OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks:... **Boolean algebra** could be viewed as a foundation, a building block or a "seed" for many possible generalizations to perform calculations relevant for the classification of materials and objects. Considering our classification problem, material/object classes are initially represented in a large sample database, used for "training." A class is represented by several physically viable samples, and each sample is statistically characterized by Laplacian eigenfunction coefficient histograms. A new, given, to-be-classified material/object must be compared to the database samples by comparing eigenfunction coefficient histograms, as described earlier in these notes.

Binary Boolean logic considers only truth values 0 (FALSE) and 1 (TRUE), providing the basis of the design of logic gates and circuits, using as elementary operators $\vee$ (OR), $\wedge$ (AND) and $\neg$ (NOT).

One can think of a **real Boolean-like function** as a function $B(b_1, \dots, b_n)$, where input variables b_i, $0 \le b_i \le 1$, $i = 1 \dots n$, determine one output variable, B, $0 \le B \le 1$.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

! • Laplacian eigenfunctions and neural networks:... **Probability** plays a crucial role in our classification setting. On a high level, one can think of the B-value as the probability defining the likelihood for a to-be-classified material/object to belong to the class for which the function B serves as the "class-identifying function." Further, the input variables $b_1, \ldots, b_n$ can also be understood as probabilities, based on, for example, certain sample- and scale-specific similarities of eigenfunction coefficient histograms.

! **Probabilities for multiple classes** would have to be modeled by multiple "class-identifying functions" $B_1(b_1, \ldots, b_n), \ldots, B_N(b_1, \ldots, b_n)$, where each real-valued function B_j, $j = 1 \ldots N$, defines a class membership probability value $0 \leq B_j \leq 1$ for the j^{th} class. In other words, B_j is a function that indicates whether a to-be-classified datum is likely to belong to class j or not. It is reasonable that the number of classes is generally small ($N < 100$), while the value of n (depending on the numbers of samples and scales used to represent a class) is very large ($n \gg N$).

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: … **Training** in this setting is used to optimize the value of an "appropriate" metric that characterizes the classification performance of the functions $B_1, \ldots, B_N$. The class samples in the large sample database can be used to perform this optimization — since exact class memberships are known for all stored and characterized samples. **Performance metrics** in our multi-class classification setting could, for example, be based on a unifying measure considering the numbers of true/false positives/negatives (TPs, TNs, FPs, FNs), as discussed in detail in these notes. In addition, one can and often must also consider memory/storage complexity and constraints, data transfer and network speed and constraints and time complexity and constraints. **Therefore, time, network and memory constraints could make it necessary to sacrifice "degree of classification correctness** (as measured by TP, TN, FP and FN numbers).

Best (optimal) approximation of functions is a method for calculating an error-minimizing function for a given discrete set of sample tuples of independent variable and associated dependent variable values. …

■ OBJECT AND MATERIAL EIGENFUNCTIONS — Cont'd.

- Laplacian eigenfunctions and neural networks:... By adopting the viewpoint of function approximation (or reconstruction), one can adapt well-known methods for best approximation to design approaches for our driving classification problem — where one must establish a multi-valued classification function $(B_1, \ldots, B_N)$ based on a given finite set of sample data, i.e., tuples $(b_1, \ldots, b_n)$ with the correct corresponding classification tuple $(B_1, \ldots, B_N)$.

! **Dimensions and resolutions** are crucially important aspects of the construction/reconstruction problem of a function $B_j(b_1, \ldots, b_n)$. For example, for $n=10$ and a resolution 10 (defining the number of equal-size sub-intervals used to discretize, to "bin," the interval $[0,1]$), the 10-dimensional domain of the 10-variate function B_j is represented by $10^{10} = 10\,000\,000\,000$ hypercubes. Thus, if one had 1000000 samples/data available for the construction of B_j, one would only have, on average, 1 sample per 10000 hypercubes. While this example is too simplistic and inadequate to discuss and define the construction of B_j precisely and properly, it demonstrates a fact:

! **The high dimensionality of B_j's domain is a "curse."** ...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... **Accuracy, certainty and error** are appropriate characteristics **one** should and must think of in terms of the "classification power" of a class-identifying function B_j. Using the potentially large value of n or large number of hyper-cubes used to discretize B_j's domain $[0,1]^n$ **is not** the proper approach for "understanding" the computational complexity of B_j's construction. In fact, one must explore and determine which subset of the totally available variables $b_1, \ldots, b_n$ — not necessarily being mutually independent — and how many of the available sample data suffice to construct the class-identifying functions $B_1, \ldots, B_N$ to ensure that a prescribed overall classification performance is satisfied.

Boolean circuit minimization, K-maps and combinatorial optimization — dimension reduction. In the context of simplifying/optimizing Boolean expression for minimal circuit design, one employs Karnaugh/Karnaugh-Veitch diagrams (K-maps) in low-dimensional cases. In high-dimensional cases, one can use combinatorial optimization methods to obtain near-optimal designs. **One should consider adaptation of such methods for designing $B_1, \ldots, B_N$.** ...