Evaluation Criteria for Instance-based Explanation

Kazuaki Hanawa^{1,2,*} **Sho Yokoi**^{2,1,*} **Satoshi Hara**^{3,†} **Kentaro Inui**^{2,1,*}
^{1}RIKEN, Japan ^{2}Tohoku University, Japan ^{3}Osaka University, Japan
*`{hanawa,yokoi,inui}@ecei.tohoku.ac.jp` †`satohara@ar.sanken.osaka-u.ac.jp`

Abstract

Explaining predictions made by complex machine learning models helps users understand and accept the predicted outputs with confidence. Instance-based explanation provides such help by identifying relevant instances as evidence to support a model's prediction result. To find relevant instances, several *relevance metrics* have been proposed. In this study, we ask the following research question: "Do the metrics actually work in practice?" To address this question, we propose two sanity check criteria that valid metrics should pass, and two additional criteria to evaluate the practical utility of the metrics. All criteria are designed in terms of whether the metric can pick up instances of desirable properties that the users expect in practice. Through experiments, we obtained two insights. First, some popular relevance metrics do not pass sanity check criteria. Second, some metrics based on cosine similarity perform better than other metrics, which would be recommended choices in practice. We also analyze why some metrics are successful and why some are not. We expect our insights to help further researches such as developing better explanation methods or designing new evaluation criteria.

1 Introduction

Explaining predictions made by complex machine learning models helps users understand and accept the predicted outputs with confidence [Ribeiro et al., 2016, Lundberg and Lee, 2017, Guidotti et al., 2018, Adadi and Berrada, 2018, Molnar, 2020]. Instance-based explanation attains this goal by showing relevant instances as evidence of the prediction made by the model. This form of explanation is particularly similar to the ways humans make decisions by referring to their prior experiences [Klein and Calderwood, 1988, Klein, 1989, Read and Cesa, 1991]. Cunningham et al. [2003] found that, thanks to the compatibility with the humans' decision-making process, instance-based explanation can make users more confident about the predictions made by the models.

A typical instance-based explanation approach is to raise training instances that are relevant to the prediction of a test instance [Koh and Liang, 2017, Khanna et al., 2019]. In this paper, we particularly focus on the methods that use a *relevance metric*, and propose evaluation criteria for such metrics.

Definition 1 (Instance-based Explanation based on Relevance Metric)**.** Let $\mathcal{D} = \{\boldsymbol{z}_{\mathrm{train}}^{(i)} = (\boldsymbol{x}_{\mathrm{train}}^{(i)}, y_{\mathrm{train}}^{(i)})\}_{i=1}^{N}$ be a set of training instances, and $\boldsymbol{x}_{\mathrm{test}}$ be a test input of interest whose predicted output is given by $\widehat{y}_{\mathrm{test}} = f(\boldsymbol{x}_{\mathrm{test}})$ with a model f. An instance-based explanation method raises the most relevant training instance $\bar{\boldsymbol{z}} \in \mathcal{D}$ to the test instance $\boldsymbol{z}_{\mathrm{test}} = (\boldsymbol{x}_{\mathrm{test}}, \widehat{y}_{\mathrm{test}})$ by $\bar{\boldsymbol{z}} = \mathrm{argmax}_{\boldsymbol{z}_{\mathrm{train}} \in \mathcal{D}} R(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}})$ using a relevance metric $R(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}_{\mathrm{train}}) \in \mathbb{R}$.

There are various relevance metrics such as *similarity* [Caruana et al., 1999], *kernel functions* [Kim et al., 2016, Khanna et al., 2019], and *influence function* [Koh and Liang, 2017].

Note that there is another approach for instance-based explanation that uses specific models that can provide explanations by its design [Kim et al., 2014, Plötz and Roth, 2018, Chen et al., 2019].

Preprint. Under review.

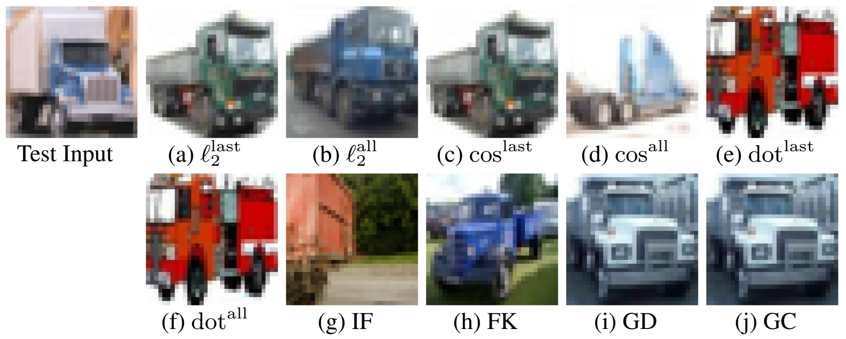

Test Input
(a) ℓ_2^{last}
(b) ℓ_2^{all}
(c) $\cos^{\mathrm{last}}$
(d) $\cos^{\mathrm{all}}$
(e) $\mathrm{dot}^{\mathrm{last}}$
(f) $\mathrm{dot}^{\mathrm{all}}$
(g) IF
(h) FK
(i) GD
(j) GC

Figure 1: Relevant instances selected for the test input using several relevance metrics (a)–(j) on CIFAR10 with CNN. See Table 2 and Section 1.2 for the metrics.

Table 1: The proposed evaluation criteria for relevance metrics.

	Sanity Checks		**Utility Evaluations**	
	Identical Instance Test	Identical Class Test	Top-k Identical Class Test	Identical Subclass Test
Can the metric raise a training instance...	...itself?	...of the same class?	...of the same class?	...of the same subclass?

However, we set aside these specific models and focus on generic relevance metrics because of their applicability to a wide range of problems.

Research Question Several studies have demonstrated the usefulness of relevance metrics for instance-based explanation [Kim et al., 2016, Koh and Liang, 2017, Khanna et al., 2019]. Our research question is whether these relevance metrics have desirable properties that the users expect in practice. In the visual explanation literature, a pioneering study by Adebayo et al. [2018] showed that some saliency methods have undesirable properties. An important implication from their study is that quantitative evaluation of explanation methods, which, in our case, are the relevance metrics, is much more important than demonstrating a few fascinating examples. As an example, in Figure 1, we show some relevant instances found in CIFAR10 [Krizhevsky, 2009] using several relevance metrics. One may find from this example that most of the metrics can raise fairly reasonable instances as relevant. However, in light of Adebayo et al. [2018], these individual examples are not helpful for the quantitative evaluation of the relevance metrics.

Contributions We provide a quantitative evaluation of popular relevance metrics. For this purpose, we design two types of evaluation criteria, as shown in Table 1. The first type of criteria includes sanity checks that every valid relevance metric should pass, similar to the study of Adebayo et al. [2018] for evaluating the saliency methods. The second type of criteria is used to evaluate the practical utility of relevance metrics. All criteria are designed in terms of whether the metric can select instances of desirable properties that the users expect in practice.

Our results based on empirical evaluations with these criteria are twofold. First, some relevance metrics do not pass the sanity check criteria as shown in Table 2, indicating they are not desirable for instance-based explanation. Second, some metrics based on cosine similarity perform better than other metrics and will be of recommended choices in practice. We also analyze why some metrics are successful and why some are not.

1.1 Preliminaries

Notations For vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^p$, we denote the dot product by $\langle \boldsymbol{a}, \boldsymbol{b} \rangle := \sum_{i=1}^{p} a_i b_i$, the ℓ_2 norm by $\|\boldsymbol{a}\| := \sqrt{\langle \boldsymbol{a}, \boldsymbol{a} \rangle}$, and the cosine similarity by $\cos(\boldsymbol{a}, \boldsymbol{b}) := {\langle \boldsymbol{a}, \boldsymbol{b} \rangle}/{\|\boldsymbol{a}\| \|\boldsymbol{b}\|}$.

Classification Problem We consider a standard classification problem as the evaluation benchmark. The model is the conditional probability $p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ with parameter $\boldsymbol{\theta}$. Let $\widehat{\boldsymbol{\theta}}$ be a trained parameter $\widehat{\boldsymbol{\theta}} = \mathrm{argmin}_{\boldsymbol{\theta}}\, \mathcal{L}_{\mathrm{train}} := \frac{1}{N} \sum_{i=1}^{N} \ell(\boldsymbol{z}_{\mathrm{train}}^{(i)}; \boldsymbol{\theta})$, where the loss function ℓ is the cross entropy $\ell(\boldsymbol{z}; \boldsymbol{\theta}) =$

Table 2: List of the relevance metrics we consider and their sanity check results. Only five metrics passed both of the two sanity check criteria. The details of the relevance metrics, the evaluation criteria, and the evaluation procedures can be found in Sections 1.2, 3, and 4, respectively.

			Sanity Checks	
Metrics		Abbrv.	Identical Instance Test	Identical Class Test
ℓ_2	$\phi(\boldsymbol{z}) = \boldsymbol{x}$	ℓ_2^x	Passed	Failed
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{last}}$	ℓ_2^{last}	Passed	Passed
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{all}}$	ℓ_2^{all}	Passed	Passed
Cosine	$\phi(\boldsymbol{z}) = \boldsymbol{x}$	$\cos^x$	Passed	Failed
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{last}}$	$\cos^{\text{last}}$	Passed	Passed
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{all}}$	$\cos^{\text{all}}$	Passed	Passed
Dot	$\phi(\boldsymbol{z}) = \boldsymbol{x}$	dot^x	Failed	Failed
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{last}}$	dot^{last}	Failed	Failed
	$\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{all}}$	dot^{all}	Failed	Failed
Gradient	Influence Function	IF	Failed	Failed
	Fisher Kernel	FK	Failed	Failed
	Grad-Dot	GD	Failed	Failed
	Grad-Cos	GC	Passed	Passed

$-\log p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ for an input-output pair $\boldsymbol{z} = (\boldsymbol{x}, y)$. The model classifies a test input $\boldsymbol{x}_{\text{test}}$ by assigning the class with the highest probability $\widehat{y}_{\text{test}} = \text{argmax}_y \, p(y \mid \boldsymbol{x}_{\text{test}}; \widehat{\boldsymbol{\theta}})$.

1.2 Relevance Metrics

We present an overview of the two types of relevance metrics considered in this study, *similarity metrics* and *gradient-based metrics*. See Table 2 for the list of metrics and their abbreviations.

Similarity Metrics We consider the following popular similarity metrics with a feature map $\phi(\boldsymbol{z})$.

- **$\boldsymbol{\ell_2}$ Metric**: $R_{\ell_2}(\boldsymbol{z}, \boldsymbol{z}') := -\|\phi(\boldsymbol{z}) - \phi(\boldsymbol{z}')\|^2$, which is a typical choice for nearest neighbor methods [Hastie et al., 2009, Abu Alfeilat et al., 2019].
- **Cosine Metric**: $R_{\cos}(\boldsymbol{z}, \boldsymbol{z}') := \cos(\phi(\boldsymbol{z}), \phi(\boldsymbol{z}'))$, which is commonly used in natural language processing tasks [Mikolov et al., 2013, Arora et al., 2017, Conneau et al., 2017].
- **Dot Metric**: $R_{\text{dot}}(\boldsymbol{z}, \boldsymbol{z}') := \langle \phi(\boldsymbol{z}), \phi(\boldsymbol{z}') \rangle$, which is a kernel function used in kernel models such as SVM [Schölkopf et al., 2002, Fan et al., 2005, Bien and Tibshirani, 2011].

As the feature map $\phi(\boldsymbol{z})$, we consider (i) an identity map $\phi(\boldsymbol{z}) = \boldsymbol{x}$; (ii) the last hidden layer $\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{last}}$, which is the latent representation of the input $\boldsymbol{x}$, one layer before the output in a deep neural network; and, (iii) all hidden layers $\phi(\boldsymbol{z}) = \boldsymbol{h}^{\text{all}}$, where $\boldsymbol{h}^{\text{all}} = [\boldsymbol{h}^1, \boldsymbol{h}^2, \ldots, \boldsymbol{h}^{\text{last}}]$ is the concatenation of all latent representations in the network. Note that the metrics with the identity map merely measure the similarity of the inputs without model information. We adopt these metrics as naive baselines to contrast with other advanced metrics that utilize the information of the models.

Gradient-based Metrics Gradient-based metrics use the gradient $\boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}} := \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$ to measure the relevance. We consider the four metrics: Influence Function (IF) [Koh and Liang, 2017], Fisher Kernel (FK) [Khanna et al., 2019], Grad-Dot (GD) [Yeh et al., 2018, Charpiat et al., 2019], and Grad-Cos (GC) [Perronnin et al., 2010, Charpiat et al., 2019]. See Appendix A for the detail.

- **IF**: $R_{\text{IF}}(\boldsymbol{z}, \boldsymbol{z}') := \langle \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{H}^{-1} \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'} \rangle$
- **GD**: $R_{\text{GD}}(\boldsymbol{z}, \boldsymbol{z}') := \langle \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'} \rangle$,
- **FK**: $R_{\text{FK}}(\boldsymbol{z}, \boldsymbol{z}') := \langle \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{I}^{-1} \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'} \rangle$,
- **GC**: $R_{\text{GC}}(\boldsymbol{z}, \boldsymbol{z}') := \cos(\boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}}, \boldsymbol{g}_{\widehat{\boldsymbol{\theta}}}^{\boldsymbol{z}'})$,

where $\boldsymbol{H}$ and $\boldsymbol{I}$ are the Hessian and Fisher information matrices of the loss $\mathcal{L}_{\text{train}}$, respectively.

2 Related Work

Evaluation of Metrics Metrics between instances play an essential role in many machine learning problems. For example, the distance between instances is essential for the distance-based methods such as nearest neighbor methods [Hastie et al., 2009]. Another example is kernel models where the kernel function represents the relationship between two instances [Schölkopf et al., 2002]. There are several studies evaluating which metrics are desirable for specific tasks [Hussain et al., 2011, Hu et al., 2016, Li and Li, 2018, Abu Alfeilat et al., 2019]. The goal of these studies is to find metrics that improve classification accuracy. The validity of metrics for instance-based explanation is usually left out of scope. Our goal is to evaluate the validity of relevance metrics for instance-based explanation; thus, the findings in these previous studies are not directly applicable to our goal.

Evaluation of Relevance Metrics One way of evaluating relevance metrics is to solve the *data cleansing problem* [Koh and Liang, 2017, Khanna et al., 2019, Hara et al., 2019]. The idea is that an outlier in the training set (e.g., an instance with an incorrect label) must be highly *relevant* to the model's prediction. If a relevance metric can find outliers included in the training set with a small false positive rate, that metric can be considered effective. However, outliers are not always effective evidence of the model's prediction, and users may not accept the model's prediction confidently if instances with incorrect labels are provided as evidence. Thus, our focus is on evaluation criteria for users who expect supporting evidence for model prediction rather than outliers.

3 Evaluation Criteria for Relevance Metrics

We propose four criteria for evaluating relevance metrics: the first two for sanity checks, and the other two for evaluating the practical utilities.

We take MNIST [LeCun et al., 1998] as an example to explain the criteria. Here, input $\boldsymbol{x}$ is an image of a handwritten digit, and output y is one of 10 classes (i.e., "0" to "9").

3.1 Two Sanity Check Criteria

We propose two sanity check criteria. Although the criteria may look trivial, some popular relevance metrics do not satisfy these requirements, as demonstrated in Section 4.1.

The First Criterion: Identical Instance Test One of the requirements for intelligent agents is to be able to solve the same problem that has been solved already. In other words, if the test instance itself appeared during model training, then the model would give the correct answer to such an instance simply because it has been observed previously. For example, assume the model predicts "3" for input image $\boldsymbol{x}_{\mathrm{test}}$, and a copy of the test instance $\boldsymbol{z}_{\mathrm{train}} = (\boldsymbol{x}_{\mathrm{test}}, 3)$ exists in the training set $\mathcal{D}$. In this case, the most relevant training instance for this prediction is trivially the copy of the test instance. This observation leads to our first sanity check criterion.

Definition 2 (Identical Instance Test)**.** The most relevant instance for the prediction of a training instance $\boldsymbol{z}_{\mathrm{train}} = (\boldsymbol{x}_{\mathrm{train}}, y_{\mathrm{train}}) \in \mathcal{D}$ with a correct prediction $\widehat{y}_{\mathrm{train}} = y_{\mathrm{train}}$ is the training instance itself: $\mathrm{argmax}_{\boldsymbol{z} \in \mathcal{D}} R(\boldsymbol{z}_{\mathrm{train}}, \boldsymbol{z}) = \boldsymbol{z}_{\mathrm{train}}$.

Note that, for training instances with incorrect predictions, we do not require the identical instance to be considered as relevant. For such cases, some other instances that mislead the predictions should be raised as relevant.

The Second Criterion: Identical Class Test Another requirement for intelligent agents is to appropriately generalize their knowledge to unseen problems, which is a fundamental goal of machine learning. To classify test instances to a certain class, the model needs to be trained using training instances from the same class with a certain generalization property. For example, a model can classify input image $\boldsymbol{x}_{\mathrm{test}}$ to class $\widehat{y}_{\mathrm{test}} = 5$ because it has seen images of "5" during training. Thus, the classification of the test instance is highly dependent on training instances of the same class. This observation leads to the second sanity check criterion.

Definition 3 (Identical Class Test)**.** The most relevant instance for the prediction of a test instance $\boldsymbol{z}_{\mathrm{test}} = (\boldsymbol{x}_{\mathrm{test}}, \widehat{y}_{\mathrm{test}})$ is a training instance of the same class as the given test instance.

$$\underset{\boldsymbol{z}=(\boldsymbol{x},y)\in\mathcal{D}}{\mathrm{argmax}}\ R\big(\boldsymbol{z}_{\mathrm{test}}, \boldsymbol{z}\big) = (\bar{\boldsymbol{x}}, \bar{y}) \implies \bar{y} = \widehat{y}_{\mathrm{test}}. \tag{1}$$

Here, we do not require the test predictions to be correct unlike the identical instance test.

3.2 Two Utility Evaluation Criteria

We now propose two additional evaluation criteria based on desirable properties in practice.

The Third Criterion: Top-k Identical Class Test The identical class test requires the most relevant instance to be of the same class as the test instance. In practice, users can be more confident about a model's output if several instances are provided as evidence. In other words, we expect that the most relevant and a first few relevant instances will be of the same class. This observation leads to the third evaluation criterion, which is a generalization of the identical class test.

Definition 4 (Top-k Identical Class Test)**.** For $\boldsymbol{z}_{\text{test}} = (\boldsymbol{x}_{\text{test}}, \widehat{y}_{\text{test}})$, let $\bar{\boldsymbol{z}}^j = (\bar{\boldsymbol{x}}^j, \bar{y}^j)$ be a training instance with the j-th largest relevance score. Then, we require $\bar{y}^j = \widehat{y}_{\text{test}}$ for any $j \in \{1, 2, \ldots, k\}$.

The Fourth Criterion: Identical Subclass Test In the fourth criterion, we consider the classification problem with subclasses in each class. In classification problems, it is common for class labels to have hierarchical structures. A typical example is ImageNet [Deng et al., 2010] whose image labels are taken from WordNet [Miller, 1995], which provides a hierarchy of the words. In such a case, each class typically has several latent subclasses.

To derive the fourth criterion, consider the task of classifying handwritten digits to two classes, "odd" and "even". In this task, subclasses, e.g., "3" and "7," belong to the "odd" class, and subclasses, e.g., "2" and "8," belong to the "even" class. Here, assume that the model classified a test instance of "7" to the "odd" class. The question is what kind of instances are expected as evidence for this output. The identical class test accepts any training instances from the "odd" class. However, from a practical perspective, the ideal instances should belong to the subclass "7" because the model classifies the test instance based on training instances whose subclass is "7". This requirement leads to the fourth evaluation criterion below.

Definition 5 (Identical Subclass Test)**.** Let $s(\boldsymbol{z})$ be its subclass for an instance $\boldsymbol{z} = (\boldsymbol{x}, y)$. Here, the most relevant instance for the prediction of a test instance $\boldsymbol{z}_{\text{test}} = (\boldsymbol{x}_{\text{test}}, \widehat{y}_{\text{test}})$ with a correct prediction $\widehat{y}_{\text{test}} = y_{\text{test}}$ is the training instance of the same subclass as the test instance.[1]

$$\operatorname*{argmax}_{\boldsymbol{z} \in \mathcal{D}} R\big(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}\big) = \bar{\boldsymbol{z}} \implies s(\bar{\boldsymbol{z}}) = s(\boldsymbol{z}_{\text{test}}). \tag{2}$$

Note that the identical subclass test depends on whether the model has obtained internal representations that can distinguish between subclasses. All datasets and models that we used in our experiments are considered to satisfy this condition. For more details, see Appendix D.

4 Evaluation Results

Here, we examine the validity of the relevance metrics based on the proposed criteria.[2] For this evaluation, we used two image datasets (MNIST [LeCun et al., 1998], CIFAR10 [Krizhevsky, 2009]) and two text datasets (TREC [Li and Roth, 2002], Customer Feedback (CF) [Liu et al., 2017]). As benchmarks, we employed logistic regression and deep neural networks trained on these datasets. The details of the datasets, models, and computing infrastructure we used are given in Appendix B.

Procedure Our experimental procedure for each evaluation criterion is as follows.

1. Train models using a subset of the training instances.[3] Then, randomly sample 500 test instances from the training set or test set depending on the evaluation criterion.
2. For each test instance, compute the relevance score with all the instances used for training. Then, compute the success rate, which is the ratio of test instances that passed the evaluation criterion.

We repeated this procedure 10 times and report the average success rate.

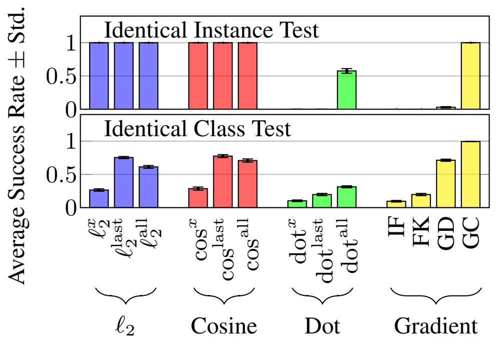
In this section, we mainly present the results for CIFAR10 with CNN, and CF with Bi-LSTM. The other results were similar, and can be found in Appendix E.

Summary of the Results We summarize the main results prior to discussing individual result.

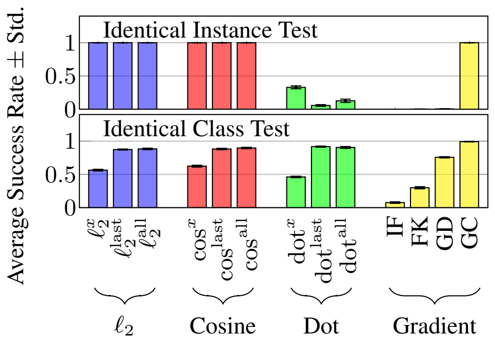
[1]We require correct predictions in this test because the subclass does not match for incorrect cases.

[2]Our code is: `https://github.com/k-hanawa/criteria_for_instance_based_explanation`

[3]We randomly sampled 10% of MNIST and CIFAR10, and 50% of TREC and CF.



(a) CIFAR10 with CNN



(b) CF with Bi-LSTM

Figure 2: Sanity Checking with Identical Instance Test and Identical Class Test

- Only five metrics ℓ_2^{last}, ℓ_2^{all}, $\cos^{\text{last}}$, $\cos^{\text{all}}$, and GC passed the sanity checks. The results indicate that the other metrics are inappropriate for instance-based explanation.
- Two cosine-based metrics, $\cos^{\text{all}}$ and GC, outperformed the other metrics; thus, they would be the recommended choices in practice.

In Section 5, we analyze why some metrics are successful and why some are not.

4.1 Sanity Checks

We first present the results for the sanity checks. For the identical instance test, we sampled 500 instances with correct predictions from the training set as test instances. For the identical class test, we sampled 500 instances from the test set as test instances.

Figure 2 shows the success rates for the identical instance test and the identical class test. To determine whether the metrics passed the sanity checks or not, we adopted 0.5 as a threshold for the success rate. If a metric passed the threshold for both tests on all the datasets, we considered the metric has passed the sanity checks. Note that 0.5 is a quite loose condition and much higher success rates would be required in practice. Therefore, it would be reasonable to consider metrics that failed to pass this threshold as impractical.

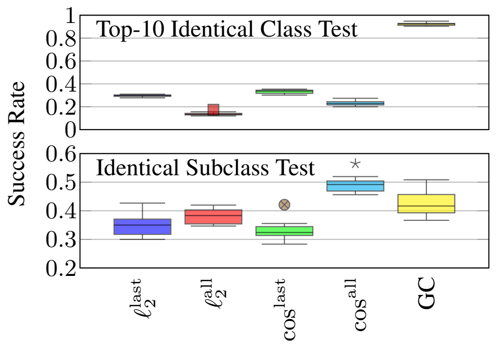
Below, we summarize three observations (see Table 2 also).

- Five metrics, ℓ_2^{last}, ℓ_2^{all}, $\cos^{\text{last}}$, $\cos^{\text{all}}$, and GC, passed both the identical instance test and the identical class test on all the datasets. The other metrics failed to pass at least one of the tests.
- The dot metrics and the gradient-based metrics, except for GC, failed the identical instance test. These metrics could not consider the training instance identical to the test instance as relevant. Note that, the ℓ_2 and cosine metrics can trivially pass the identical instance test by their definition.
- The metrics with the identity feature map ℓ_2^x, $\cos^x$, and dot^x failed the identical class test. Remind that we adopted these metrics without model information as naive baselines. As expected, these metrics had limited capabilities in practice.

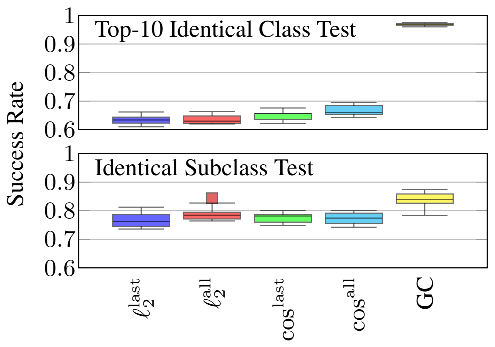
4.2 Utility Evaluations

We examine the five metrics that passed the sanity checks using the two additional utility evaluation criteria. For the top-k identical class test, we sampled 500 instances from the test set as test instances. For the identical subclass test, we used modified datasets: we split the dataset into two classes by randomly assigning existing classes to one of the two classes “A” and “B”. The new classes “A” and “B” now contain the original data classes as subclasses, which can be used for the identical subclass test. We then sampled 500 instances with correct predictions from the test set as test instances.

Figure 3 shows the success rates for the top-10 identical class test and the identical subclass test. There are two observations. First, GC outperformed the other metrics on the top-10 identical class test. Note that, a similar trend was observed also on the other datasets and the models (see Appendix E). Second, GC and $\cos^{\text{all}}$ performed well on the identical subclass test. The results on the other datasets



(a) CIFAR10 with CNN



(b) CF with Bi-LSTM

Figure 3: Utility Evaluations with Top-10 Identical Class Test and Identical Subclass Test

indicated that GC performed particularly well on the text datasets, while $\cos^{\text{all}}$ performed well on the image datasets (see Appendix E).

To conclude, the results on both sanity checks and utility evaluations indicate that only GC passed all the tests with high success rates, which would be a recommended choice for instance-based explanation. Note that $\cos^{\text{all}}$ would also be a reasonable choice in practice depending on the application.

5 Why some metrics are successful and why some are not?

We observed that the dot metrics and the popular gradient-based metrics, IF, FK, and GD, were not successful, while GC was. Here, we analyze why some metrics failed and why GC performed well. In Appendix C, we also discuss a way to *repair* IF and FK to improve their performances based on the findings in this section.

Failure of the Dot Metrics and Gradient-based Metrics To understand the failure, we reformulate IF, FK, and GD as dot metrics of the form $R_{\text{dot}}(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}}) = \langle \phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}) \rangle$ so that the following discussion to hold true for any metrics in this form. It is easy to see that IF, FK, and GD can be expressed in this form by defining the feature maps by $\phi(\boldsymbol{z}) = \boldsymbol{H}^{-1/2} g(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$, $\phi(\boldsymbol{z}) = \boldsymbol{I}^{-1/2} g(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$, and $\phi(\boldsymbol{z}) = g(\boldsymbol{z}; \widehat{\boldsymbol{\theta}})$, respectively.[4]

Given a criterion, let $\boldsymbol{z}_{\text{train}}^{(i)}$ be a desirable instance for a test instance $\boldsymbol{z}_{\text{test}}$. The failures of the dot metrics indicate the existence of an undesirable instance $\boldsymbol{z}_{\text{train}}^{(j)}$ such that $\langle \phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}^{(i)}) \rangle < \langle \phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}^{(j)}) \rangle$. The following sufficient condition for $\boldsymbol{z}_{\text{train}}^{(j)}$ is useful to understand the failure.

$$\|\phi(\boldsymbol{z}_{\text{train}}^{(i)})\| < \|\phi(\boldsymbol{z}_{\text{train}}^{(j)})\| \cos(\phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}^{(j)})). \tag{3}$$

The condition implies that any instance with an extremely large norm and cosine slightly larger than zero can be the candidate of $\boldsymbol{z}_{\text{train}}^{(j)}$. In our experiments, we observed that the condition on the norm is especially crucial. Indeed, as we can see in Figure 4, even though instances with extremely large norms were scarce, only such extreme instances were selected as relevant instances by IF, FK, and GD. That is, these metrics tend to consider such extreme instances as relevant ones. By contrast, GC was not attracted by large norms because it completely cancels out the norm through normalization.

Figure 5 shows some training instances frequently found to be relevant in the identical instance test on CIFAR10 with CNN. When using IF, FK, and GD, these training instances were frequently selected as relevant with several test instances regardless of their classes, simply because the training instances had large norms. In these metrics, the term $\cos(\phi(\boldsymbol{z}_{\text{test}}), \phi(\boldsymbol{z}_{\text{train}}))$ seems to have only negligible effects. GC was the only exception: the instances with small norms were selected as relevant with test instances of the same class only.

Success of GC We now analyze why GC performed well in particular in the identical class test. To simplify the discussion, we consider linear logistic regression whose conditional distribution

[4]We can make the Hessian matrix $\boldsymbol{H}$ to be positive definite using a regularization [Koh and Liang, 2017].

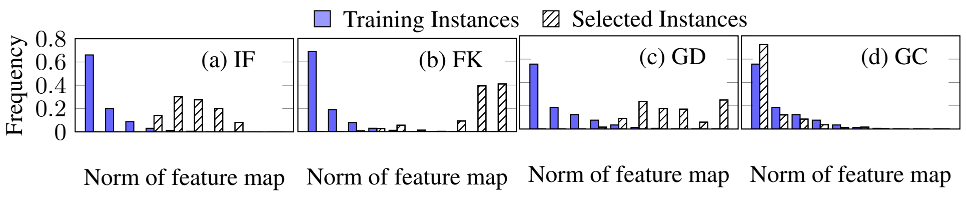
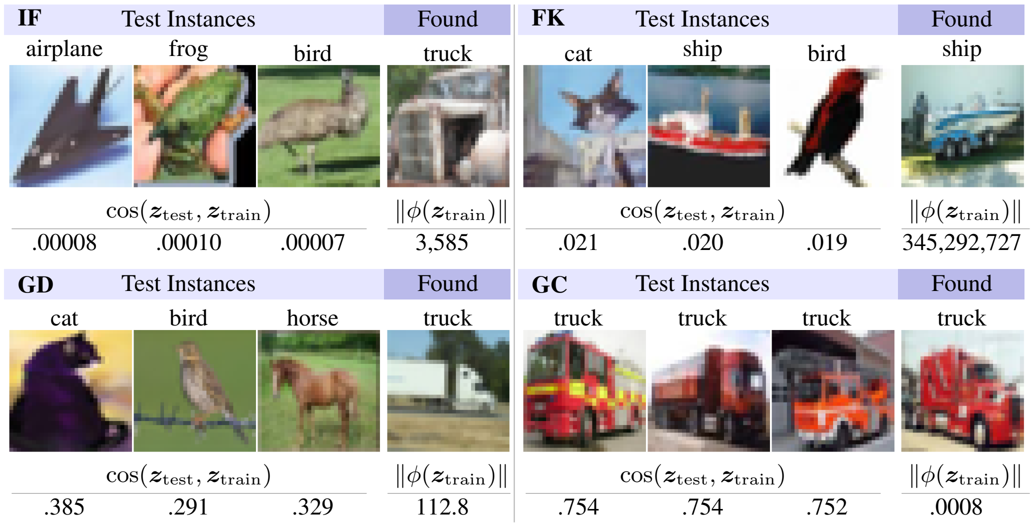


Figure 4: Distributions of the norms of the feature maps of all the training instances (colored), and the instances selected by the identical class test (meshed) on CIFAR10 with CNN.



IF
Test Instances
Found
airplane
frog
bird
truck
$\cos(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}})$
$\|\phi(\boldsymbol{z}_{\text{train}})\|$
.00008
.00010
.00007
3,585
FK
Test Instances
Found
cat
ship
bird
ship
$\cos(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}})$
$\|\phi(\boldsymbol{z}_{\text{train}})\|$
.021
.020
.019
345,292,727
GD
Test Instances
Found
cat
bird
horse
truck
$\cos(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}})$
$\|\phi(\boldsymbol{z}_{\text{train}})\|$
.385
.291
.329
112.8
GC
Test Instances
Found
truck
truck
truck
truck
$\cos(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}})$
$\|\phi(\boldsymbol{z}_{\text{train}})\|$
.754
.754
.752
.0008

Figure 5: Examples of the training instances frequently found to be relevant with multiple test instances on CIFAR10 with CNN, the cosine between them, and the norm of the training instances.

$p(y \mid \boldsymbol{x}; \boldsymbol{\theta})$ is given by the y-th entry of $\sigma(W\boldsymbol{x})$, where σ is the softmax function, $\boldsymbol{\theta} = W \in \mathbb{R}^{C \times d}$, and C and d are the number of classes and the dimensionality of $\boldsymbol{x}$, respectively. With some algebra, we obtain $R_{\text{GC}}(\boldsymbol{z}, \boldsymbol{z}') = \cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'}) \cos(\boldsymbol{x}, \boldsymbol{x}')$ for $\boldsymbol{z} = (\boldsymbol{x}, y)$ and $\boldsymbol{z}' = (\boldsymbol{x}', y')$, where $\boldsymbol{r}^{\boldsymbol{z}} = \sigma(W\boldsymbol{x}) - \boldsymbol{e}_y$ is the *residual* for the prediction on $\boldsymbol{z}$ and $\boldsymbol{e}_y$ is a vector whose y-th entry is one and zero otherwise. Here, the term $\cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'})$ plays an essential role in GC. From the definition, $r_c^{\boldsymbol{z}} \leq 0$ if $c = y$ and $r_c^{\boldsymbol{z}} \geq 0$ otherwise. Thus, $\cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'}) \geq 0$ always holds true when $y = y'$, while $\cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'})$ can be negative for $y \neq y'$. Hence, the chance of $R_{\text{GC}}(\boldsymbol{z}, \boldsymbol{z}')$ being positive can be larger for the instances from the same class compared to the ones from a different class.

Figure 6 shows that $\cos(\boldsymbol{r}^{\boldsymbol{z}}, \boldsymbol{r}^{\boldsymbol{z}'})$ is essential also for deep neural networks. Here, for each test instance $\boldsymbol{z}_{\text{test}}$ on CIFAR10 with CNN, we randomly sampled two training instances $\boldsymbol{z}_{\text{train}}$, one with the same class and the other with a different class, and computed $R_{\text{GC}}(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}})$ and $\cos(\boldsymbol{r}^{\boldsymbol{z}_{\text{test}}}, \boldsymbol{r}^{\boldsymbol{z}_{\text{train}}})$.

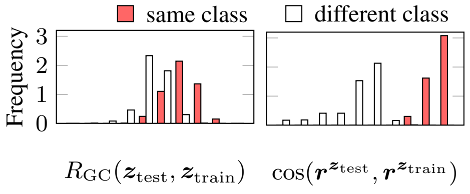


Figure 6: Distributions of $R_{\text{GC}}(\boldsymbol{z}_{\text{test}}, \boldsymbol{z}_{\text{train}})$ and $\cos(\boldsymbol{r}^{\boldsymbol{z}_{\text{test}}}, \boldsymbol{r}^{\boldsymbol{z}_{\text{train}}})$ for training instances with the same / different classes on CIFAR10 with CNN.

We also note that $\cos(\boldsymbol{r}^{\boldsymbol{z}_{\text{test}}}, \boldsymbol{r}^{\boldsymbol{z}_{\text{train}}})$ alone was not helpful for the identical subclass test, whose success rate was around the chance level. We thus conjecture that, while $\cos(\boldsymbol{r}^{\boldsymbol{z}_{\text{test}}}, \boldsymbol{r}^{\boldsymbol{z}_{\text{train}}})$ is particularly helpful for the identical class test, the use of the entire gradient is still essential for GC to meet our evaluation criteria.

6 Conclusion

We proposed four criteria to evaluate relevance metrics for instance-based explanation: two criteria for sanity checks, and the other two criteria for evaluating the practical utilities of relevance metrics. Our quantitative evaluations based on these criteria revealed that some popular relevance metrics do not even meet the minimal requirements for the instance-based explanation; thus, the use of such metrics would not be appropriate for instance-based explanation. The results also suggest, as Adebayo et al. [2018] have demonstrated for saliency methods, that quantitative evaluation is essential for instance-based explanation. Further investigation of evaluation criteria that meet the practical requirements remain as an important future direction. Designing better relevance metrics based on the evaluation criteria would be essential as well.

Broader Impact

Interpretable machine learning and explainable AI technologies are studied extensively in recent years [Ribeiro et al., 2016, Lundberg and Lee, 2017, Guidotti et al., 2018, Adadi and Berrada, 2018, Molnar, 2020]. Researchers aim to realize a society where humans and machine learning systems can coexist better, by making models less black-box and promoting users' trust to the systems. However, recent studies [Adebayo et al., 2018, Rudin, 2019, Aïvodji et al., 2019] revealed some negative aspects of these researches, e.g., some of the methods can mislead the users by providing wrong explanations. This paper follows these lines of researches, and we tried to reveal possible negative aspects of current instance-based explanation methods.

As a positive aspect, we expect our study to prevent people from using inappropriate instance-based explanation methods. This will save people from making wrong decisions mislead by wrong explanations and subsequent various damages, such as health problems and economic losses.

A possible downside, on the other hand, is that our study might discourage researchers working on completely new idea for relevance metrics. Some novel idea might be dismissed just because they failed to pass our sanity checks. We would like to remind that this is not what we desire. The relevance metrics can be used for applications other than instance-based explanation as well. Data cleansing that we mentioned in Section 2 is one such application. We expect that our evaluation criteria to be used appropriately in the context of instance-based explanation, but not for rejecting novel ideas regardless of the contexts.

Acknowledgement

We thank Dr. Ryo Karakida and Dr. Takanori Maehara for their helpful advice. We also thank Overfit Summer Seminar[5] for an opportunity that inspired this research. We also thank our laboratory members for their helpful comments. Customer Feedback Analysis dataset is obtained through personal communication with Dr. Alberto Poncelas. Satoshi Hara is supported by JSPS KAKENHI Grant Number JP20K19860.

References

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You" Explaining the Predictions of Any Classifier. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, 2016.

Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774, 2017.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51 (5):1–42, 2018.

Amina Adadi and Mohammed Berrada. Peeking Inside the Black-box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

Christoph Molnar. *Interpretable Machine Learning*. Lulu. com, 2020.

Gary A Klein and Roberta Calderwood. How Do People Use Analogues to Make Decisions? In *Proceedings of the DARPA Workshop on Case-Based Reasoning, 1988*, pages 209–223, 1988.

[5]`https://sites.google.com/view/mimaizumi/event/mlcamp2018`

Gary A Klein. Strategies of Decision Making. Technical report, 1989.

Stephen J Read and Ian L Cesa. This Reminds Me of the Time When…: Expectation Failures in Reminding and Explanation. *Journal of Experimental Social Psychology*, 27(1):1–25, 1991.

Pádraig Cunningham, Dónal Doyle, and John Loughrey. An Evaluation of the Usefulness of Case-Based Explanation. In *International Conference on Case-Based Reasoning*, pages 122–130. Springer, 2003.

Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1885–1894, 2017.

Rajiv Khanna, Been Kim, Joydeep Ghosh, and Sanmi Koyejo. Interpreting Black Box Predictions using Fisher Kernels. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, volume 89, pages 3382–3390, 2019.

Rich Caruana, Hooshang Kangarloo, John David N. Dionisio, Usha Sinha, and David Johnson. Case-Based Explanation of Non-Case-Based Learning Methods. In *Proceedings of the AMIA Symposium*, pages 212–215, 1999.

Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples Are Not Enough, Learn to Criticize! Criticism for Interpretability. In *Advances in neural information processing systems 29*, pages 2280–2288, 2016.

Been Kim, Cynthia Rudin, and Julie A Shah. The Bayesian Case Model: A Generative Approach for Case-Based Reasoning and Prototype Classification. In *Advances in Neural Information Processing Systems 27*, pages 1952–1960, 2014.

Tobias Plötz and Stefan Roth. Neural Nearest Neighbors Networks. In *Advances in Neural Information Processing Systems 31*, pages 1087–1098. 2018.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This Looks Like That: Deep Learning for Interpretable Image Recognition. In *Advances in Neural Information Processing Systems 32*, pages 8930–8941. 2019.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity Checks for Saliency Maps. In *Advances in Neural Information Processing Systems 31*, pages 9505–9515. 2018.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

Haneen Arafat Abu Alfeilat, Ahmad B.A. Hassanat, Omar Lasassmeh, Ahmad S. Tarawneh, Mahmoud Bashir Alhasanat, Hamzeh S. Eyal Salman, and V.B. Surya Prasath. Effects of Distance Measure Choice on K-Nearest Neighbor Classifier Performance: A Review. *Big Data*, 7(4): 221–248, 2019.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. 2013.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Lo\"\ic Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, 2017.

Bernhard Schölkopf, Alexander J Smola, and Francis Bach. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working Set Selection Using Second Order Information for Training Support Vector Machines. *Journal of Machine Learning Research*, 6: 1889–1918, 2005.

Jacob Bien and Robert Tibshirani. Prototype Selection for Interpretable Classification. *Annals of Applied Statistics*, 5(4):2403–2424, 2011.

Chih-Kuan Yeh, Joon Kim, Ian En-Hsu Yen, and Pradeep K Ravikumar. Representer Point Selection for Explaining Deep Neural Networks. In *Advances in Neural Information Processing Systems 31*, pages 9291–9301, 2018.

Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. Input Similarity from the Neural Network Perspective. In *Advances in Neural Information Processing Systems 32*, pages 5342–5351. 2019.

Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé Poirier. Large-Scale Image Retrieval With Compressed Fisher Vectors. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3384–3391, 2010.

Muhammad Hussain, Summrina Kanwal Wajid, Ali Elzaart, and Mohammed Berbar. A Comparison of SVM Kernel Functions for Breast Cancer Detection. In *Proceedings of the 8th International Conference on Computer Graphics, Imaging and Visualization*, pages 145–150, 2011.

Li Yu Hu, Min Wei Huang, Shih Wen Ke, and Chih Fong Tsai. The Distance Function Effect on k-Nearest Neighbor Classification for Medical Datasets. *SpringerPlus*, 5(1):1304, 2016.

Zhou Li and Chunxiang Li. Selection of Kernel Function for Least Squares Support Vector Machines in Downburst Wind Speed Forecasting. In *Proceedings of the 11th International Symposium on Computational Intelligence and Design*, volume 2, pages 337–341, 2018.

Satoshi Hara, Atsushi Nitanda, and Takanori Maehara. Data Cleansing for Models Trained with SGD. In *Advances in Neural Information Processing Systems 32*, pages 4213–4222. 2019.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. pages 248–255. Institute of Electrical and Electronics Engineers (IEEE), 2010.

George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11): 39–41, 1995.

Xin Li and Dan Roth. Learning Question Classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics*, 2002.

Chao-Hong Liu, Yasufumi Moriya, Alberto Poncelas, and Declan Groves. IJCNLP-2017 Task 4: Customer Feedback Analysis. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*, pages 26–33, 2017.

Cynthia Rudin. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, 1(5):206, 2019.

Ulrich Aïvodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. Fairwashing: the risk of rationalization. In *Proceedings of the 36th International Conference on Machine Learning*, pages 161–170, 2019.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv preprint:1802.03426*, 2018.

A Gradient-based Metrics

In gradient-based metrics, we consider a model with parameter $\boldsymbol{\theta}$, its loss $\ell(\boldsymbol{z};\boldsymbol{\theta})$, and its gradient $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z};\boldsymbol{\theta})$ to measure relevance, where $\boldsymbol{z}=(\boldsymbol{x},y)$ is an input-output pair.

Influence Function [Koh and Liang, 2017] Koh and Liang [2017] proposed to measure relevance according to "how largely the test loss will increase if the training instance is omitted from the training set." Here, the model parameter trained using all of the training set is denoted by $\widehat{\boldsymbol{\theta}}$, and the parameter trained using all of the training set except the i-th instance $\boldsymbol{z}_{\text{train}}^{(i)}$ is denoted by $\widehat{\boldsymbol{\theta}}_{-i}$. The relevance metric proposed by Koh and Liang [2017] is then defined as the difference between the test loss under parameters $\widehat{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{\theta}}_{-i}$ as follows:

$$R_{\text{IF}}(\boldsymbol{z}_{\text{test}},\boldsymbol{z}_{\text{train}}^{(i)}) := \ell(\boldsymbol{z}_{\text{test}};\widehat{\boldsymbol{\theta}}_{-i}) - \ell(\boldsymbol{z}_{\text{test}};\widehat{\boldsymbol{\theta}}). \tag{4}$$

Here, a greater value indicates that the loss on the test instance increases drastically by removing the i-th training instance from the training set. Thus, the i-th training instance is essential relative to predicting the test instance; therefore, it is highly relevant.

In practice, the following approximation is used to avoid computing $\widehat{\boldsymbol{\theta}}_{-i}$ explicitly.

$$R_{\text{IF}}(\boldsymbol{z}_{\text{test}},\boldsymbol{z}_{\text{train}}^{(i)}) \approx \langle \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\text{test}};\widehat{\boldsymbol{\theta}}), \boldsymbol{H}^{-1}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\text{train}}^{(i)};\widehat{\boldsymbol{\theta}}))\rangle, \tag{5}$$

where $\boldsymbol{H}$ is the Hessian matrix of the loss $\mathcal{L}_{\text{train}}$.

Fisher Kernel [Khanna et al., 2019] Khanna et al. [2019] proposed to measure the relevance of instances using the Fisher kernel as follows:

$$R_{\text{FK}}(\boldsymbol{z}_{\text{test}},\boldsymbol{z}_{\text{train}}^{(i)}) := \langle \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\text{test}};\widehat{\boldsymbol{\theta}}), \boldsymbol{I}^{-1}\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\text{train}}^{(i)};\widehat{\boldsymbol{\theta}})\rangle, \tag{6}$$

where $\boldsymbol{I}$ is the Fisher information matrix of the loss $\mathcal{L}_{\text{train}}$.

Grad-Dot, Grad-Cos [Perronnin et al., 2010, Yeh et al., 2018, Charpiat et al., 2019] Charpiat et al. [2019] proposed to measure relevance according to "how largely the loss will decrease when a small update is added to the model using the training instance." This can be computed as the dot product of the loss gradients, which we refer to as Grad-Dot.

$$R_{\text{GD}}(\boldsymbol{z}_{\text{test}},\boldsymbol{z}_{\text{train}}) := \langle \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\text{test}};\widehat{\boldsymbol{\theta}}), \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\text{train}};\widehat{\boldsymbol{\theta}})\rangle. \tag{7}$$

Note that a similar metric is studied by Yeh et al. [2018] as the *representer point value*.

As a modification of Grad-Dot, Charpiat et al. [2019] also proposed the following cosine version, which we refer to as Grad-Cos.

$$R_{\text{GC}}(\boldsymbol{z}_{\text{test}},\boldsymbol{z}_{\text{train}}) := \cos(\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\text{test}};\widehat{\boldsymbol{\theta}}), \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{z}_{\text{train}};\widehat{\boldsymbol{\theta}})). \tag{8}$$

Note that the use of the cosine between the gradients is also proposed by Perronnin et al. [2010].

B Experimental Setup

B.1 Datasets and Models

MNIST [LeCun et al., 1998] The MNIST dataset is used for handwritten digit image classification tasks. Here, input $\boldsymbol{x}$ is an image of a handwritten digit, and the output y consists of 10 classes ("0" to "9"). We adopted logistic regression and a CNN as the classification models. The CNN has six convolutional layers, and max-pooling layers for each two convolutional layers. The features obtained by these layers are fed into the global average pooling layer followed by a single linear layer. The number of the output channels of all the convolutional layers is set to 16. We trained the models using the Adam optimizer with a learning rate of 0.001. We used randomly sampled 5,500 training instances to train the models.

CIFAR10 [Krizhevsky, 2009] The CIFAR10 dataset is used for object recognition tasks. Here, input $\boldsymbol{x}$ is an image containing a certain object, and output y consists of 10 classes, e.g., "bird" or "airplane." Note that we used the same models as for the MNIST dataset. In addition, we adopted MobileNetV2 [Sandler et al., 2018] as a model with a higher performance than the previous model. We trained the models using the Adam optimizer with a learning rate of 0.001. In the experiments, we first pre-trained the models using all the training instances of CIFAR10, and then trained the models using randomly sampled 5,000 training instances. Without the pre-training, the classification performance of the models dropped significantly.

Note that we did not examine IF and FK on MobileNetV2 because the matrix inverse in these metrics required too much time to calculate even with the conjugate gradient approximation proposed by Koh and Liang [2017].

TREC [Li and Roth, 2002] The TREC dataset is used for question classification tasks. Here, input $\boldsymbol{x}$ is a question sentence, and output y is a question category consisting of six classes, e.g., "LOC" and "NUM." We used bag-of-words logistic regression and a two-layer Bi-LSTM as the classification models. In the Bi-LSTM, the last state is fed into one linear layer. The word embedding dimension is set to 16, and the dimension of the LSTM is set to 16 also. We trained the models using the Adam optimizer with a learning rate of 0.001. We used randomly sampled 2,726 training instances to train the models.

Customer Feedback (CF) [Liu et al., 2017] The Customer Feedback (CF) dataset is used for sentence classification tasks comprising multiple languages. Here, input $\boldsymbol{x}$ is a sentence, and output y is a category comprising six classes, e.g., "comment" and "request." We used the same models as TREC. We trained the models using the Adam optimizer with a learning rate of 0.001. We used randomly sampled 4,088 training instances to train the models.

B.2 Computing Infrastructure

In our experiments, training of the models was run on a NVIDIA GTX 1080 GPU with Intel Xeon Silver 4112 CPU and 64GB RAM. Testing and computing relevance metrics were run on Xeon E5-2680 v2 CPU with 256GB RAM.

C Repairing Gradient-based Metrics

As described in Section 5, we found that training instances with extremely large norms were selected as relevant by IF, FK, and GD. Thus, to repair these metrics, we need to design metrics that can ignore instances with large norms. A simple yet effective way of repairing the metrics is to use ℓ_2 or cosine instead of the dot product. As Figure 2 shows, the ℓ_2 and cosine metrics performed better than the dot metrics. Indeed, the ℓ_2 metrics do not favor instances with large norms that lead to large ℓ_2-distance, and, through normalization, the cosine metrics completely ignore the effect of the norms

We name the repaired metrics of IF, FK, and GD based on the ℓ_2 metric as ℓ_2^{IF}, ℓ_2^{FK}, and ℓ_2^{GD}, respectively, and the repaired metrics based on the cosine metric as $\cos^{\mathrm{IF}}$ and $\cos^{\mathrm{FK}}$, and $\cos^{\mathrm{GD}}$, respectively[6]. We observed that these repaired metrics attained higher success rates on several evaluation criteria. The details of the results can be found in Appendix E.

D Do the models capture subclasses?

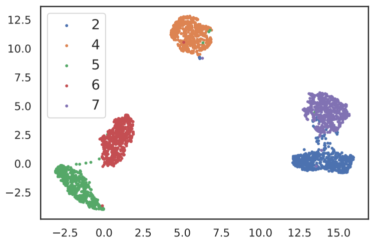
The identical subclass test requires the model to obtain internal representations that can distinguish subclasses. Here, we confirm that this condition is satisfied for all the datasets and models we used in the experiments. We consider that the model captures the subclasses if the latent representation $\boldsymbol{h}^{\mathrm{all}}$ has cluster structures. Figure 8 visualizes $\boldsymbol{h}^{\mathrm{all}}$ for each dataset and model using UMAP [McInnes et al., 2018]. The figures show that the instances from different subclasses are not mixed completely random. MNIST and TREC have relatively clear cluster structures, while CIFAR10 and CF have vague clusters without explicit boundaries. These figures imply that the models capture subclases (although it may not be perfect).

[6]Note that $\cos^{\mathrm{GD}}$ is the same as GC.

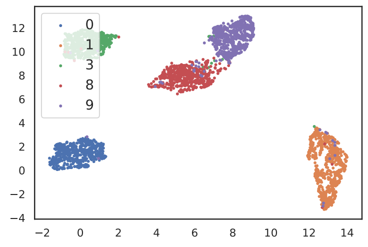
E Complete Evaluation Results

We show the complete results of the identical instance test in Table 3, the identical class test in Table 4, the top-10 identical class test in Table 5, the identical subclass test in Table 6, and the top-10 identical subclass test in Table 7. The results we present here are consistent with our observations in Section 4.

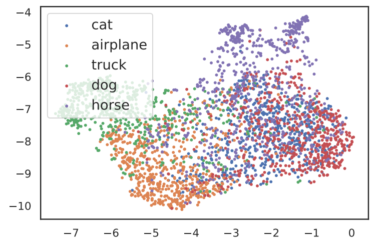
- Five metrics, ℓ_2^{last}, ℓ_2^{all}, $\cos^{\mathrm{last}}$, $\cos^{\mathrm{all}}$, and GC, passed both the identical instance test and the identical class test on all the datasets. The other metrics failed to pass at least one of the tests.
- The dot metrics and the gradient-based metrics, except for GC, failed the identical instance test.
- The metrics with the identity feature map ℓ_2^x, $\cos^x$, and dot^x failed the identical class test.
- GC outperformed the other metrics on the top-10 identical class test.
- GC and $\cos^{\mathrm{all}}$ performed well on the identical subclass test, where GC performed particularly well on the text datasets, while $\cos^{\mathrm{all}}$ performed well on the image datasets.



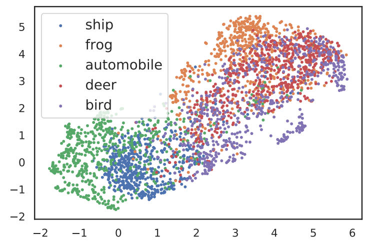
(a) MNIST with CNN. $y = \mathrm{A}$.



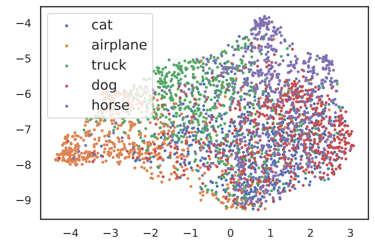
(b) MNIST with CNN. $y = \mathrm{B}$.



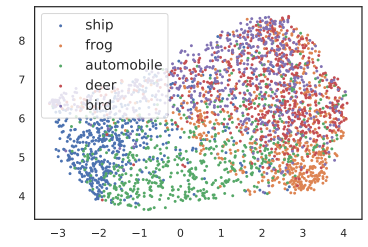
(c) CIFAR10 with MobileNetV2. $y = \mathrm{A}$.



(d) CIFAR10 with MobileNetV2. $y = \mathrm{B}$.



(e) CIFAR10 with CNN. $y = \mathrm{A}$.



(f) CIFAR10 with CNN. $y = \mathrm{B}$.

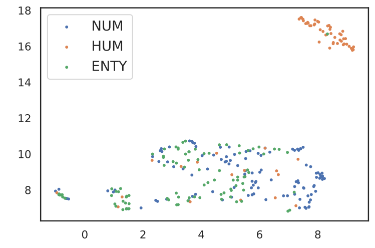
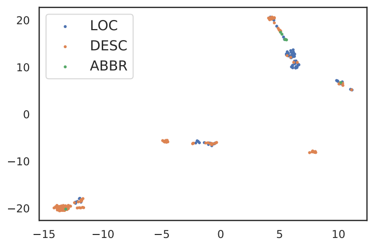
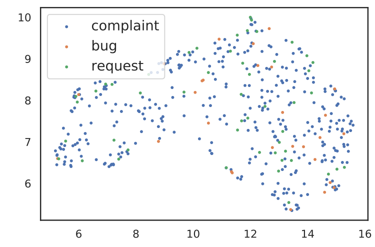


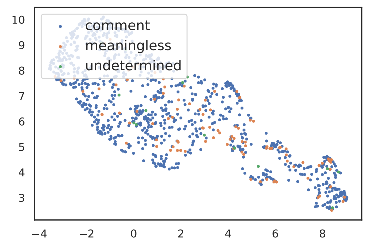
Figure 7: TREC with LSTM. $y = \mathrm{A}$.



(a) TREC with LSTM. $y = \mathrm{B}$.



(b) CF with LSTM. $y = \mathrm{A}$.



(c) CF with LSTM. $y = \mathrm{B}$.

Figure 8: visualization of $\boldsymbol{h}^{\text{all}}$ in each dataset and model using UMAP.

Table 3: Average success rate $\pm$ std. of each relevancy metric for identical instance test. The metrics prefixed with $\diamondsuit$ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.98 ± 0.00	0.92 ± 0.00	0.89 ± 0.01	0.72 ± 0.02	0.35 ± 0.01
ℓ_2^x	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
ℓ_2^{last}	1.00 ± 0.00	-	1.00 ± 0.00	1.00 ± 0.00	-
ℓ_2^{all}	1.00 ± 0.00	-	1.00 ± 0.00	1.00 ± 0.00	-
$\cos^x$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
$\cos^{\text{last}}$	1.00 ± 0.00	-	1.00 ± 0.00	1.00 ± 0.00	-
$\cos^{\text{all}}$	1.00 ± 0.00	-	1.00 ± 0.00	1.00 ± 0.00	-
dot^x	0.26 ± 0.03	0.28 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
dot^{last}	0.00 ± 0.00	-	0.00 ± 0.00	0.00 ± 0.00	-
dot^{all}	0.39 ± 0.04	-	1.00 ± 0.00	0.58 ± 0.04	-
IF	0.02 ± 0.01	0.36 ± 0.04	-	0.00 ± 0.00	0.35 ± 0.03
$\diamondsuit\ \ell_2^{\text{IF}}$	1.00 ± 0.00	1.00 ± 0.00	-	1.00 ± 0.00	1.00 ± 0.00
$\diamondsuit\ \cos^{\text{IF}}$	1.00 ± 0.00	1.00 ± 0.00	-	1.00 ± 0.00	1.00 ± 0.00
FK	0.00 ± 0.00	0.00 ± 0.00	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit\ \ell_2^{\text{FK}}$	1.00 ± 0.00	1.00 ± 0.00	-	1.00 ± 0.00	1.00 ± 0.00
$\diamondsuit\ \cos^{\text{FK}}$	1.00 ± 0.00	1.00 ± 0.00	-	1.00 ± 0.00	1.00 ± 0.00
GD	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.01	0.03 ± 0.01	0.00 ± 0.00
GC	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
$\diamondsuit\ \ell_2^{\text{grad}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

	TREC		CF	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg
Parameter size	20K	7K	51K	14K
Accuracy	0.86 ± 0.01	0.81 ± 0.02	0.71 ± 0.03	0.73 ± 0.01
ℓ_2^x	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
ℓ_2^{last}	1.00 ± 0.00	-	1.00 ± 0.00	-
ℓ_2^{all}	1.00 ± 0.00	-	1.00 ± 0.00	-
$\cos^x$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
$\cos^{\text{last}}$	1.00 ± 0.00	-	1.00 ± 0.00	-
$\cos^{\text{all}}$	1.00 ± 0.00	-	1.00 ± 0.00	-
dot^x	0.62 ± 0.03	0.61 ± 0.03	0.33 ± 0.02	0.32 ± 0.04
dot^{last}	0.07 ± 0.01	-	0.06 ± 0.01	-
dot^{all}	0.14 ± 0.03	-	0.13 ± 0.02	-
IF	0.00 ± 0.00	0.52 ± 0.03	0.00 ± 0.00	0.62 ± 0.04
$\diamondsuit\ \ell_2^{\text{IF}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
$\diamondsuit\ \cos^{\text{IF}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
FK	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit\ \ell_2^{\text{FK}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
$\diamondsuit\ \cos^{\text{FK}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
GD	0.00 ± 0.00	0.04 ± 0.02	0.00 ± 0.00	0.11 ± 0.02
GC	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
$\diamondsuit\ \ell_2^{\text{grad}}$	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00

Table 4: Average success rate $\pm$ std. of each relevancy metric for identical class test. The metrics prefixed with $\diamondsuit$ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.98 ± 0.00	0.92 ± 0.00	0.89 ± 0.01	0.72 ± 0.02	0.35 ± 0.01
ℓ_2^x	0.93 ± 0.01	0.88 ± 0.01	0.26 ± 0.02	0.26 ± 0.02	0.24 ± 0.02
ℓ_2^{last}	0.99 ± 0.01	-	1.00 ± 0.00	0.75 ± 0.02	-
ℓ_2^{all}	0.98 ± 0.00	-	0.93 ± 0.02	0.61 ± 0.02	-
$\cos^x$	0.94 ± 0.01	0.88 ± 0.01	0.30 ± 0.03	0.29 ± 0.02	0.26 ± 0.02
$\cos^{\text{last}}$	0.99 ± 0.01	-	1.00 ± 0.00	0.78 ± 0.02	-
$\cos^{\text{all}}$	0.98 ± 0.00	-	0.97 ± 0.01	0.71 ± 0.02	-
dot^x	0.69 ± 0.02	0.68 ± 0.02	0.09 ± 0.02	0.10 ± 0.01	0.11 ± 0.02
dot^{last}	0.67 ± 0.02	-	1.00 ± 0.00	0.20 ± 0.02	-
dot^{all}	0.96 ± 0.01	-	0.96 ± 0.01	0.31 ± 0.01	-
IF	0.09 ± 0.01	0.26 ± 0.02	-	0.10 ± 0.01	0.09 ± 0.02
$\diamondsuit$ ℓ_2^{IF}	0.72 ± 0.01	0.62 ± 0.02	-	0.14 ± 0.01	0.14 ± 0.01
$\diamondsuit$ $\cos^{\text{IF}}$	0.82 ± 0.01	0.69 ± 0.02	-	0.12 ± 0.01	0.13 ± 0.02
FK	0.10 ± 0.01	0.21 ± 0.02	-	0.20 ± 0.02	0.20 ± 0.02
$\diamondsuit$ ℓ_2^{FK}	0.77 ± 0.02	0.93 ± 0.01	-	0.82 ± 0.01	0.98 ± 0.00
$\diamondsuit$ $\cos^{\text{FK}}$	0.92 ± 0.01	0.97 ± 0.01	-	0.93 ± 0.01	0.99 ± 0.00
GD	0.30 ± 0.01	0.87 ± 0.01	0.26 ± 0.03	0.71 ± 0.02	1.00 ± 0.00
GC	0.99 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	1.00 ± 0.00
$\diamondsuit$ ℓ_2^{grad}	0.94 ± 0.01	0.99 ± 0.00	0.97 ± 0.01	0.99 ± 0.00	1.00 ± 0.00

	TREC		CF	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg
Parameter size	20K	7K	51K	14K
Accuracy	0.86 ± 0.01	0.81 ± 0.02	0.71 ± 0.03	0.73 ± 0.01
ℓ_2^x	0.70 ± 0.00	0.75 ± 0.00	0.56 ± 0.01	0.56 ± 0.02
ℓ_2^{last}	0.89 ± 0.00	-	0.87 ± 0.01	-
ℓ_2^{all}	0.88 ± 0.00	-	0.88 ± 0.01	-
$\cos^x$	0.73 ± 0.00	0.76 ± 0.00	0.62 ± 0.01	0.64 ± 0.01
$\cos^{\text{last}}$	0.89 ± 0.00	-	0.88 ± 0.01	-
$\cos^{\text{all}}$	0.90 ± 0.00	-	0.90 ± 0.01	-
dot^x	0.33 ± 0.00	0.34 ± 0.00	0.46 ± 0.01	0.47 ± 0.02
dot^{last}	0.93 ± 0.00	-	0.92 ± 0.01	-
dot^{all}	0.93 ± 0.00	-	0.91 ± 0.01	-
IF	0.29 ± 0.00	0.86 ± 0.00	0.08 ± 0.01	0.81 ± 0.01
$\diamondsuit$ ℓ_2^{IF}	0.98 ± 0.00	0.95 ± 0.00	0.99 ± 0.00	0.95 ± 0.01
$\diamondsuit$ $\cos^{\text{IF}}$	0.99 ± 0.00	0.96 ± 0.00	0.99 ± 0.00	0.98 ± 0.01
FK	0.28 ± 0.00	0.24 ± 0.00	0.30 ± 0.02	0.29 ± 0.01
$\diamondsuit$ ℓ_2^{FK}	0.99 ± 0.00	0.96 ± 0.00	0.92 ± 0.01	0.96 ± 0.01
$\diamondsuit$ $\cos^{\text{FK}}$	1.00 ± 0.00	0.96 ± 0.00	0.93 ± 0.01	0.98 ± 0.00
GD	0.49 ± 0.00	1.00 ± 0.00	0.76 ± 0.01	1.00 ± 0.00
GC	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	1.00 ± 0.00
$\diamondsuit$ ℓ_2^{grad}	1.00 ± 0.00	1.00 ± 0.00	0.99 ± 0.00	1.00 ± 0.00

Table 5: Average success rate ± std. of each relevancy metric for top-10 identical class test. The metrics prefixed with ♢ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.98 ± 0.00	0.92 ± 0.00	0.89 ± 0.01	0.72 ± 0.02	0.35 ± 0.01
ℓ_2^x	0.63 ± 0.02	0.63 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
ℓ_2^{last}	0.95 ± 0.01	-	0.98 ± 0.01	0.30 ± 0.01	-
ℓ_2^{all}	0.89 ± 0.01	-	0.64 ± 0.05	0.14 ± 0.01	-
$\cos^x$	0.67 ± 0.02	0.65 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
$\cos^{\text{last}}$	0.97 ± 0.01	-	0.98 ± 0.01	0.33 ± 0.02	-
$\cos^{\text{all}}$	0.92 ± 0.00	-	0.84 ± 0.03	0.23 ± 0.02	-
dot^x	0.19 ± 0.01	0.20 ± 0.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
dot^{last}	0.42 ± 0.03	-	0.98 ± 0.01	0.04 ± 0.01	-
dot^{all}	0.88 ± 0.01	-	0.79 ± 0.03	0.05 ± 0.01	-
IF	0.00 ± 0.00	0.00 ± 0.00	-	0.00 ± 0.00	0.00 ± 0.00
♢ ℓ_2^{IF}	0.25 ± 0.01	0.10 ± 0.01	-	0.00 ± 0.00	0.00 ± 0.00
♢ $\cos^{\text{IF}}$	0.59 ± 0.02	0.17 ± 0.01	-	0.00 ± 0.00	0.00 ± 0.00
FK	0.00 ± 0.00	0.02 ± 0.01	-	0.00 ± 0.00	0.06 ± 0.01
♢ ℓ_2^{FK}	0.23 ± 0.03	0.65 ± 0.02	-	0.25 ± 0.02	0.87 ± 0.01
♢ $\cos^{\text{FK}}$	0.59 ± 0.01	0.82 ± 0.02	-	0.54 ± 0.02	0.93 ± 0.01
GD	0.00 ± 0.00	0.41 ± 0.02	0.00 ± 0.00	0.15 ± 0.01	1.00 ± 0.00
GC	0.95 ± 0.01	0.99 ± 0.01	0.92 ± 0.02	0.92 ± 0.01	1.00 ± 0.00
♢ ℓ_2^{grad}	0.57 ± 0.02	0.95 ± 0.01	0.78 ± 0.03	0.80 ± 0.01	0.99 ± 0.00

	TREC		CF	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg
Parameter size	20K	7K	51K	14K
Accuracy	0.86 ± 0.01	0.81 ± 0.02	0.71 ± 0.03	0.73 ± 0.01
ℓ_2^x	0.23 ± 0.00	0.23 ± 0.00	0.04 ± 0.01	0.04 ± 0.01
ℓ_2^{last}	0.68 ± 0.00	-	0.64 ± 0.01	-
ℓ_2^{all}	0.66 ± 0.00	-	0.64 ± 0.02	-
$\cos^x$	0.24 ± 0.00	0.24 ± 0.00	0.10 ± 0.01	0.10 ± 0.01
$\cos^{\text{last}}$	0.69 ± 0.00	-	0.65 ± 0.02	-
$\cos^{\text{all}}$	0.68 ± 0.00	-	0.67 ± 0.02	-
dot^x	0.05 ± 0.00	0.05 ± 0.00	0.04 ± 0.01	0.04 ± 0.01
dot^{last}	0.75 ± 0.00	-	0.81 ± 0.01	-
dot^{all}	0.84 ± 0.00	-	0.79 ± 0.02	-
IF	0.00 ± 0.00	0.24 ± 0.00	0.00 ± 0.00	0.04 ± 0.01
♢ ℓ_2^{IF}	0.83 ± 0.00	0.47 ± 0.00	0.93 ± 0.01	0.53 ± 0.02
♢ $\cos^{\text{IF}}$	0.91 ± 0.00	0.65 ± 0.00	0.94 ± 0.01	0.64 ± 0.02
FK	0.01 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
♢ ℓ_2^{FK}	0.90 ± 0.00	0.71 ± 0.00	0.66 ± 0.01	0.74 ± 0.02
♢ $\cos^{\text{FK}}$	0.95 ± 0.00	0.77 ± 0.00	0.64 ± 0.01	0.81 ± 0.01
GD	0.11 ± 0.00	0.99 ± 0.00	0.26 ± 0.01	0.99 ± 0.00
GC	0.96 ± 0.00	1.00 ± 0.00	0.97 ± 0.00	1.00 ± 0.00
♢ ℓ_2^{grad}	0.94 ± 0.00	1.00 ± 0.00	0.96 ± 0.00	0.98 ± 0.00

Table 6: Average success rate $\pm$ std. of each relevancy metric for identical subclass test. The metrics prefixed with $\diamondsuit$ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.99 ± 0.00	0.88 ± 0.01	0.92 ± 0.01	0.84 ± 0.03	0.71 ± 0.03
ℓ_2^x	0.93 ± 0.01	0.96 ± 0.02	0.26 ± 0.02	0.29 ± 0.04	0.31 ± 0.03
ℓ_2^{last}	0.89 ± 0.02	-	0.29 ± 0.04	0.35 ± 0.04	-
ℓ_2^{all}	0.97 ± 0.01	-	0.49 ± 0.04	0.38 ± 0.03	-
$\cos^x$	0.95 ± 0.01	0.96 ± 0.02	0.29 ± 0.03	0.31 ± 0.04	0.31 ± 0.03
$\cos^{\text{last}}$	0.89 ± 0.02	-	0.32 ± 0.03	0.33 ± 0.03	-
$\cos^{\text{all}}$	0.98 ± 0.00	-	0.71 ± 0.04	0.50 ± 0.03	-
dot^x	0.70 ± 0.03	0.75 ± 0.03	0.09 ± 0.02	0.11 ± 0.03	0.09 ± 0.02
dot^{last}	0.24 ± 0.04	-	0.22 ± 0.02	0.20 ± 0.01	-
dot^{all}	0.94 ± 0.01	-	0.68 ± 0.03	0.25 ± 0.03	-
IF	0.12 ± 0.01	0.39 ± 0.05	-	0.06 ± 0.02	0.08 ± 0.02
$\diamondsuit$ ℓ_2^{IF}	0.62 ± 0.04	0.76 ± 0.03	-	0.17 ± 0.02	0.12 ± 0.02
$\diamondsuit$ $\cos^{\text{IF}}$	0.70 ± 0.02	0.87 ± 0.02	-	0.15 ± 0.02	0.09 ± 0.02
FK	0.19 ± 0.03	0.14 ± 0.02	-	0.11 ± 0.01	0.11 ± 0.02
$\diamondsuit$ ℓ_2^{FK}	0.81 ± 0.02	0.76 ± 0.03	-	0.31 ± 0.03	0.24 ± 0.02
$\diamondsuit$ $\cos^{\text{FK}}$	0.91 ± 0.02	0.85 ± 0.02	-	0.37 ± 0.03	0.23 ± 0.02
GD	0.42 ± 0.05	0.48 ± 0.03	0.20 ± 0.02	0.24 ± 0.03	0.21 ± 0.04
GC	0.97 ± 0.01	0.98 ± 0.01	0.54 ± 0.03	0.43 ± 0.04	0.39 ± 0.03
$\diamondsuit$ ℓ_2^{grad}	0.91 ± 0.02	0.95 ± 0.01	0.28 ± 0.03	0.38 ± 0.03	0.34 ± 0.03

	TREC		CF	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg
Parameter size	20K	7K	51K	14K
Accuracy	0.90 ± 0.02	0.88 ± 0.01	0.79 ± 0.02	0.87 ± 0.01
ℓ_2^x	0.78 ± 0.03	0.78 ± 0.02	0.60 ± 0.03	0.58 ± 0.03
ℓ_2^{last}	0.76 ± 0.02	-	0.77 ± 0.02	-
ℓ_2^{all}	0.77 ± 0.03	-	0.79 ± 0.02	-
$\cos^x$	0.82 ± 0.02	0.81 ± 0.02	0.67 ± 0.02	0.66 ± 0.02
$\cos^{\text{last}}$	0.75 ± 0.02	-	0.78 ± 0.02	-
$\cos^{\text{all}}$	0.77 ± 0.02	-	0.78 ± 0.02	-
dot^x	0.33 ± 0.03	0.34 ± 0.03	0.45 ± 0.04	0.46 ± 0.04
dot^{last}	0.40 ± 0.03	-	0.78 ± 0.03	-
dot^{all}	0.59 ± 0.03	-	0.80 ± 0.03	-
IF	0.31 ± 0.02	0.49 ± 0.03	0.56 ± 0.03	0.68 ± 0.03
$\diamondsuit$ ℓ_2^{IF}	0.68 ± 0.02	0.79 ± 0.02	0.77 ± 0.02	0.78 ± 0.02
$\diamondsuit$ $\cos^{\text{IF}}$	0.72 ± 0.01	0.75 ± 0.03	0.78 ± 0.03	0.83 ± 0.03
FK	0.30 ± 0.02	0.16 ± 0.02	0.47 ± 0.03	0.31 ± 0.03
$\diamondsuit$ ℓ_2^{FK}	0.73 ± 0.03	0.78 ± 0.02	0.75 ± 0.03	0.81 ± 0.02
$\diamondsuit$ $\cos^{\text{FK}}$	0.81 ± 0.02	0.79 ± 0.01	0.80 ± 0.04	0.81 ± 0.02
GD	0.45 ± 0.02	0.60 ± 0.02	0.70 ± 0.03	0.79 ± 0.02
GC	0.81 ± 0.01	0.87 ± 0.02	0.84 ± 0.03	0.81 ± 0.02
$\diamondsuit$ ℓ_2^{grad}	0.78 ± 0.02	0.88 ± 0.02	0.81 ± 0.02	0.84 ± 0.02

Table 7: Average success rate $\pm$ std. of each relevancy metric for top-10 identical subclass test. The metrics prefixed with $\diamondsuit$ are the ones we have repaired. The results with the average success rate over 0.5 are colored.

	MNIST		CIFAR10		
Model	CNN	logreg	MobilenetV2	CNN	logreg
Parameter size	12K	8K	2.2M	12K	31K
Accuracy	0.99 ± 0.00	0.88 ± 0.01	0.92 ± 0.01	0.84 ± 0.03	0.71 ± 0.03
ℓ_2^x	0.64 ± 0.02	0.71 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
ℓ_2^{last}	0.54 ± 0.04	-	0.00 ± 0.00	0.00 ± 0.00	-
ℓ_2^{all}	0.85 ± 0.02	-	0.08 ± 0.02	0.01 ± 0.00	-
$\cos^x$	0.67 ± 0.02	0.74 ± 0.03	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00
$\cos^{\text{last}}$	0.57 ± 0.05	-	0.00 ± 0.00	0.00 ± 0.00	-
$\cos^{\text{all}}$	0.89 ± 0.02	-	0.16 ± 0.02	0.02 ± 0.01	-
dot^x	0.21 ± 0.02	0.23 ± 0.03	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
dot^{last}	0.08 ± 0.02	-	0.00 ± 0.00	0.00 ± 0.00	-
dot^{all}	0.79 ± 0.03	-	0.13 ± 0.02	0.01 ± 0.01	-
IF	0.01 ± 0.01	0.00 ± 0.00	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit\ \ell_2^{\text{IF}}$	0.14 ± 0.03	0.16 ± 0.02	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit\ \cos^{\text{IF}}$	0.37 ± 0.02	0.35 ± 0.04	-	0.00 ± 0.00	0.00 ± 0.00
FK	0.00 ± 0.00	0.00 ± 0.00	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit\ \ell_2^{\text{FK}}$	0.22 ± 0.02	0.30 ± 0.03	-	0.00 ± 0.00	0.00 ± 0.00
$\diamondsuit\ \cos^{\text{FK}}$	0.58 ± 0.02	0.46 ± 0.04	-	0.00 ± 0.00	0.00 ± 0.00
GD	0.00 ± 0.00	0.01 ± 0.01	0.01 ± 0.01	0.00 ± 0.00	0.00 ± 0.00
GC	0.86 ± 0.03	0.87 ± 0.02	0.06 ± 0.02	0.01 ± 0.01	0.01 ± 0.01
$\diamondsuit\ \ell_2^{\text{grad}}$	0.50 ± 0.03	0.69 ± 0.04	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00

	TREC		CF	
Model	Bi-LSTM	logreg	Bi-LSTM	logreg
Parameter size	20K	7K	51K	14K
Accuracy	0.90 ± 0.02	0.88 ± 0.01	0.79 ± 0.02	0.87 ± 0.01
ℓ_2^x	0.27 ± 0.05	0.25 ± 0.02	0.05 ± 0.01	0.06 ± 0.03
ℓ_2^{last}	0.30 ± 0.02	-	0.34 ± 0.04	-
ℓ_2^{all}	0.34 ± 0.02	-	0.34 ± 0.03	-
$\cos^x$	0.28 ± 0.04	0.27 ± 0.02	0.12 ± 0.02	0.12 ± 0.03
$\cos^{\text{last}}$	0.30 ± 0.02	-	0.34 ± 0.03	-
$\cos^{\text{all}}$	0.34 ± 0.02	-	0.36 ± 0.04	-
dot^x	0.05 ± 0.01	0.05 ± 0.01	0.03 ± 0.01	0.04 ± 0.02
dot^{last}	0.14 ± 0.01	-	0.56 ± 0.03	-
dot^{all}	0.17 ± 0.02	-	0.55 ± 0.03	-
IF	0.00 ± 0.00	0.01 ± 0.01	0.00 ± 0.00	0.01 ± 0.01
$\diamondsuit\ \ell_2^{\text{IF}}$	0.11 ± 0.02	0.24 ± 0.02	0.31 ± 0.02	0.30 ± 0.04
$\diamondsuit\ \cos^{\text{IF}}$	0.22 ± 0.03	0.25 ± 0.02	0.35 ± 0.03	0.38 ± 0.02
FK	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.01
$\diamondsuit\ \ell_2^{\text{FK}}$	0.28 ± 0.04	0.26 ± 0.02	0.35 ± 0.03	0.38 ± 0.03
$\diamondsuit\ \cos^{\text{FK}}$	0.41 ± 0.03	0.25 ± 0.02	0.37 ± 0.04	0.40 ± 0.03
GD	0.10 ± 0.02	0.01 ± 0.00	0.04 ± 0.02	0.21 ± 0.03
GC	0.37 ± 0.03	0.37 ± 0.02	0.39 ± 0.03	0.42 ± 0.03
$\diamondsuit\ \ell_2^{\text{grad}}$	0.24 ± 0.03	0.34 ± 0.02	0.38 ± 0.03	0.54 ± 0.02