OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

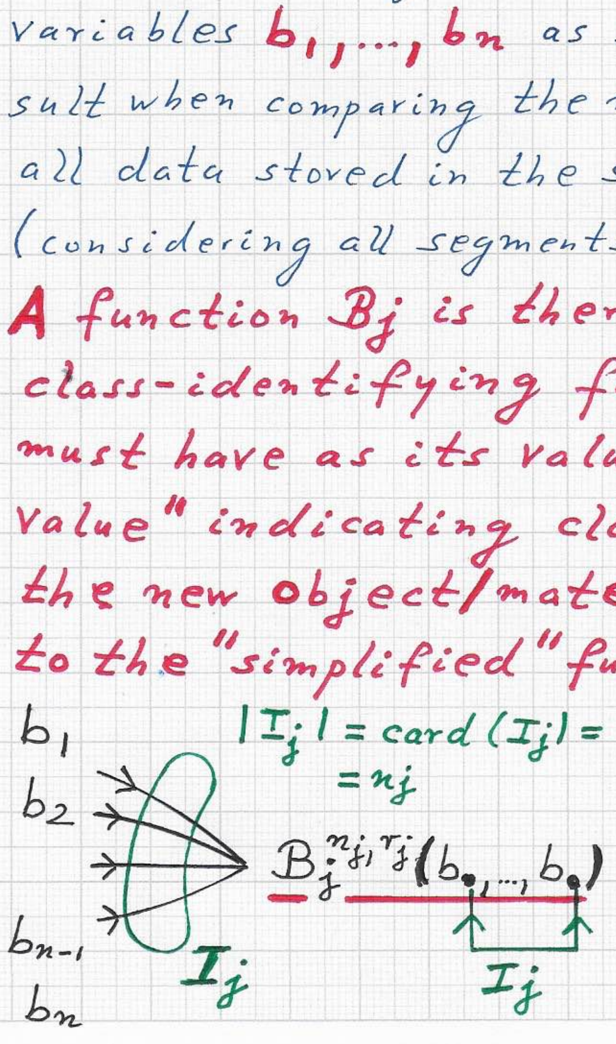
• Laplacian eigenfunctions and neural networks:… • **Note.** The goal is to determine whether a new, to-be-classified object/material belongs to **one** (or possibly more than one) of the classes to be recognized (or to "class 0"). At this stage of data processing and analysis, one can think of the values of the variables $b_1, \ldots, b_n$ as "degrees of similarity" — between the new object/material and the many class samples belonging to the classes to be recognized, represented at multiple scales via the calculated and stored Laplacian eigenfunction coefficient histograms. **Therefore, the classifying functions B_j (and $B_j^{n_j, r_j}$) use these "lower-level" comparison and similarity data $b_1, \ldots, b_n$ to calculate a "high-level" value defining the "optimal" measure of similarity between the new object/material and the classes to be recognized.** In other words, the "probability value" of the function $B_j^{n_j, r_j}$ must be "good enough" to make it possible to satisfy, for example, an overall classification requirement based on a performance metric using the numbers of TN, TP, FN and FP cases obtained in practical application.

…

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... We recall that a new, to-be-classified object/material is characterized via **histograms of the (absolute) values of the coefficients of a multi-scale representation by Laplacian eigenfunctions.** Therefore, the new object/material can be and is compared at every available **scale (sc)** with all available samples/**segment (sg)** for all **classes (cl)** to be recognized. It is possible to view the variables $b_1, \ldots, b_n$ as similarity values that result when comparing the new object/material with all data stored in the sample/segment database (considering all segments, all classes, all scales).

A function B_j is therefore a "higher-level" class-identifying function, $j = 1 \ldots N$, that must have as its value the "best-possible value" indicating class-j membership of the new object/material. B_j is "reduced" to the "simplified" function $B_j^{n_j, r_j}$ for efficiency.

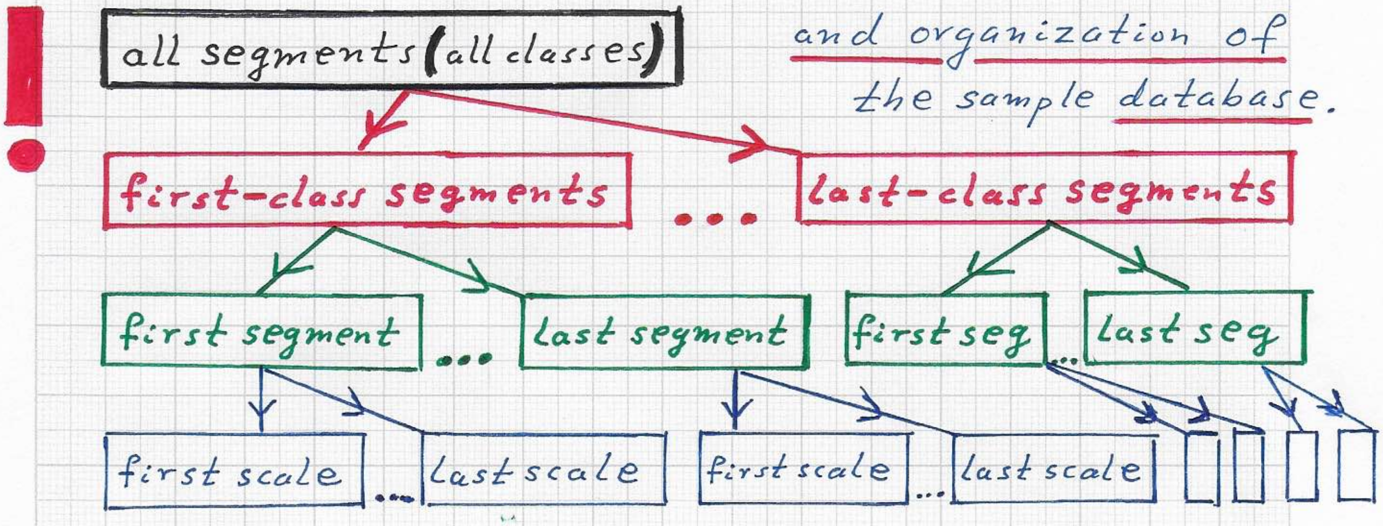


The left figure shows at a high **abstract** level how the independent similarity variables $b_1, \ldots, b_n$ are used to define a reduced function $B_j^{n_j, r_j}$ ...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

It is important to keep in mind the architecture and organization of the sample database.



The figure included here avoids a confusing, precise multi-index notation and focuses on the conceptual design of the sample database that stores the characterizations of all segments of all classes at all scales (based on Laplacian eigenfunction expansions) via coefficient-value histograms (at the scale level): The database stores all segments of all classes; segments are assigned to segment groups, from first-class to last-class; segment groups contain the segments for the groups, from first to last segment; and each segment is represented via its multiple "scale signatures" (from first to last), i.e., via scale-specific absolute-value coefficient histograms of Laplacian eigenfunction expansions.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks:… We consider a specific scenario to explain the "relationship" between the similarity values $b_1, \dots, b_n$ and the probability- and training-based values of the real-valued class-identifying functions $B_1, \dots, B_N$. For example, if a new, to-be-classified object/material has specific similarity values $b_1, \dots, b_n$ after comparing it at all scales and for all samples of all classes stored in the database, then its (optimized) values for $B_1, \dots, B_N$ should be $B_1 = .01$, $B_2 = .02, \dots, B_{17} = .92, \dots, B_N = B_{100} = .01$. The assignments of these B_j-values would be used as the conditions to be the constraints in the least-squares construction of the described best-approximating Haar wavelet expansions.
- **Note.** Computability, complexity theory and the universal (function) approximation theorem(s) are fundamental and crucially important areas in computer science, but they are not the foci of these high-level and application-driven notes. These areas have also increased in relevance in the context of artificial neural networks for the development of a sound theoretical basis concerning their capabilities and limitations. …

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks: ...

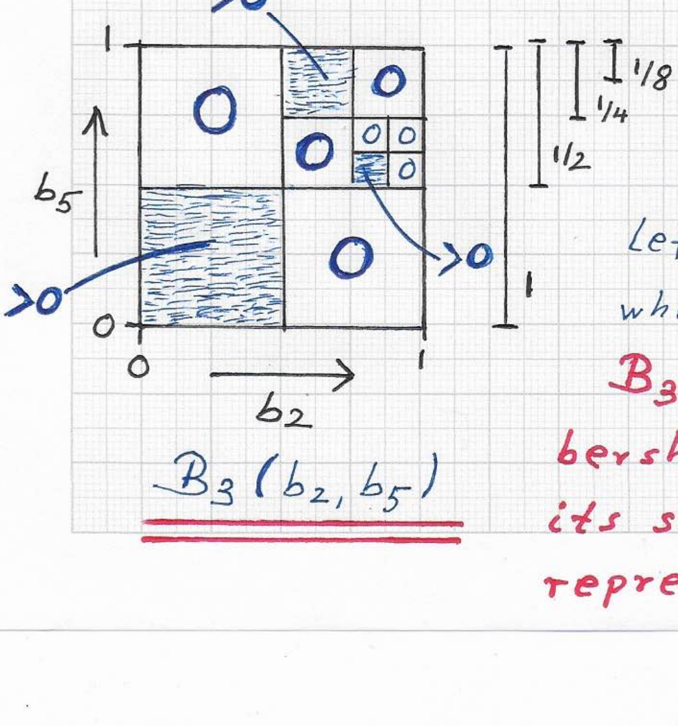
	b_1	...	b_n	B_1	...	B_N
1	b_1^1	...	b_n^1	B_1^1	...	B_N^1
...	$\vdots$		$\vdots$	$\vdots$		$\vdots$
S	b_1^S	...	b_n^S	B_1^S	...	B_N^S

$\Downarrow$ best approximation via least squares, using dyadic multi-resolution Haar wavelet tensor product expansion

$B_1(b_1,\ldots,b_n),\ldots,B_N(b_1,\ldots,b_n)$

$\Downarrow$ reduction of class-specific dimensions, selection of needed resolution levels

$B_1^{n_1,r_1}(\underbrace{b_\circ,\ldots,b_\circ}_{I_1}),\ldots,B_N^{n_N,r_N}(\underbrace{b_\circ,\ldots,b_\circ}_{I_N})$



$B_3(b_2,b_5)$

The table and illustrations on this page capture several of the relevant concepts used in the described method. The table shows the data tuples (rows) used for the construction of class-identifying functions B_j: "similarity tuples" $(b_1^k,\ldots,b_n^k)$ with dependent "class probability tuples $(B_1^k,\ldots,B_N^k)$. These data tuples are used as input for the least-squares computations of the functions $B_j^{n_j,r_j}$ (including dimension reduction and resolution selection). For example, the left figure shows a scenario where **the reduced function $B_3(b_2,b_5)$ returns class-3 membership value >0 in three of its shown three-level subdivision representation (levels 0,1,2,3)...**