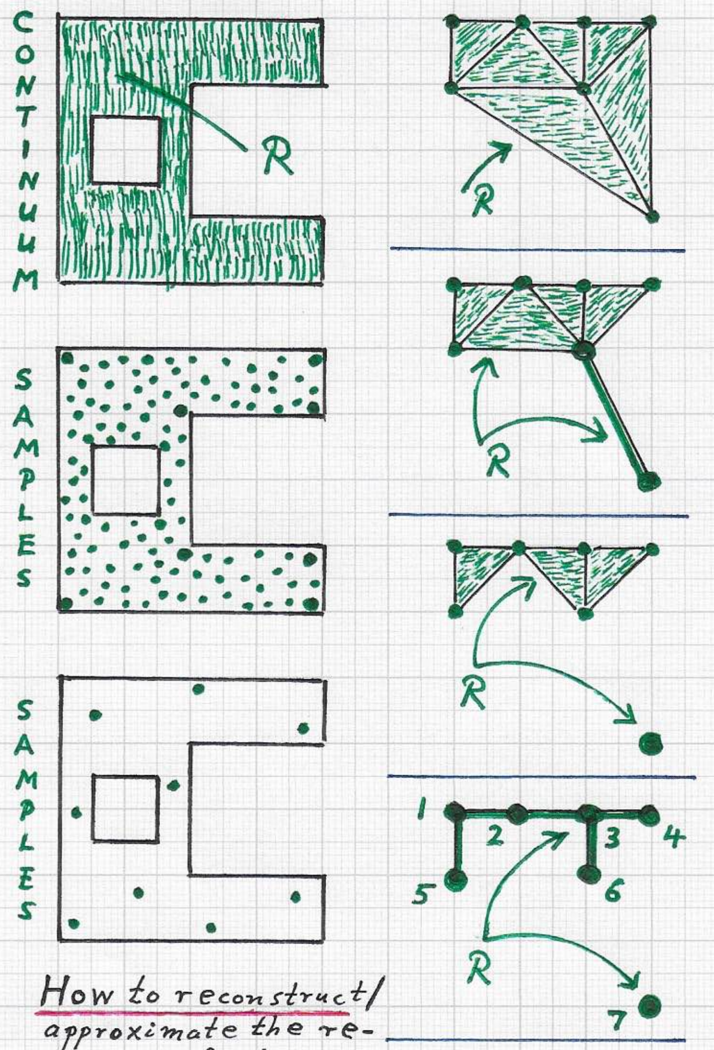
■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... Only using simplices/triangles as sub-regions is, of course, an option to represent a non-convex region R. We are interested in representations/approximations of a region R based on a "small number of 'optimized' elements" — keeping in mind the goal of achieving computational efficiency.



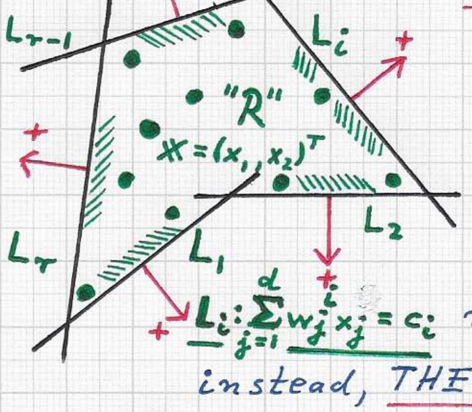
How to reconstruct/approximate the region in feature space from a finite set of feature points/vectors?

The left figure illustrates sampling a continuum region R by sets of feature points/vectors (left column) and the construction of sets of simplices, i.e., points (0D simplices), line segments (1D simplices) and triangles (2D simplices) as potentially viable **approximations** of the unknown continuum feature space region, from seven feature points/vectors. **One must prioritize efficiency**, regarding storage and computations!

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... While it is important, desirable and conceptually advantageous to consider the relationship between geometry and threshold logic-based classification, one must **emphasize** best-possible performance of a material classification system in system design. If one wants to include geometrical concepts in the design, they must be adapted in ways that support performance in terms of high TN and TP values; low FN and FP values; low computation times for classification decisions; low storage cost for data and data structures involved in classification decision-making etc. Thus, any adapted GEOMETRICAL APPROACH MUST OPTIMIZE THE CLASSIFICATION SYSTEM'S PERFORMANCE FUNCTION. The concept of using intersections and unions of half-spaces defined via linear polynomials is a viable choice for classification system design. Here, the goal is NOT the geometrical representation of some region "R"; instead, THE GOAL is to determine a minimal number of optimally "placed" linear decision boundaries L_i (used for Boolean decision-making) in a (feature) space of minimal dimension...



$L_i: \sum_{j=1}^{d} w_j^i x_j = c_i$

here: $d=2$, $r=5$

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... Depending on the specific context, we use the notation L_i to represent the linear polynomial $L_i = c_i + \sum_{j=1}^{d} w_j^i x_j$ or the Boolean variable defined as $L_i := \left(0 \geq c_i + \sum_{j=1}^{d} w_j^i x_j \right)$. Thus, the "linear Boolean constraints" L_i, $i = 1 \ldots r$, are applied to a d-dimensional space $(x_1, \ldots, x_d)$. For fixed values of d and r, the goal is to compute values for $c_i, w_1^i, \ldots, w_d^i$ such that the performance function of the classification system is optimized. Only a finite set of discrete point data in the $(x_1, \ldots, x_d)$-space, associated with available classified materials belonging to a class cl, can be used to define the r "linear Boolean constraints" L_i — and the final, overall Boolean logic expression using L_i variables (Boolean variables) as parameters to arrive at a decision for class cl (belonging / not belonging to class cl). One obtains a SYSTEM OF LINEAR INEQUALITIES:

AND

$$
\left[
\boxed{
\begin{array}{ll}
 & L_1 : w_1^1 x_1 + \ldots + w_d^1 x_d + c_1 \leq 0 \\
\wedge & L_2 : w_1^2 x_1 + \ldots + w_d^2 x_d + c_2 \leq 0 \\
 & \cdots \qquad \cdots \qquad \cdots \\
\wedge & L_r : w_1^r x_1 + \ldots + w_d^r x_d + c_r \leq 0
\end{array}
}
\right]
\;\hat{=}\;
\boxed{
\begin{bmatrix} w_1^1 & \ldots & w_d^1 \\ \vdots & & \vdots \\ w_1^r & \ldots & w_d^r \end{bmatrix}
\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}
\leq -
\begin{bmatrix} c_1 \\ \vdots \\ c_r \end{bmatrix}
}
$$

$\hat{=}$ $\underline{\underline{W \mathbf{x} \leq -\mathbf{c}}}$, Our goal is the computation of a matrix W and a vector $-\mathbf{c}$, from a large set of points $\mathbf{x}$, for (near-) optimal classification.

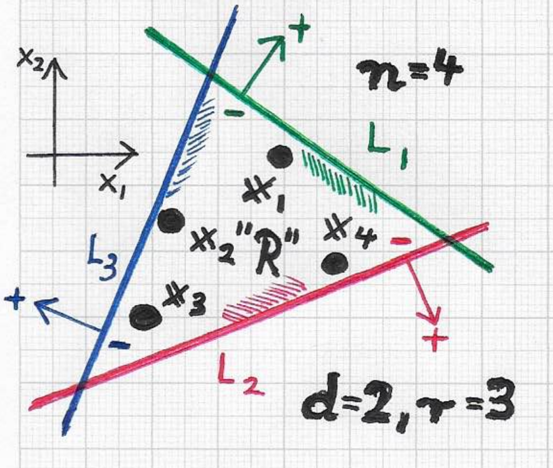
$\mathbf{x} = (x_1, \ldots, x_d)^T$ being one of many points.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

• **Note.** First, the system of linear inequalities presented on the previous page is the system as it applies to ONE point $\mathbf{x} = (x_1, \ldots, x_d)^T$. In fact, this system applies to the entire SET of points, called $\{\mathbf{x}_k = (x_1^k, \ldots, x_d^k)^T\}_{k=1}^{n}$ in the following. Second, a system of linear inequalities "combines" the individual inequalities in a "Logical-AND fashion" — considering the fact that it is the goal when attempting to find a solution(s) to calculate a solution(s) that satisfies (satisfy) all inequalities. The example sketched in the left figure clarifies the general case.



The three implicit line equations involved are:

$$L_1: \quad w_1^1 x_1 + w_2^1 x_2 + c_1 = 0$$
$$L_2: \quad w_1^2 x_1 + w_2^2 x_2 + c_2 = 0$$
$$L_3: \quad w_1^3 x_1 + w_2^3 x_2 + c_3 = 0$$

The figure indicates the outward (+) normals of all lines, and one can thus establish the system of linear inequalities that defines $\mathbf{x}_k$ as a point "left of L_i":

L_1-inequalities: $\quad w_1^1 x_1^k + w_2^1 x_2^k \le -c_1 \;, k=1\ldots4,$

L_2-inequalities: $\quad w_1^2 x_1^k + w_2^2 x_2^k \le -c_2 \;, k=1\ldots4,$

L_3-inequalities: $\quad w_1^3 x_1^k + w_2^3 x_2^k \le -c_3 \;, k=1\ldots4$.

Thus, we obtain a system with **12** inequalities.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- *Laplacian eigenfunctions and neural networks:...*

$$\begin{bmatrix} w_1^1\, x_1^1 + w_2^1\, x_2^1 \\ \vdots \\ w_1^1\, x_1^4 + w_2^1\, x_2^4 \\ w_1^2\, x_1^1 + w_2^2\, x_2^1 \\ \vdots \\ w_1^2\, x_1^4 + w_2^2\, x_2^4 \\ w_1^3\, x_1^1 + w_2^3\, x_2^1 \\ \vdots \\ w_1^3\, x_1^4 + w_2^3\, x_2^4 \end{bmatrix} \le - \begin{bmatrix} c_1 \\ \vdots \\ c_1 \\ c_2 \\ \vdots \\ c_2 \\ c_3 \\ \vdots \\ c_3 \end{bmatrix}$$

The complete system consists of 12 inequalities, for this example. The objective is the calculation of values for the line parameters $w_1^1, w_2^1, c_1, w_1^2, w_2^2, c_2$ and w_1^3, w_2^3, c_3. By re-writing the system (left, top) to an equivalent system (left, bottom), one obtains a representation that immediately shows that the system to be solved consists of 12 inequalities for 9 parameter values. This re-written system also makes obvious that a solution must satisfy the condition that all 4 points $\mathbf{x}_k$, $k=1\ldots4$, lie in the negative half-spaces (−) of all 3 lines L_i, $i=1\ldots3$. (In this case, one can say that the points $\mathbf{x}_k$ "lie in the negative half-spaces," "are on the left sides of all lines" or "are in the inside/interior of R.")

Re-writing the system:

$$\begin{bmatrix} x_1^1 & x_2^1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ & & & & \vdots & & & & \\ x_1^4 & x_2^4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1^1 & x_2^1 & 1 & 0 & 0 & 0 \\ & & & & \vdots & & & & \\ 0 & 0 & 0 & x_1^4 & x_2^4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1^1 & x_2^1 & 1 \\ & & & & \vdots & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & x_1^4 & x_2^4 & 1 \end{bmatrix} \begin{bmatrix} w_1^1 \\ w_2^1 \\ c_1 \\ w_1^2 \\ w_2^2 \\ c_2 \\ w_1^3 \\ w_2^3 \\ c_3 \end{bmatrix} \le \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mathbb{X} \cdot \mathbb{W} \le \mathbb{0}$$

...