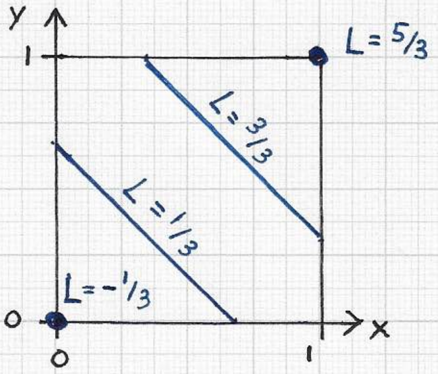
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...



Contours of $L = -\frac{1}{3} + x + y$, the best linear approximation of $\varrho = x^2 + y^2$.

L is not a "restricted" convex combination of $F_1 = x$ and $F_2 = y$. For the purpose constructing $F(x,y)$ as a convex combination, **L** is not allowed.

• Note. The minimal least-squares error between L and ϱ is

$$\int_0^1\int_0^1 (L-\varrho)^2\,dx\,dy = 1/90. \approx \mathbf{0.01}.$$

The minimal least-squares error between the optimal convex combination of $F_1 = x$ and $F_2 = y$ and ϱ is

$$\int_0^1\int_0^1 (F-\varrho)^2\,dx\,dy = 29/360 \approx \mathbf{0.08}.$$

• **It is unclear how to incorporate the aspect of function NORMALIZATION...**

• **Note.** ... For example, the "full" linear polynomial $\mathbf{L(x,y)}$ uses basis functions $\mathbf{1}$, $\mathbf{x}$ and $\mathbf{y}$. When optimally approximating $\varrho(x,y) = x^2 + y^2$, the resulting **3×3** system is

$$\begin{bmatrix} \langle 1,1\rangle & \langle 1,x\rangle & \langle 1,y\rangle \\ \langle x,1\rangle & \langle x,x\rangle & \langle x,y\rangle \\ \langle y,1\rangle & \langle y,x\rangle & \langle y,y\rangle \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{1,0} \\ c_{0,1} \end{bmatrix} = \begin{bmatrix} \langle 1, x^2+y^2\rangle \\ \langle x, x^2+y^2\rangle \\ \langle y, x^2+y^2\rangle \end{bmatrix},$$

evaluated for the domain $[0,1]^2$, leading to

$$\begin{bmatrix} 1 & 1/2 & 1/2 \\ 1/2 & 1/3 & 1/4 \\ 1/2 & 1/4 & 1/3 \end{bmatrix} \begin{bmatrix} c_{0,0} \\ c_{1,0} \\ c_{0,1} \end{bmatrix} = \begin{bmatrix} 2/3 \\ 5/12 \\ 5/12 \end{bmatrix} \Rightarrow \begin{bmatrix} c_{0,0} \\ c_{1,0} \\ c_{0,1} \end{bmatrix} = \begin{bmatrix} -1/3 \\ 1 \\ 1 \end{bmatrix}.$$

The resulting best approximation is

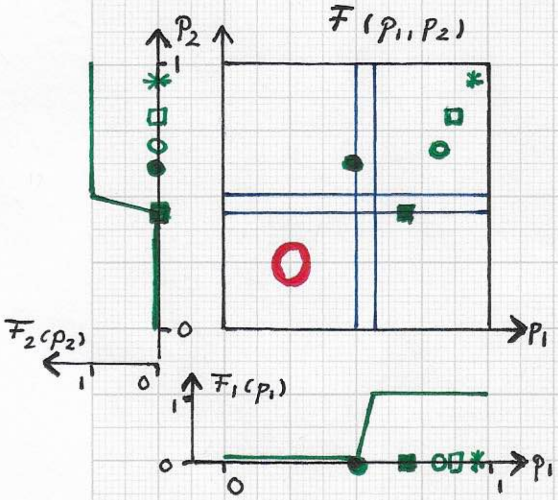
$$\mathbf{L(x,y) = -1/3 + 1x + 1y}.$$

• **Note.** Merely for exploratory curiosity one could also use a linear function of the form $\hat{F}(x,y) = c_{0,0} + w F_1(x) + (1-w) F_2(y)$ $= c_{0,0}\cdot 1 + wx + (1-w)y \equiv c_{0,0}\cdot 1 + (x-y)w + y$. Thus, one computes a least-squares solution for $1\cdot c_{0,0} + (x-y)\,w = \varrho(x,y) - y = x^2 + y^2 - y$. The resulting optimal values are $c_{0,0} = 1/6$ and $w = 1/2$, i.e., $\hat{F} = 1/6 + 1/2\,x + 1/2\,y$. $\int_0^1\int_0^1 (\hat{F}-\varrho)^2\,dx\,dy = 19/360 \approx 0.05$ is the error.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS — Cont'd.

• Laplacian eigenfunctions and neural networks: ...



Samples have associated tuples (p_1, p_2) in the 2D domain of F. The shown five tuples project to five corresponding points on the p_1- and p_2-line, respectively.

The O-region is defined by the minimal p_1- and p_2-values. The O-region's right boundary is defined by the minimal p_1-value; its top boundary is defined by the minimal p_2-value.

The best-possible model for and computation especially of the "RAMP" region of $F(p_1, p_2)$ is crucial for F's ultimate classification performance.

An important aspect to explore and to understand is the relationship between the sets $\{p_j^1\}_{j=1}^S$, $\{p_j^2\}_{j=1}^S$ and $\{(p_j^1, p_j^2)\}_{j=1}^S$, where S is the number of material samples used to define the individual univariate decider functions $F_1(p_1)$ and $F_2(p_2)$. One can view the p_j^1-values as projections of (p_j^1, p_j^2)-tuples onto the p_1-line, and the p_j^2-values as projections of (p_j^1, p_j^2)-tuples onto the p_2-line. The figure (left) sketches p-tuples in the 2D domain of F and their projections onto the p_1- and p_2-line.

The algebraic model for and the (optimal) computation of this model for $F(p_1, p_2)$ depend on the 'semantically correct' design of the relationships between $F_1(p_1)$, $F_2(p_2)$ and $F(p_1, p_2)$ AND the respective discrete sets $\{p_j^1\}$, $\{p_j^2\}$ and $\{(p_j^1, p_j^2)\}$.

It is therefore appropriate to recall the meaning of p-values and F-functions.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...

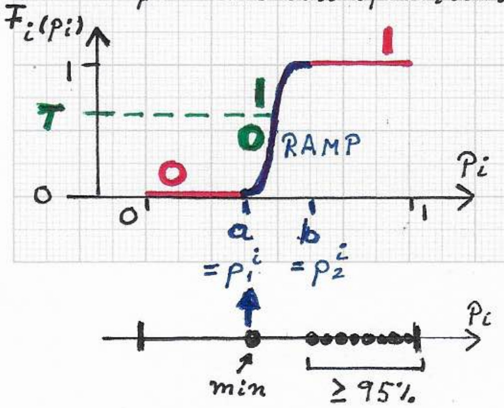
Example of a viable definition of a univariate decider function $F_i(p_i)$:

i) Considering all available p-values for scale i, define the value of a such that $[0, a]$ is the maximal interval in the p_i-domain not containing a single one of the given p-values.

ii) Analogously, define the value of b such that $[b, 1]$ is the minimal interval in the p_i-domain containing $\geq 95\%$ of the given p-values.

iii) Use the interval (a, b) to define a RAMP over it, e.g., using a simple Li-RAMP function or a more complicated proper Bernstein-Bézier polynomial.

iv) Via 'experiments' determine a value for a threshold T that optimizes classification.



This is the meaning of p-values:

• The value of p_i defines the degree of match, the degree of similarity of two image segments / materials at scale i. The value of p_i uses a measure appropriate for assessing the similarity of two coefficient value histograms for the two associated eigenfunction expansions, considering only scale i.

This is the purpose of decider functions F:

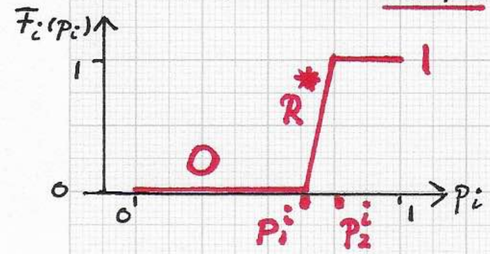
• The general purpose of a decider function F is 'binary decision-making': Using one p_i-value or multiple p_i-values as input, a decider function must fundamentally yield 0 (NO) or 1 (YES) as output result, indicating a class mis-match (0) or class match (1). Practically, F is designed with three regions: 0-region, 1-region and RAMP-region (from-0-to-1-region).

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

• Meaning of a decider function $F_i(p_i)$:



The function $F_i(p_i)$ is constructed with all the stored sample coefficient histogram data for scale i (of a specific class) and various not-stored scale-i histogram data for the same class.

The construction/training process employed to define $F_i(p_i)$ makes is possible to state this:

Given the coefficient histogram for scale i of a not-yet-classified image segment **S**, **S** is a match for a class at scale i WHEN THE SIMILARITY MEASURE p_i FOR THE TWO CLASSES BEING COMPARED SATISFIES $p_i \geq p_2^i$. Of coure, the value of p_2^i is different for different classes. In other words, when one considers **C** classes, one must use **C** decider functions $F_i(p_i)$.

* R = RAMP

We also should recall the meaning of p-values relative to all the data stored in the "sample database": Given a new, unclassified image segment, we compare it with all stored data, i.e., with all segments of all classes at all (eigenfunction expansion based) scales. Thus, we compute $p_{cl,sg,sc}$-values, where the triple index (cl, sg, sc) represent class, segment and scale indices. **The discussion on the previous pages was concerned with constructive methods primarily for univariate decider functions $F_i(p_i)$ for a specific material class and various scales i, $i = 1 \ldots H$. The principle employed was this: "The sample database" stores eigenfunction coefficient histogram data for several segments belonging to a specific class, at H scales.** TO GENERATE THE NEEDED DISTRIBUTION DATA FOR THE DEFINITION OF $F_i(p_i)$ FOR THIS **CLASS**, WE USE ADDITIONAL SEGMENT DATA OF THIS CLASS.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks: ...

- Univariate decider functions make possible the determination of "class match" or "class mismatch" using all available scales:

	scales 1	...	H
$class_1$	F_1^1	...	F_H^1
⋮	⋮	⋮	⋮
$class_C$	F_1^C	...	F_H^C

Table of all individual univariate decider functions

$F_i^{cl} = F_i^{cl}(p_i^{cl})$,

where cl is the class index and i is the scale index. Further, $F_i^{cl} = 0$ or $F_i^{cl} = 1$ or $F_i^{cl} = R_i^{cl}(p_i^{cl})$.

The information describing the observed and allowable VARIATION captured in the multiple image segments stored in the database is, at this point, represented inherently in the decider functions; the construction of F_i^{cl} uses exactly this VARIATION of segment behavior, e.g., texture.

THIS ADDITIONAL SEGMENT DATA IS COMPARED, AT ALL H SCALES, AGAINST THE STORED DATA FOR THIS CLASS IN THE DATABASE. *** SINCE ALL DATA BEING COMPARED BELONG TO THE SAME CLASS, THE CALCULATED SIMILARITY VALUES p_i (scale i) WILL BE "HIGH" – 'CLOSE TO 1 AND FAR FROM 0.' *** THEREFORE, THE RESULTING DISTRIBUTION OF VALUES ON THE p_i-LINE, IN THE INTERVAL [0,1], IS AN IDEAL BASIS FOR THE DEFINITION OF THE 0-, 1- AND RAMP-REGIONS FOR $F_i(p_i)$. THUS, IN OUR MULTI-CLASS CLASSIFICATION SETTING THESE DECIDER FUNCTIONS MUST BE ESTABLISHED FOR ALL CLASSES, AND ALL SCALES.

The table (left) summarizes all possible univariate decider functions $F_i^{cl}(p_i^{cl})$, $cl = 1 \dots C$, $i = 1 \dots H$. The model for and (near-)optimal computation of the overall decider function F is crucial for classification.

...