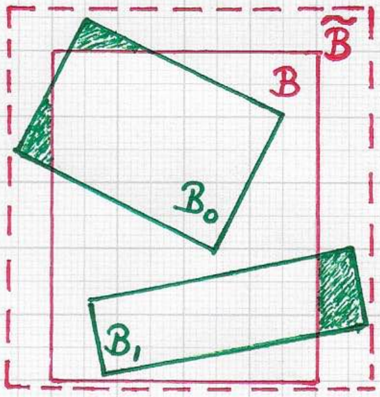
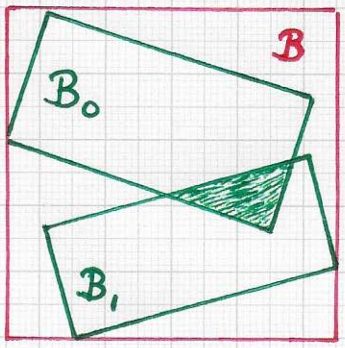
OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ...





Issues with splitting.

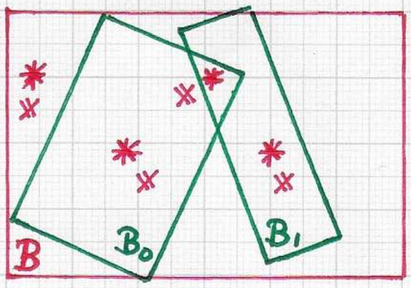
In summary, the described repeated-box-splitting method can, in principle, lead to two "geometrical" issues that one must keep in mind when designing and implementing the method via an effective algorithm. The two left figures illustrate these issues: (i) the splitting of box $\mathcal{B}$ into boxes $\mathcal{B}_0$ and $\mathcal{B}_1$ can introduce corner vertices of boxes $\mathcal{B}_0$ and $\mathcal{B}_1$ that lie outside $\mathcal{B}$, see top figure; concerning this issue, an approach for expanding $\mathcal{B}$ minimally to box $\widetilde{\mathcal{B}}$ has been discussed; (ii) the splitting of box $\mathcal{B}$ can produce child boxes $\mathcal{B}_0$ and $\mathcal{B}_1$ that intersect, see bottom figure; handling this intersection "problem" has not yet been explored; in addition, when expanding a box $\mathcal{B}$ to box $\widetilde{\mathcal{B}}$ the larger box $\widetilde{\mathcal{B}}$ might also lead to additional box-box intersections as a consequence of the expansion. Most likely, these "geometrical" issues arise rarely in a practical application and such box expansions and intersections are relatively "small." Nevertheless, a general implementation must consider them.

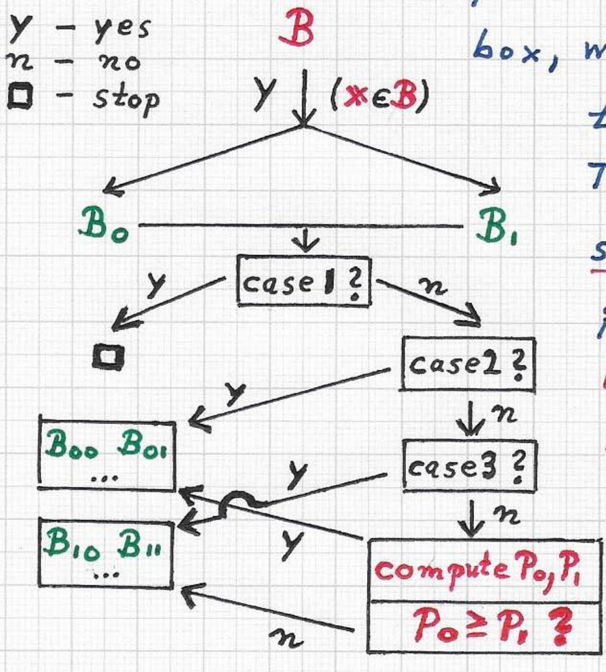
• • •

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

case	$* \in B_0$	$* \in B_1$
1	F	F
2	T	F
3	F	T
4	T	T





Determining unique tree traversal for possible cases arising from box splitting.

The truth table and the figures on this page concern issue (ii): handling the case of a box-box intersection scenario resulting from splitting a parent box B into child boxes B_0 and B_1. In the context of calculating the needed probability value P for a point $*$, four cases must be considered. The truth table includes these cases; 'T' (true) implies that $*$ is in the respective box, while 'F' (false) implies that the point is not in the box. The figure under the table shows these four location possibilities for a point $*$. If $*$ is inside one child box of B or inside both children of B (intersection area), one will have to determine ONE child box for the unique traversal of the tree hierarchy of boxes. The bottom-left figure shows a flow diagram for tree traversal.

•••

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks: ... Case 1 is simple, as * is not inside B_0 and not inside B_1; processing can be stopped, and the value of probability P is zero. Cases 2 and 3 are handled straightforward, as * is inside one child box only; if * is inside B_0 (B_1), then this child box will be considered for further splitting it into its children B_{00} and B_{01} (B_{10} and B_{11}). Thus, unique and unambiguous tree traversal is assured for cases 1-3. Concerning case 4, one must determine whether one should continue via the sub-tree of box B_0 or of box B_1. It is computationally not desirable to consider the traversal of the sub-trees of both B_0 and B_1. Keeping this desire in mind, one can handle case 4 as follows: If * lies in the intersection region of boxes B_0 and B_1, then it will be possible to compute class-membership probability values for *, i.e., P_0 and P_1. For this purpose, one would not calculate and consider the "tightness" T values of boxes B_0 and B_1; one would apply the described linear mapping directly to these two boxes, mapping them — and all points inside them, including * — to normalized u-space. ("Normalized u-space" refers to the cuboid $[-1,1]^D$ in D-dimensional space.)

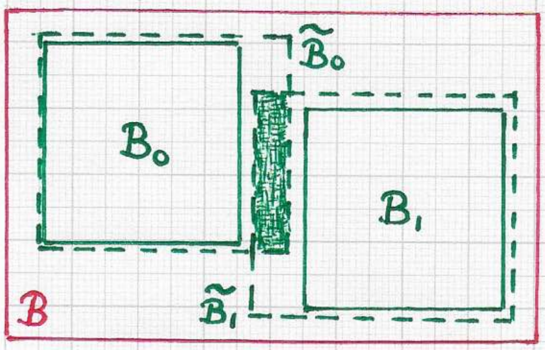
• • •

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:... Thus, one obtains two image locations/points for $*$, since the two boxes B_0 and B_1 each define a linear transformation for the mapping to $[-1,1]^D$. For these two image points, one can then calculate two probability values P_0 and P_1 — by evaluating the specific probability function defined for this class over the domain $[-1,1]^D$. It is important to keep in mind that the resulting P_0- and P_1-values only serve one purpose: If $P_0 \geq P_1$, the sub-tree of B_0 will be used for the continuation of tree traversal, i.e., the child boxes B_{00} and B_{01} will be considered next etc. If $P_1 > P_0$, the sub-tree of B_1 will be used for the continuation of tree traversal, i.e., the child boxes B_{10} and B_{11} will be considered next etc. The flow diagram shown in the bottom-left figure on p. 22 (7/29/2023) also describes this process. Therefore, the values of P_0 and P_1 are **NOT** used or interpreted as class-membership probabilities of $*$; these values simply ensure that the traversal of the box tree **NEVER** consider both subtrees of a box for the determination of a class-membership probability for a to-be-classified point $*$. One must also keep in mind that the geometrical expansion of a box B to a box $\widetilde{B}$ can introduce box-box intersections; this case is discussed next. •••

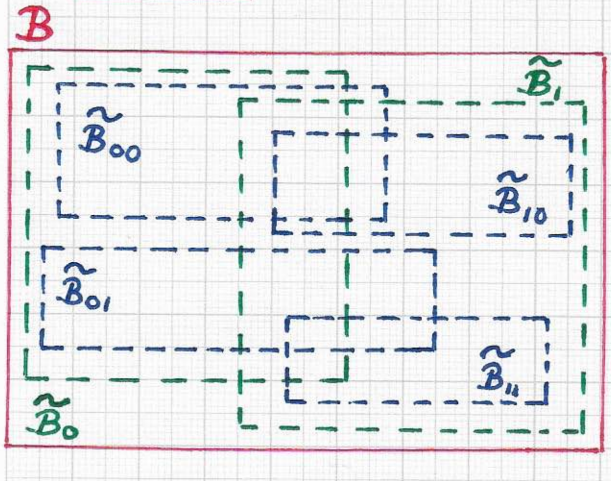
OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: …



Expansion of boxes B_0 and B_1 leading to new intersection.

Such additional box-box intersections can, in principle, be introduced by the tree construction process and box expansions that are a consequence of the described repeated box splitting method. The left figure sketches a very simple example. Here, the boxes B_0 and B_1 "initially" do not intersect. When applying repeated splitting to these two boxes (and possibly their children) it can become necessary to expand B_0 and/or B_1. In the illustrated example, both boxes are expanded; the resulting boxes are $\tilde{B}_0$ and $\tilde{B}_1$ – and their intersection is shown as a shaded area. Of course, it



Intersections of expanded boxes.

is theoretically possible that very complex box-box intersections are introduced, see left figure. Intuitively, such intersections of expanded boxes do not adversely affect a proper classification outcome for a point $*$. As pointed out, tree traversal for $*$ must not lead to the traversal of more than one sub-tree of a node. …