■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...

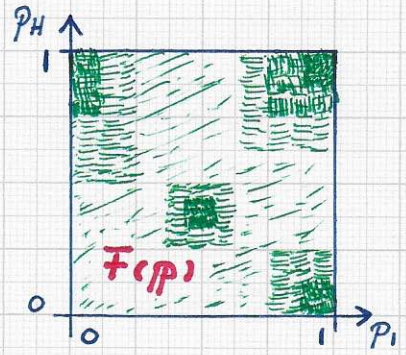


Illustration of the domain of the H-variate decider function $F = F(p_1, ..., p_H)$.

The input tuples $(p_1, ..., p_H)$ define points in the H-dimensional unit hypercube $[0,1]^H$ since $p_i \in [0,1]$.

One can assume that the function $F = F(\underline{p})$ returns a value in $[0,1]$ as well. Values of this function are shown via different degrees of shading.

A multi-layer neural network implementation of this function must be able to compute exactly the values of $F(\underline{p})$, by encoding the underlying "Boolean logical circuit" - defined either in binary-value logic or real-value-based fuzzy logic.

→ Recall: The value of p_i is the measure of similarity of segments S and s, at scale i, based on their eigenfunction coefficient value histograms for eigenfunction i (scale i).

The figure (left) shows the domain of the decider function $F = F(\underline{p})$ and indicates values generated by F. One can think of the definition of F in several different ways:

- One defines explicitly, in the language of Boolean and relational algebra, the conditions that the value(s) of $(p_1, ..., p_H$ and) $F(\underline{p})$ must satisfy for an unclassified segment S to belong to the class that segment s belongs to. (F computes a real value.)
- Employ a standard training method - "learning the function $F(\underline{p})$" - by using a large (but finite) number of classified training samples. (Weights and activation functions needed for a neural network are 'learned'.)
- Given a somehow established neural network implementation of $F(\underline{p})$, use a "reverse engineering" approach to devise an equivalent logical-gates-based definition of $F(\underline{p})$.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...

• Conceptual approach to the "organization" of conditions for p-value subsets used to arrive at a final decider function value $F(p)$:

→ There are H p-values defining the degree of match/similarity between S and s.

→ There are totally $2^H - 1$ possible combinations of p-values, defining p-value tuples with $1, 2, \ldots, H$ components.

→ Denoting the original complete p-value tuple as

$$p = (p_1, p_2, \ldots, p_H),$$

the set of 1-component tuples is

$$\{(p_1), (p_2), \ldots, (p_H)\}$$

of cardinality $\binom{H}{1} = H$;

the set of 2-component tuples is

$$\{(p_1, p_2), \ldots, (p_{H-1}, p_H)\}$$

of cardinality $\binom{H}{2}$; ...

and the set of H-component tuples is

$$\{(p_1, p_2, \ldots, p_H)\}$$

of cardinality $\binom{H}{H} = 1$.

...

• Considering the first of these three approaches, an expert would have to express a match of segments S and s, as far as class membership is concerned, in Boolean algebra via a precise definition, e.g., in the form

IF (a complete and complex condition involving $p_1, \ldots, p_H$ and arithmetic, relational and Boolean operators is TRUE)

THEN S and s belong to the same class

Since p_i-values are real numbers — and not binary truth values — all Boolean operators must be designed to generate a "probability for TRUE" or a "probability for FALSE". The final implication "belonging to the same class" also has an associated probability, i.e., the value of $F(p)$.

Keep in mind that a database of classified samples could contain about 10^4 samples; the approach of manually and explicitly defining match conditions is generally not a viable option.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

- Laplacian eigenfunctions and neural networks: ...

- "Organization" of p-value subsets and the decider function $F(p)$:

  → Based on the 2^H-1 p-value subsets, one can consider the use of a decision-making layer that applies conditions to these subsets:

  APPLY A CONDITION TO
  (p_1),
  (p_2),
  ⋮
  (p_H),
  (p_1,p_2),
  (p_1,p_3),
  ⋮
  (p_{H-1},p_H),
  ⋮
  $(p_1,p_2,\ldots,p_H)$,
  COMPUTE THE PARTIA RESULTS OF THESE CONDITIONS,
  APPLY ADDITIONAL CONDITIONS TO THESE RESULTS,
  COMPUTE ...
  ...
  COMPUTE THE FINAL RESULT $F(p)$.

- Before considering a training method, one should attempt to obtain some level of understanding of the "meaning" of the geometrical setting associated with p-tuples and the relationship to the decider function $F(p)$:

  i) The domain of allowable p-tuples is the H-dimensional unit hyper-cube $[0,1]\times\ldots\times[0,1]=[0,1]^H$.

  ii) The domain of the decider function is also $[0,1]$; this function F has a tuple $p=(p_1,\ldots,p_H)$ as argument and computes a value $F\in[0,1]$.

  iii) In an ideal, expert-controlled training scenario, an expert would specify correct, expected F-values for a large, but finite, set of tuples $(p_1,\ldots,p_H)$; an interpolation method could then be used – subject to satisfying certain conditions – to evaluate F for any p-tuple input.

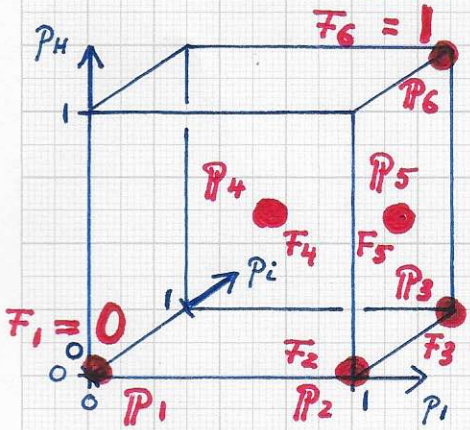
  IDEAL: Defining F analytically

  ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

• "Interpolation conditions" for the decider function $F(\mathbb{p})$:



• Locations / tuples for which decider function values are specified – "conditions" for $F(\mathbb{p})$:

$F_j = F(\mathbb{p}_j) = F(p_1^j, \ldots, p_H^j)$.

Here:

$F_1 = F(0,0,0) = 0$

$F_2 = F(1,0,0) = \ldots$

$F_3 = F(1,1,0) = \ldots$

$F_4 = F(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}) = \ldots$

$F_5 = F(1, \frac{1}{2}, \frac{1}{2}) = \ldots$

$F_6 = F(1,1,1) = 1$,

$0 \le F_j \le 1$.

⇒ Define:

$F(\mathbb{p})$, $0 \le F(\mathbb{p}) \le 1$ for $\mathbb{p} \in [0,1]^H$

These ways of viewing the needed definition of the decider function $F(\mathbb{p})$ can all be understood as attempts to define this function "as precisely as possible when a precise definition is impossible." In a mathematically informal way, one can describe the problem and desired solution in a more colloquial manner:

"We can compute the values of $p_1, \ldots, p_H$, with $0 \le p_i \le 1$, for a classified image segment S and an unclassified image segment s; the p_i-values represent the 'degree of similarity' at scales $1, \ldots, H$ of S and s. Given a tuple $\mathbb{p} = (p_1, \ldots, p_H) \in [0,1]^H$, the decider function $F(\mathbb{p})$ must have as its value a real number in $[0,1]$ that represents a probability of s being of the same material class as S when taking all H scales into account."

Thus, it is possible to understand the construction of $F(\mathbb{p})$ as an approximation problem. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

• Possible ways to think about the construction of a decider function $F(\mathbb{p})$:

- A p-tuple $(p_1, \ldots, p_H)$ describes the degree of similarity of two image segments S and s at scales 1...H. Here, similarity or match is purely defined via coefficient-value similarity of eigenfunction expansions.
- Only an application and an expert can ultimately define to what degree the segments S and s are to be considered a "class match." Thus, an expert must be involved in the design of $F(\mathbb{p})$.
- First: In a first step an expert could design individual scale-specific functions (univariate) that define a match for each p_i:

  $F_1(p_1), \ldots, F_H(p_H)$.

- Second: In a second step, one could "combine" the functions $F_1(p_1), \ldots, F_H(p_H)$ into the final decider function, i.e., $F = F(F_1(p_1), \ldots, F_H(p_H))$, using a linear combination or tensor product.

Understanding $F(\mathbb{p})$ as an interpolation function that interpolates a finite number of values

$$F_j = F(\mathbb{p}_j) = F(p_1^j, \ldots, p_H^j), \; j = 1 \ldots J,$$

makes it necessary to specify the values of F_j for specific tuples $\mathbb{p}_j$, choose a particular type of interpolation function F, and define / compute / "learn" the values of parameters of F in an "optimal" way. Two interpolation conditions for F are self-evident, they are:

i) $F(\mathbb{0}) = F(0, 0, \ldots, 0) = 0$.

ii) $F(\mathbb{1}) = F(1, 1, \ldots, 1) = 1$.

These conditions simply state that two material image segments S and s are NEVER of the same class when they have ZERO similarity values p_i for all H scales; and that S and s are ALWAYS of the same class when they have maximally possible similarity values $p_i = 1$, for all scales $i = 1 \ldots H$. The specification of all other F_j and $\mathbb{p}_j$ values is application-dependent and requires the knowledge of an expert.

...