■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... For the design and implementation of an optimization algorithm one must consider several closely related questions and issues. These are just some of the important ones:

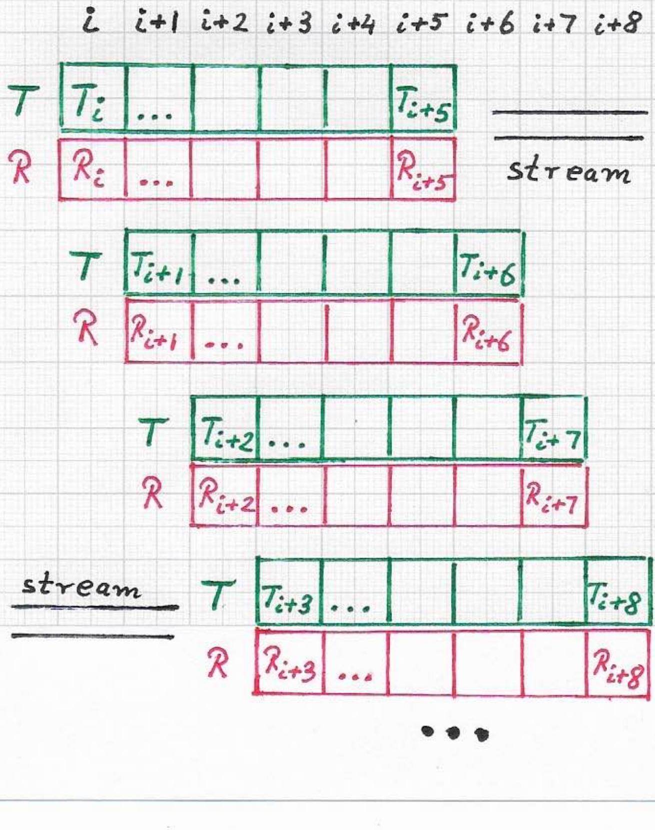
- Many performance measures/metrics exist to characterize the performance of a multi-class classification method/system. We must determine which ones are appropriate for our application. For this purpose, we briefly review some standard measures/metrics commonly used.
- In the context of our application, a combinatorial optimization approach is appropriate. Since such an approach can generally only consider a subset of all combinatorially possible parameters and parameter values, one must select a computationally viable approach. For example, simulated annealing is an approach that can be considered — complexity-wise.
- Ideally, a classification system is based on a software architecture that supports "continually ongoing adjustments and improvements" of the classification optimization process. In other words, the system should at all times monitor its performance, and optimization should be a permanent process to improve classification results. ...

Bernd Hamann
Stratovan

11/14/2022

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... We first sketch ideas that one can use to continually update and adjust material classification parameter values, with the goal of optimizing system performance all the time, in a quasi real-time learning process. **!** To make this possible, one can view the materials as a "stream of to-be-classified data." Such a streaming model allows one, for example, to assess classification correctness and performance for data stream sequences that have been processed and classified – and adjust classification parameter values, with optimal system performance being the goal. The following illustration sketches the basic idea of such a self-adapting learning system:



... material data stream ...

T : array of 6 subsequent TRUE material labels

R : array of (corresponding) 6 subsequent material labels RESULTING from classification

In our setting, material labels are integer labels in $\{0, 1, \ldots, C\}$. A classification of a material i is correct $\Leftrightarrow T_i = R_i$. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... For example, the system can monitor its classification performance in the following way: When the next material segment of true, correct class cl_{true} is classified by the system, the class label resulting is cl_{result}. Based on the class label generated by the system, the corresponding performance counter(s) is(are) incremented, i.e., the value of #TN, #FN, #TP, #FP, $\#TP_1, \ldots, \#TP_C$, $\#FP_1, \ldots,$ or $\#FP_C$. Further, to assess system performance for the most recently streamed and classified materials, one must base the values of these counters on only the most recent 100 or 1000 or 10000 materials that have been processed. The specific counter values for these most recent materials can now be used to COMPUTE THE SYSTEM'S "CURRENT" OVERALL PERFORMANCE VALUE. By computing and storing the temporal sequence of such subsequent overall performance values ~~one sees that~~ (i) system performance is (nearly) stable; (ii) system performance is improving; or (iii) SYSTEM PERFORMANCE IS DETERIORATING. Case (iii) warrants adjustment of classification parameter values via the chosen combinatorial optimization method. IN FACT, ONE CAN AND SHOULD USE ALL THE AVAILABLE DATA FOR THE "DETERIORATION PERIOD" FOR OPTIMIZATION.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... On a very high level of characterization, classification system performance is defined by the numbers of TRUE and FALSE classification outcomes. Further, one can summarize some of the most important goals for system behavior influencing the optimization process as follows:

1) TRUE classifications should be maximized.
2) FALSE classifications should be minimized.
3) In the context of threat recognition, an FN is "unacceptable" – an FP is "tolerable."
4) Ultimately, the goal is to recognize all threats as threats, i.e., TP = number of really existing threats – and accepting that a "small" number of FPs result from classification.
5) (See 4)) Minimizing the number of FPs at the expense of not recognizing really existing threats is not a "viable" goal.
6) When a real threat material from one of the classes $1, \ldots, C$ must be recognized, it is of primary interest to recognize the material as a threat; it is only of secondary interest to recognize the exact class index of the threat material.
7) Based on principles 1) – 6) – and additional ones – one can establish scores for the outcomes TP, FP, TN, FN, TP_1, FP_1, ..., TP_C, FP_C.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... Keeping the guiding principles from the previous page in mind, one can devise a scoring scheme for the classification system's performance. As suggested before, performance could be assessed by considering the classification outcomes for the last **K** classified materials. The correct, TRUE, class membership string is written as $(T_1, \ldots, T_K)$, and the corresponding classification string is written as result string $(R_1, \ldots, R_K)$. When comparing R_i with the correct class T_i, one determines the classification type: TP, FP, TN, FN, ..., TP_c or FP_c. Further, one can define a principle-based score s_i for each of these classification types – and sum up these scores for all **K** classification outcomes. For clarification, we consider an example:

- We are concerned with classes 0 (=non-threat), 1 and 2.
- We consider classification performance for $K = 3$.
- We associate the following scores with outcomes: $TP(1)$, $FP(-1)$, $TN(1)$, $FN(-1)$, $TP_1(\frac{1}{2})$, $FP_1(-\frac{1}{2})$, $TP_2(\frac{1}{2})$, $FP_2(-\frac{1}{2})$.
- The correct, TRUE class string is $(0, 1, 2)$.
- There are $C^K = 3^3 = 27$ possible result strings, i.e., $(0,0,0)$, $(0,0,1)$, $(0,0,2)$, $(0,1,0)$, ..., $(2,2,1)$, $(2,2,2)$.
- We can compute the overall score of each result string: $(0,0,0)$: $1+(-1)+(-1) = \underline{-1}$, $(0,0,1)$: $1+(-1)+(1+(-\frac{1}{2})) = \underline{\frac{1}{2}}$, ..., $(0,1,2)$: $1+1+\frac{1}{2}+1+\frac{1}{2} = \underline{4}$, ..., $(2,2,2)$: $-1-\frac{1}{2}+1-\frac{1}{2}+1+\frac{1}{2} = \underline{\frac{1}{2}}$.

...