, bottom) is merely provided to remind us that the open intervals (p_1^i, p_2^i) have a "general R-function" defined for them; an R-function defines the "RAMP" of an individual decider function $F_i(p_i)$ and monotonically increases from 0 to 1. "R functions" must be 'combined' in certain cuboids, producing R_1; R_2; R_1, R_2.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

Examples of possible 'combinations' of prototype, normalized R-functions:



Definition of a 'combination' of two prototype R-functions – Bernstein-Bézier polynomials – for the unit cuboid (square) via average:

$$pol(x) = \sum_{i=2}^{3} B_i^3(x) = 3x^2 - 2x^3$$

$$pol(y) = \sum_{j=1}^{3} B_j^3(y) = 3y - 3y^2 + y^3$$

$$\Rightarrow (pol(x)+pol(y))/2 = (3x^2-2x^3+3y-3y^2+y^3)/2 = R_1(x) \circledast R_2(y) = R_1 \circledast R_2 = R_{1,}R_2$$

$\Rightarrow$ ADDITIVE 'combination': $R_{1,}R_2 = (R_1+R_2)/2$.

Just like the L-functions described previously (L standing for Linear and multi-linear functions), R-functions must be defined for certain (2^H-1) cuboids in the H-dimensional Cartesian product domain space of a decider function $F(p_1,\dots,p_H)$. The 'combination' of R-functions with the 0-function, 1-function or with only R-functions for a specific cuboid can be defined in different ways. For example, two R-functions can be combined in an ADDITIVE or MULTIPLICATIVE way:

$$R(p_1,p_2) = (R_1(p_1) + R_2(p_2))/2$$

and

$$R(p_1,p_2) = R_1(p_1) \cdot R_2(p_2)$$

are two specific ways one can combine two R-functions over the Cartesian product cuboid of their two "RAMP" intervals. We write the combination of an $R_1(p_1)$-function and the 0-, 1- or an $R_2(p_2)$-function as $R_1 \circledast 0 = 0$, $R_{\blacksquare} \circledast 1 = R_{\blacksquare}$ or $R_1 \circledast R_2 = R_{1,}R_2$. ...