Relative Confusion Matrix: Efficient Comparison of Decision Models

Luc-Etienne Pommé, Romain Bourqui, Romain Giot, David Auber

▶ **To cite this version:**

Luc-Etienne Pommé, Romain Bourqui, Romain Giot, David Auber. Relative Confusion Matrix: Efficient Comparison of Decision Models. IEEE, pp.98-103, 10.1109/IV56949.2022.00025 . hal-03706436

HAL Id: hal-03706436
https://hal.science/hal-03706436v1

Submitted on 27 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Relative Confusion Matrix: Efficient Comparison of Decision Models

Luc-Etienne Pommé, Romain Bourqui, Romain Giot and David Auber
Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR5800, F-33400 Talence, France
luc.pomme-cassierou@u-bordeaux.fr

***Abstract*—Current machine learning and deep learning approaches are cutting-edge methods for solving classification tasks. Comparing the performances of classification models has become a prominent task since the outbreak of these techniques. The performance of such classification models is measured by the ratio between the correctly predicted samples and the others. The most widely used visualization to represent this information is the Confusion matrix. Yet, if this technique is suited to apprehend one model performances, very few works use this representation to compare models. In that paper, we present the Relative Confusion Matrix (RCM), a new matrix visualization that leverages Confusion matrices and a color encoding to expose the class-wise differences of performances between two models. We conduct a user evaluation to compare RCM with two confusion matrix variants. Our results show that RCM encoding leads to a more efficient comparison of two models than existing approaches.**

***Index Terms*—Confusion Matrix, Models comparison, Evaluation**

Introduction

Machine Learning (ML) and Deep Learning (DL) algorithms are widely used to solve detection [14], segmentation [15] or classification [16] problems. In the literature, instances of theses algorithms are called models. Prior to be used, these models must be trained to setup their hyperparameters. Then these models can be benchmarked to measure their accuracy and behavior. The focus of that paper is to present a comparison and a user evaluation of different visualization techniques which enable to compare results obtained by two different trained models on the same classification task.

A classification task consists in assigning to an input data (for instance, an image) a label called *class* (for instance "cat") from a bounded dictionary (ie. alphabet). One calls a *prediction* the output class. To benchmark models, one computes predictions on an input dataset for which classes are already known (*ground truth*) and then predictions are compared to ground truths.

The performance of a model can be evaluated at different levels. (1) At a global-level, measures such as the accuracy (good predictions / number of samples) provide a score for the entire test set. (2) At a class-level, measures are used to assess the performance of the model through the prism of a specific class. For instance, the recall of a class indicates whether the samples of that class were correctly classified.

A lot of works [2], [6], [7], [13] leverage class-level measures for models analysis and rely on a visualisation called *confusion matrix* (CM) [8].

A CM is a matrix where the rows represent the ground truth of input samples and the columns represent the predictions of a model, in the same order. The cell (row, col) counts the percentage of samples of ground truth row predicted as class col. Standard representation of CM fill each cell using a gradient of colors mapped to the percentages. Some variants display the percentages in each cell. A dark color is tied to a high percentage while a light color is tied to a low percentage.

Confusion matrices were designed to compare the class performance within a model. However, when designing a new model, a common approach consists in training several model and then selecting the best one. In such a case, confusion matrices may not be sufficient to efficiently compare the performance of two (or more) models. While the accuracy can provide some cues about the overall behaviors of the compared models, it does not support the comparison of models at a class level. In particular, when comparing two models it is important to assess (1) whether one of the models achieves better performance for all classes or (2) one model is better for a set of classes but worst for another one. Identifying these classes can also be important to the application domain; some classes may be more crucial than others.

In this work, we study different strategies to compare two models that solve the same classification task. The contribution of this paper is twofold. First, we present the Relative Confusion Matrix (RCM) which allows a class-level performance comparison between two models. Last, we conducted a user evaluation to compare RCM with two existing visualization methods to compare two models at a class-level.

In the next section, we provide a brief state of the art of model comparison techniques. Then, we present RCM and how it differs from other visualizations. Then, we describe the user evaluation protocol and the results of the evaluation. Finally, we discuss the results and draw conclusions.

Related Work

In this paper, we are interested in the visual comparison of two models performances. This section discusses about existing techniques for that purpose.

To visually compare quantitative information, it is well established that position-based [4] visualizations like bar charts are the most efficient approaches. They are particularly adapted to compare one or two variable over a few models [3], for example to compare a user response time or an accuracy error over several views during a user evaluation. For binary

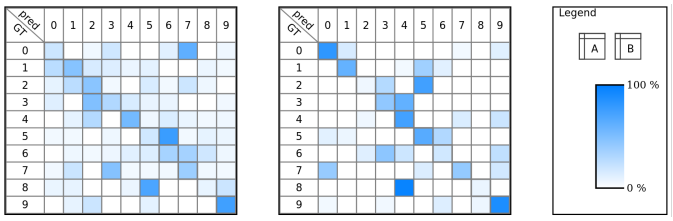


Fig. 1: Example of Twin Confusion Matrix (TCM) representation

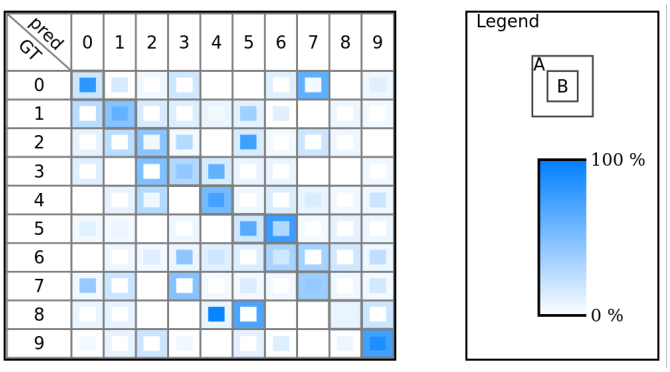


Fig. 2: Example of Combined Confusion Matrix (CCM) representation.

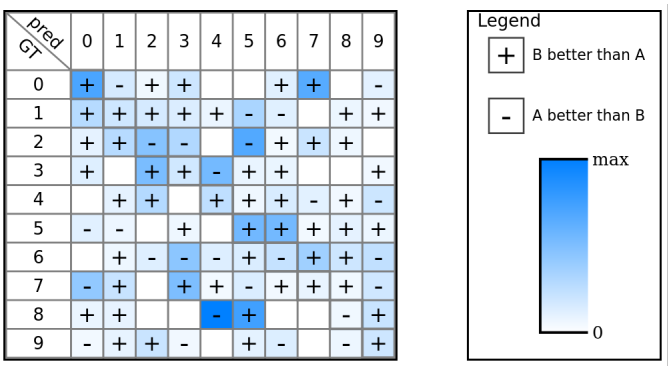


Fig. 3: Example of Relative Confusion Matrix (RCM) representation.

classification models, the Receiver Operating Characteristic (ROC) [12] curve in a form of a line chart can be drawn. The curve gives information about how much the model benefits, making correct predictions for a class, compared to how much it costs, incorrectly classifying samples from another class into that class. However, it is less suitable for problems with more than two classes due to overlaps. In Squares [13], Ren *et al.* compare multiple classification models in rows using an horizontal histogram per class to show the distribution of samples in each class.

Alsallakh *et al.* [7] spread the classes along a confusion wheel to visualize more information about the content of the classes. In particular, for each class, they use colors to differentiate common performance measures like true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN) in histograms of classification probabilities. In this representation, each class along the wheel requires enough space to display a curved histogram. This may be problematic when comparing two spatially distant classes, especially when the number of classes increases.

Most of the time, to compare the performance of a model at a class-level, a confusion matrix is used [6], [10]. CM have the advantage of being model-agnostic. Instead of filling the cells of a CM using a gradient of colors, NEO [9] encodes the percentages in the cells with rectangles of different sizes.

As Hinterreiter et al. [2] stated, very few works address the problem of comparing two models at a class level. In general, this comparison is implicit. For example, Li et al. [5] display one CM per neural network obtained through iterations of pruning. The CM of two distinct models are not displayed at a time, which may require memory attention from the user to compare both models.

For comparison problems in general, Gleicher [1] suggested three ways to design components. The first and most common way to compare objects is the juxtaposition of instances of the same visualization. Talbot et al. [6] compared multiple classifiers by juxtaposing confusion matrices. The second way is the superimposition of two instances of the same visualization. Basak et al. [11] make the parallel between superimposed directed weighted graphs and superimposed matrices. In each cell of their CM, they display two colored concentric rectangles to encode the result of two models on that cell. The third way uses a special encoding to combine the information of the two objects. In this last category, ConfusionFlow [2] suggest to compare multiple confusion matrices in a single visualization using lines in each cell. The points of the lines in each cell estimate the variation of the percentage of samples in that cell for all models. This representation allows a global comparison of several models at a class-level.

Design Choices

In this section, we present the three visualizations we decided to compare in our user evaluation. First, we describe our contribution: the **Relative Confusion Matrix (RCM)**. This matrix is designed specifically to compare two models that solve the same classification task, and more precisely, to show where a model B improves or deteriorates the performance of a model A. From now on, we consider both model process the same input data and have the same number of output classes.

The design of the RCM is very similar to the design of standard confusion matrices: it is a square table where rows represent all possible ground truths and columns represent all possible predictions, sorted with the same order.

Two different models may have different percentages in every cell. The goal of RCM is to highlight these differences between two models A and B. Each cell of RCM contains the cell-by-cell absolute percentages difference between B and A. To ease the interpretation of RCM, a gradient of colors is used to fill in the cells. A dark color is used when the difference is high. A light color is used when the difference is small. In

addition, if there is some difference between the two models on a cell, a + (resp. -) is displayed on that cell to indicate that B is better than A (resp. A better than B). A blank cell indicates there is no difference between A and B on that cell.

We chose $+$ and $-$ to indicate which model is the best on a cell. It is particularly interesting in a context of B being a variant of A. Then, the symbol answers the question of B improving or deteriorating the performance of the cell compared to A (Fig. 3). A or B could have been a reasonable choice but these letters are written with more pixels than simple $+$ and $-$, and they contain loops. Especially when the number of classes is high, the color perception of the cells could be altered with letters. Moreover, we did not use more than one gradient of colors (e.g to differentiate diagonal cells from the other cells or to differentiate cells showing improvements from cells showing deterioration) for RCM to avoid comparisons between the lightness of two or more different colors.

In our evaluation, we compared **RCM** with two matrix-based representations: the **Twin Confusion Matrix (TCM)**, which is a special case of the EnsembleMatrix [6] and the **Combined Confusion Matrix (CCM)** which is based on Basak et al. [11] matrix. This way, we compare visualizations of the three categories: juxtaposition, superimposition and special encoding [1]. We call **TCM** the juxtaposition of two confusion matrices, one for A, the other for B. We consider this representation as the baseline of our comparisons. The **CCM** superimposes the matrices of both A and B in a single matrix. Each cell contains two nested colored rectangles indicating the performance of each model on that cell.

The goal of the evaluation is to study three hypothesis: first, we make the assumption that comparing two models with a superimposed visualization or a visualization that encodes the difference is better than a side-by-side visualization (H_{vis}). Second, we want to study, for the comparison of two models, if symbols are better suited than colors (H_{symb}). Third, we make the assumption that finding the darkest color is easier than comparing contrasts (H_{dark}).

User Evaluation

In this section, we describe the protocol we established to build our evaluation. It is based on the Purchase protocol [3]. We define four tasks to solve on these visualizations, how data have been generated, and how the sessions were organized.

A. Visualizations

This evaluation targets the comparison of two models on three visualizations (TCM, CCM and RCM). This means we are not interested in how a model performs alone, but relatively to another. Thus, none of the tasks targets the performance of a single model. We focus on the functionalities all visualizations share.

B. Tasks

Our tasks are inspired from those defined for Squares [13] to compare models. For example, one of their tasks consists in detecting the classifier with the largest number of errors. Another of their tasks consists in comparing distributions of classes. We have retained the two ideas of giving the best model under a given condition (T1 and T2) and comparing two distributions (T3 and T4), which led to the following tasks, formulated as questions:

- T1: Which model does correctly classify more samples for the highlighted class?
- T2: Which model is the best on the highlighted cell?
- T3: On which class is there the most difference of correct predictions between the two models?
- T4: Among incorrect predictions, which pair (GT, prediction) shows the greatest difference between the two models?

For T1 (resp. T2), given a diagonal cell (resp. non-diagonal cell) the best model on that cell has to be selected. In both cases, the given cell is chosen randomly and displayed with a distinct color directly on the matrix or on the matrices. Both the line and the column of the chosen cell are framed with that color. To solve T3 (resp. T4), the diagonal cell (resp. non-diagonal cell) that show the greatest difference between models A and B has to be selected. The difference between T1 and T2 (resp. T3 and T4) is that the reasoning is inverted. Indeed, diagonal cells refer to correct predictions while non-diagonal cells refer to incorrect predictions.

C. Dataset

We randomly generated data (matrices) to ensure generalizability of the results [11]. For each trial, we generated a pair of matrices of same size, one for A and one for B. We generated two pairs of matrices per task (T1-T4), per visualization (TCM, CCM, RCM) per size (2x2, 10x10 or 30x30). We added two constraints in our generator that aims to simulate models with relatively average or high performance, which is often the case with real models. First, we arbitrarily fixed the same density at 0.3 for all matrices, which means only 30% of the nondiagonal cells of each individual matrix contain at least one sample. Thus, the percentage of nondiagonal cells that are different between A and B are in $[30\%, 60\%]$. Our generator imposes as a second constraint that all classes of each model have a minimal recall of 0.3. This means for each class, at least 30% of samples of that class are correctly classified, on the diagonal. For T1 and T2, the highlighted cells are chosen randomly with a uniform probability distribution. For all matrices created for a specific task, we know the answer by construction. For each trial, we verified the uniqueness of the answer. We generated several additional pairs of matrices with various sizes for the tutorial, and for training [3].

D. Protocol

During the whole evaluation, all the participants are supervised by the same person. Each participant starts the evaluation with a tutorial that explains what confusion matrices are, how TCM, CCM and RCM are built, and the four tasks to solve. Fig. 1 and Fig. 2 are parts of the tutorial and are accompanied by textual explanations and examples. For each task in the

tutorial, the participant is given one answered example per visualization followed by one example per visualization to train. The participant can see the correction for all examples, retry or come back at any point in the tutorial, and ask as many questions as she wants to the supervisor until the end of the tutorial. Once the participants have finished the tutorial, they are all given the same instructions and indications for the next part of the evaluation. The instructions insisted on not answering randomly or too fast, but correctly as fast as possible.

For the whole evaluation, the resolution of all matrices in pixels are the same. 2x2 matrices are displayed in 300x350, 10x10 matrices are displayed in 400x450, and 30x30 matrices are displayed in 500x550.

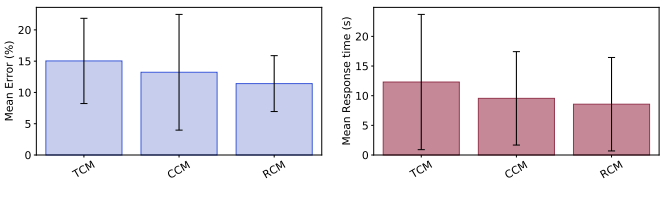
During the evaluation, the trials are grouped by task since we do not measure how participants adapt to a task, but how they solve it. For generalizability, two latin squares are used to ensure the participants do not face neither the same trials in the same order nor the same tasks in the same order. However, all participants are given one minute to answer each trial, with three seconds break between each trial. A break is scheduled between each task. At the end, the participants are asked to give their opinions on the visualizations and tasks through a short survey.

RESULTS

In this section, we present the results of our user evaluation. We analyze the results under three conditions which are the three visualizations (TCM, CCM, RCM).

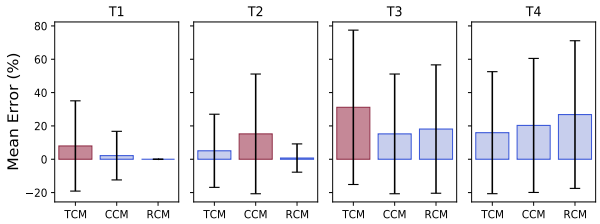
Participants. 26 volunteers performed the evaluation in total with a mean age of 31.5 years. We had to not consider 3 of them due to their lack of focus and because some trials were answered randomly according to both the participants and the supervisor. For each participant, we collected both the answer to compute the Error Percentage (EP), and the mean Response Time (RT) per trial. Some participants did not answer within the time limit for specific trials. We decided to not include their answers to these trials in the response time measurements. In the following, we report the statistical analysis and results.

Significance per visualization. We start with a high-level comparison to detect if there is a global significant difference between the visualizations for both EP and RT. For at least one condition, the data are not normally distributed for both EP and RT. We chose a non-parametric statistical approach, using the Kruskal-Wallis independent measures test, with a significance threshold of 0.05. This test indicates an overall significant difference on the mean RT ($p\text{-}value_{RT} = 4.9 * 10^{-8} < 0.05$). However, the same test on the EP does not reveal any significant difference ($p\text{-}value_{EP} = 0.17 > 0.05$). For further posthoc pairwise comparison on RT, we use the Conover test with corrected $p\text{-}values$ which reveals a significant difference between all pairs of conditions. In Fig. 4, a red bar indicates a significant pairwise difference to all other conditions. The mean response times of all conditions reveal that globally, *RCM* is the most efficient visualization. We conduct further experiments to analyze the reason of these differences.

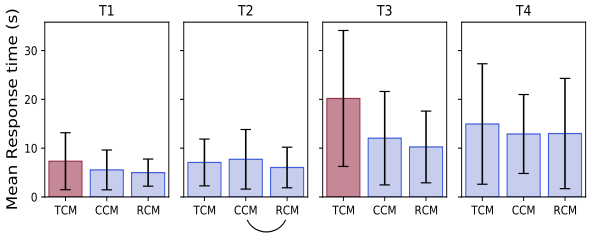


(a) Mean Error Percentage (MEP) per visualization (b) Mean Response time (MRT) per visualization

Fig. 4: Mean Error Percentages and Mean Response Times with standard deviation bars per visualization. Statistical tests showed that the visualization had a significant effect on the MRT. A bar with a red color indicates that the posthoc pairwise comparisons of this visualization against all others is significant ($p\text{-}value < 0.05$) according to Conover test [17].



(a) Mean Error Percentage (MEP) per task, per visualization



(b) Mean Response time (MRT) per task, per visualization

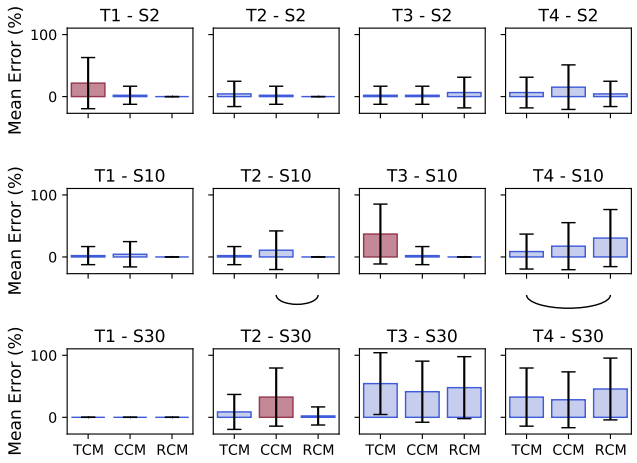
Fig. 5: Mean Error Percentages and Mean Response Times with standard deviation bars per task and per visualization. A red bar indicates that the posthoc pairwise comparisons of this visualization against all others is significant. An arc connecting two blue bars indicates a significant posthoc comparison between the two visualizations.

Significance per task, per visualization. By aggregating data per task and per visualization before computing the mean RT and Error, we aims to highlight what visualizations characteristics may be more efficient than others. At this level, a Kruskal-Wallis test reveals an overall significant difference for both EP and the mean RT per task. Fig.5 shows the posthoc pairwise comparisons between each visualization. From this figure, we can notice several interesting results. For T1 and T3, the only significant differences are between TCM and the other visualizations. TCM has the worst response times and the worst error percentages. The juxtaposition of the two matrices of A and B forces the participants to memorize

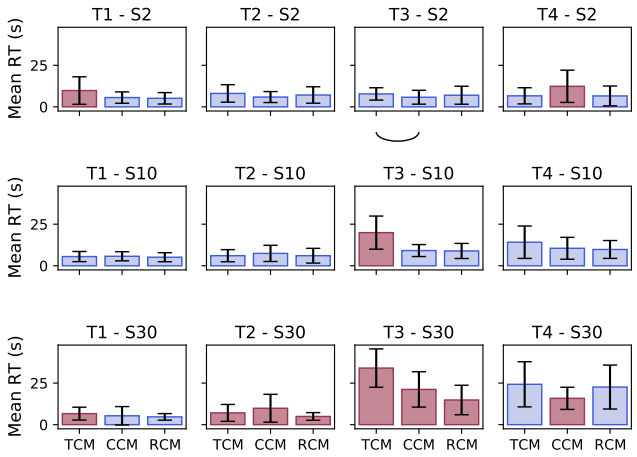
colors for comparisons of two distant cells which increases the response time of each trial and the risk of errors. Interestingly, the pairwise tests do not reveal any significant difference neither for EP nor RT when comparing all nondiagonal cells to detect the greatest difference of incorrect predictions (T4). Though, we observe a large standard deviation for this task that has to be explored with further investigation. The second significant difference lies in T2. There is a significant mean RT difference between CCM and RCM, RCM being more efficient. However, CCM is more confusing to participants than other visualizations. Participants answer relatively fast but tend to be misled by the color inversion of CCM between diagonal and nondiagonal cells, until they realize their mistake.

Significance per task, per matrix size, per visualization. From these results, we investigate further to see whether the size of matrices impact the performances on the different visualizations. These measures are shown in Fig. 6. In this figure, the same phenomenon described before appears on CCM with T2. The EP difference is more relevant as the matrices size grows. The examples of CCM with a size of 30x30 have indeed a small visible difference between light colors. With comparable matrices size, RCM outperforms both TCM and CCM: participants make less errors within less time. As shown in Fig. 6, statistical tests also reveal participants answered significantly faster for RCM without committing significantly more or less mistakes. The only case CCM outperforms RCM is for the comparison of all nondiagonal cells to find the greatest difference (T4) on large matrices (30x30). The reason of RCM being less efficient than CCM could be the density of black symbols in the matrices that affect the light colors perception. The last important remark is that TCM almost never outperforms the other visualizations.

Participants preference. All participants rated the three visualizations between 0 and 4, 4 being the best mark. We collected their rates for each individual task, and globally. We analyzed them with the Kruskal-Wallis test. Globally, the overall difference between the three visualizations is significant. Posthoc comparisons using the Conover test reveal a significant difference between all pairs after $p\text{-}values$ correction. In general, users preferred **RCM** to **CCM** to **TCM**. This tendency is exactly the same for each individual task (see Fig 7).



(a) Mean Error Percentage (MEP) per task, per matrix size, per visualization



(b) Mean Response time (MRT) per task, per matrix size, per visualization

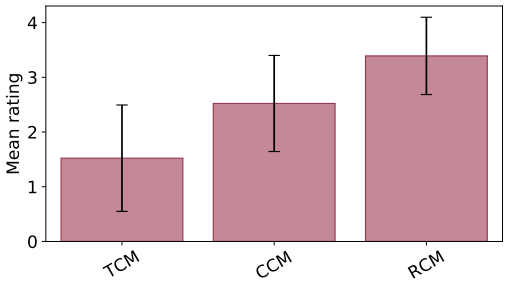
Fig. 6: Mean Error Percentages and Mean Response Times with standard deviation bars per task, per matrix size, and per visualization. A red bar indicates that the posthoc pairwise comparisons of this visualization against all others is significant. An arc connecting two blue bars indicates a significant posthoc comparison between the two visualizations.

Discussion and Conclusion

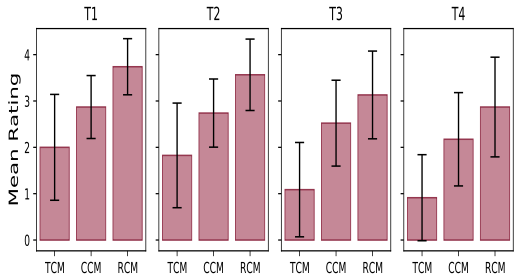
The user evaluation we conducted reveals that most users are not comfortable with manipulating side-by-side visualizations to compare models. Such comparisons are relatively slow and require from the user to memorize at least four different colors at a time. As we expected, the spatial distance between the elements to compare makes the tasks hard to solve. This validates our hypothesis H_{vis}.

Surprisingly, the symbols do not make the tasks T1 and T2 faster to solve. After surveying our participants, it appeared that, unlike for the two other visualizations (**TCM** and **CCM**), they needed to check systematically the legend on the left to remember the meaning of the + and - symbols. This validation step they performed only for **RCM** artificially increased their response time. We strongly believe that an expert of the three visualizations that would be able to learn the meaning of the symbols without constantly using the legend, would answer significantly more efficiently on **RCM** (H_{symb}).

The results on task T3 validated half of the hypothesis H_{dark}. Indeed, participants found the darkest diagonal cell of **RCM** faster than the greatest contrast of **TCM** or **CCM**. However, when the matrix is large and dense, the symbols that better highlight the small differences tend to disturb the perception of the colors in the cells. Some participants mentioned this problem for large matrices. To solve this



(a) Mean ratings per visualization



(b) Mean ratings per task, per visualization

Fig. 7: Mean Ratings per visualization, and per task per visualization with standard deviation bars. All bars are red, indicating that all the rating differences between pairs of visualizations are significant. The means and standard deviation indicate RCM is the most preferred visualization.

problem, we suggest to make **RCM** interactive by adding a slider that determines the minimum required amount of differences in a cell, from which symbols $+$ and $-$ need to be displayed. Otherwise, this threshold could control the percentage of cells that show the greatest difference between the two models in which symbols have to be displayed.

Participants also mentioned the neighborhood of a cell can falsify the perception of its color(s), especially for **TCM** and **CCM**, when the aforesaid cell is close to the diagonal dark cells.

In the end, we validate the effectiveness of the Relative Confusion Matrix to compare two classification models, up to 30 classes. To address more classes and reduce the number of rows and columns of the confusion matrices, we suggest to establish a hierarchy that groups similar classes if it does not exist, and apply hierarchical interactions, as described by Görtler et. al. [9]. To enhance the most important symbols for large and dense matrices, we suggest to add an interaction like a slider to threshold the display of symbols.

Eventually, one may not want to compare only two classification models. To adapt our visualization to more classification models, we can imagine to designate each model with a unique symbol, that can be written with as few pixels as possible. Instead of displaying $+$ and $-$ symbols in the cells, we could display the symbol representing the best model in this cell. Then, the color of the cells could represent how much this model is better in this cell than the other.

ACKNOLEDGMENT

We thank the Nouvelle-Aquitaine Region, Bordeaux Métropole and SUEZ, le LyRE for mainly funding and supporting this work through the Convention N°AAPR2020-2019-8171810.

REFERENCES

[1] Gleicher, M. (2017). Considerations for visualizing comparison. IEEE transactions on visualization and computer graphics, 24(1), 413-423.

[2] Hinterreiter, A., Ruch, P., Stitz, H., Ennemoser, M., Bernard, J., Strobelt, H., & Streit, M. (2020). ConfusionFlow: A model-agnostic visualization for temporal analysis of classifier confusion. IEEE Transactions on Visualization and Computer Graphics.

[3] Purchase, H. (2012). Experimental Human-Computer Interaction: A Practical Guide with Visual Examples. Cambridge University Press.

[4] Mackinlay, J. (1986). Automating the Design of Graphical Presentations of Relational Information. ACM Trans. Graph., 5(2), 110-141.

[5] Li, G., Wang, J., Shen, H. W., Chen, K., Shan, G., Lu, Z. (2020). CNNPruner: Pruning Convolutional Neural Networks with Visual Analytics. IEEE Transactions on Visualization and Computer Graphics, 1.

[6] Talbot, J., Lee, B., Kapoor, A., Tan, D. S. (2009). EnsembleMatrix: interactive visualization to support machine learning with multiple classifiers. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1283–1292.

[7] Alsallakh, B., Hanbury, A., Hauser, H., Miksch, S., Rauber, A. (2014). Visual methods for analyzing probabilistic classification data. IEEE Transactions on Visualization and Computer Graphics, 20(12), 1703–1712.

[8] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv preprint arXiv:2010.16061.

[9] Görtler, J., Hohman, F., Moritz, D., Wongsuphasawat, K., Ren, D., Nair, R., Kirchner, M., Patel, K. (2021). Neo: Generalizing Confusion Matrix Visualization to Hierarchical and Multi-Output Labels. ArXiv Preprint ArXiv:2110.12536.

[10] Amershi, S., Chickering, M., Drucker, S. M., Lee, B., Simard, P., Suh, J. (2015). Modeltracker: Redesigning performance analysis tools for machine learning. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 337–346.

[11] Basak, A., Bach, B., Henry Riche, N., Isenberg, T., Fekete, J.-D. (2013). Weighted graph comparison techniques for brain connectivity analysis. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 483–492.

[12] Fawcett, T. (2006). An introduction to ROC analysis. Pattern recognition letters, 27(8), 861-874.

[13] Ren, D., Amershi, S., Lee, B., Suh, J., Williams, J. D. (2016). Squares: Supporting interactive performance analysis for multiclass classifiers. IEEE Transactions on Visualization and Computer Graphics, 23(1), 61–70.

[14] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence, 39(6):1137–1149, 2016.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proc. of the IEEE conf. on computer vision and pattern recognition, pp.580–587, 2014.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90, 2017

[17] W. J. Conover and R. L. Iman (1979), On multiple-comparisons procedures, Tech. Rep. LA-7677-MS, Los Alamos Scientific Laboratory.