Visualization of Confusion Matrix for Non-Expert Users

Emma Beauxis-Aussalet* Lynda Hardman†

CWI - Information Access Group

ABSTRACT

Machine Learning techniques can automatically extract information from a variety of multimedia sources, e.g., image, text, sound, video. But it produces imperfect results since the multimedia content can be misinterpreted. Machine Learning errors are commonly measured using confusion matrices. They encode type I and II errors for each class of information to extract. Non-expert users encounter difficulties in understanding and using confusion matrices. They need to be read both column- and row-wise, which is tedious and error prone, and their technical concepts need explanations. Further, the visualizations commonly used by Machine Learning experts make use of complex metrics derived from confusion matrices (e.g., Precision/Recall, F1 scores). These can be overwhelming and misleading for non-experts. Derived metrics convey specific types of errors, and may be inappropriate for specific use cases. For instance, type II errors (False Negative) are critical for medical diagnosis while type I errors (False Positive) are more tolerated. In the case of optical sorting of manufactured products (defect detection), the sensitivity to errors can be the opposite. Non-experts may use inappropriate metrics for their use case, or misinterpret them. We propose a novel visualization design that addresses such issues with non-experts users. We specify the potential misinterpretations that can arise in typical use cases of machine learning applications. We argue that our visualization is likely to be easier to understand and to minimize the risk of misinterpretation, and so for all kind of use cases. We conclude by discussing future empirical evaluations of our design.

Index Terms: H.5.2 [Information Interface and Presentation (e.g., HCI)]: Prototyping— [I.2.1]: Artificial Intelligence—Applications and Expert Systems

1 USE CASES

Confusion matrices are the major mean to evaluate errors in classification problem (the sorting of items into classes, i.e., categories or kinds of items). Supervised machine learning is a typical application for confusion matrices. They encode the complete specification of misclassifications: the numbers of misclassified items for each pair {original class in which items should be classified, incorrect class in which items are erroneously classified}. Items are known to belong to an original class from a trusted set of pre-classified items (a *ground-truth*). Confusion matrices are used for: i) inspecting errors for each class (e.g., Fig. 1); ii) tuning software parameters such as detection thresholds (e.g., each set of parameter values yield a matrix); iii) comparing software versions (e.g., each software yield a matrix). For binary classification (i.e., items are either selected or discarded), the selection of software version, or parameter settings, basically rely on the tolerance to unclassified, missed items (False Negatives *FN* or *Type II* errors), and classified but deceiving items (False Positives *FP* or *Type I* errors). For multiclass problems (i..e, items are classified into several categories), type I and II errors are appraised for each class. The sensitivity to either error type depends on application domains. In some domains, type I (FP) are critical while type II (FN) are more tolerated: e.g., fraud detection involving automatic suspension of services (bank, mail, social media), biometric identification, recommendation, optical sorting (referred here as *Case A*). In other domains, type II (FN) are critical while type I (FP) are more tolerated: e.g., medical diagnosis, threat detection (*Case B*). Others are sensitive to both error types: e.g., character recognition, monitoring of population dynamics, ecology research (*Case C*).

*e-mail: emalb@cwi.nl

†e-mail: lynda.hardman@cwi.nl

2 ISSUES WITH CONFUSION MATRICES

Analyzing misclassifications is complicated since FN for one class are FP for another. E.g. in Fig. 1, the cell with orange background indicates both 9 FN missed for Barracuda classification, and 9 FP added for Clown Fish classification. To fully understand the errors impacting one class, users need to inspect the matrix both column-wise (e.g., to inspect FN in Fig. 1) and row-wise (e.g., to inspect FP in Fig. 1). We consider that memorizing all cell values, and their semantic, is a major issue: users can forget cell values, or may read only columns or rows. Confusion matrices are usually synthesized by cumulating misclassifications for each class, thus reducing the number of data cells to read. E.g. in Fig. 1, the orange cell is counted in both blue cells. Such basic metrics are equivalent to considering each class as a binary classification problem. Users can no longer distinguish which classes are likely to be confused with another. This is an issue in domains analyzing trends over time (e.g., population dynamics). For instance, an important increase of one class implies an increase of its FN, and can thus induce a deceiving increase in other classes. Confusion matrices are usually synthesized further by deriving advanced metrics from the basic metrics: basically rates of correct and incorrect classifications over total numbers of items to detect or discard. Fig. 1 shows widely used metrics and their formulas. Advanced metrics are complicated for non-experts. They may not know which metrics suit their use case, or misinterpret them. E.g., high TN may conceal critical errors by yielding low *FP Rate* and high *Accuracy*. *Precision* does not convey the errors critical for *Case A*, nor *Recall* and *FP Rate* for *Case B*, nor *Accuracy* and *F1 score* convey the errors critical for neither *Case A and B*. For *Case C*, using only one metric amongst *Precision, Recall* and *FP Rate* does not convey sufficient information.

Machine Learning experts usually visualize confusion matrices by plotting together pairs of advanced metrics, e.g., TP and FP Rates (ROC curves) or Precision/Recall. Fig. 2 shows such visualization for 2 software (with different ground-truth size), and for different parameter setting (the points A to D represent 4 detection thresholds). The risk to overwhelm and confuse users is amplified with pairs of metrics. For selecting a threshold, non-experts may not identify the point offering the best tradeoff between type I and II errors: it is case-depends and there is not one universal best point on the curves. For comparing errors between classes, curves can be drawn for each class, giving errors for each parameter settings while only one setting will be used. It adds unnecessary information, which saturates user memory and is error-prone. Finally, classifiers' errors can appear identical in one type of curve, while another view would reveal further differences (Fig. 2).

		Classification from Ground-Truth				Basic Metrics				Advanced Metrics					
		Anchovy	Barracuda	Clown Fish	Other	TP	FP	FN	TN	Precision TP/(TP+FP)	Recall or TP Rate TP/(TP+FN)	FP Rate FP/(FP+TN)	Accuracy (TP+TN)/All	F1 score 2TP/(2TP+FP+FN)	Matthews Cor. Coefficient
Classification from the Software	Anchovy	85	12	0	15	85	27	15	273	0.76	0.85	0.09	0.90	0.80	0.73
	Barracuda	12	75	4	9	75	25	25	275	0.75	0.75	0.08	0.88	0.75	0.67
	Clown Fish	0	9	95	6	95	15	5	285	0.86	0.95	0.05	0.95	0.90	0.87
	Other	3	4	1	70	70	8	30	292	0.90	0.70	0.03	0.91	0.79	0.74

Figure 1: Confusion matrix (left) and derived metrics

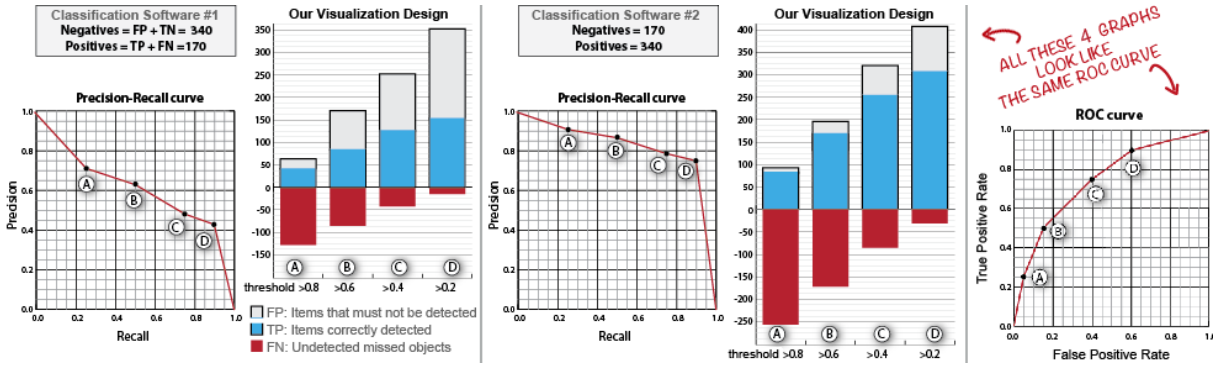

Figure 2: Alternative visualizations of confusion matrices: our design, and equivalent ROC and Precision/Recall curves.

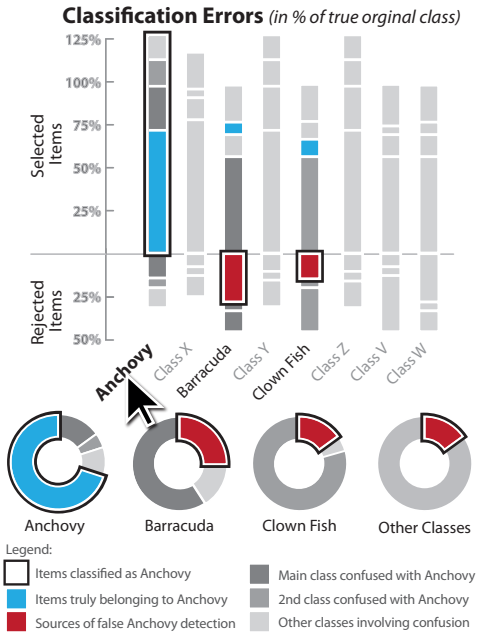

Figure 3: Visualization for analyzing of inter-classes confusions.

3 Visualization Design

To ease non-experts' interpretation, we display only True Positives *TP* (correct classifications), FN and FP (Fig. 2). True Negatives *TN* (items correctly discarded) are omitted since they are potentially uninteresting (e.g., not contained in end-results) and misleading (e.g., high TN yield high *Accuracy*). We primarily show raw numbers of errors (e.g., in Fig. 2) which are more tangible, without rates and advanced metrics, which require higher abstraction level from users. It preserves some information hidden in ROC and Precision/Recall curves: the numbers of item in the ground-truth. It displays both type I and II errors, without attempting to show their proportion relatively to case-dependent frames of reference. Hence it suits most use cases, whereas inappropriate expert visualizations (e.g., ROC curve) may conceal relevant errors.

Classes can have heterogeneous numbers of ground-truth items, thus being difficult to compare. Hence our design in Fig. 3 reports errors proportionally to the TP of each class. This visualization design details further the inter-classes confusions. It indicates the 2 classes producing the most FP and FN, as well as the average errors for all other classes. Highlighting the classes likely to be confused with another indicates potential biases in end-results (e.g., high confusion between classes yielding correlated but unrepresentative trends in end-results). Such information are lost in the usual basic metrics, since FN and FP are cumulated over classes.

4 Future Design Evaluation

Initial user study shows that basic and advanced metrics (Fig. 1) are likely to be overwhelming for non-experts [1]. It suggests that expert visualizations (ROC and Precision/Recall curves) are not recommendable for non-experts. To empirically validate our design, we will compare user responses when using either our visualization, or the usual ROC and Precision/Recall curves. We will investigate i) user effectiveness and efficiency in selecting case-depend optimal parameter settings and software versions (including ground-truth quantity); and ii) user level of confidence in their interpretation of end-results, and its consistency with the actual uncertainty. Experiments possibly include experts and non-experts, to compare visualization requirements for each audience. An application of our design, and its integration in a data analysis interface, can already be demonstrated (prior demo [2, 3]).

References

[1] E. Beauxis-Aussalet, E. Arslanova, L. Hardman, and J. van Ossenbruggen. A case study of trust issues in scientific video collections. In *Proceedings of the 2nd ACM international workshop on Multimedia analysis for ecological data*. ACM, 2013.

[2] E. Beauxis-Aussalet and L. Hardman. Interactive visualization of video data for fish population monitoring. In *Creating the Difference: Proceedings of the CHI Sparks Conference*, 2014.

[3] E. Beauxis-Aussalet and L. Hardman. Uncertainty-aware visualization of fish populations. In *Demo at the International Working Conference on Advanced Visual Interfaces*. ACM, 2014.