Learning Tversky Similarity

Javad Rahnama and Eyke Hüllermeier$^{(\boxtimes)}$

Heinz Nixdorf Institute and Institute of Computer Science, Paderborn University,
Paderborn, Germany
javad.rahnama@uni-paderborn.de, eyke@upb.de

Abstract. In this paper, we advocate Tversky's ratio model as an appropriate basis for computational approaches to semantic similarity, that is, the comparison of objects such as images in a semantically meaningful way. We consider the problem of learning Tversky similarity measures from suitable training data indicating whether two objects tend to be similar or dissimilar. Experimentally, we evaluate our approach to similarity learning on two image datasets, showing that is performs very well compared to existing methods.

Keywords: Similarity · Machine learning · Semantic features · Image data

1 Introduction

Similarity is an important cognitive concept and a key notion in various branches of artificial intelligence, including case-based reasoning [26], information retrieval [5], machine learning [6], data analysis [30] and data mining [4], among others. Specified as a numerical (real-valued) function on pairs of objects, it can be applied in a rather generic way for various problems and purposes [21]. In particular, similarity is of great interest for structured objects such as text and images. In such domains, similarity measures are normally not defined by hand but learned from data, i.e., they are automatically extracted from sample data in the form of objects along with information about their similarity.

In the image domain, numerous methods of that kind have been proposed, based on different types of feature information, including visual [12,37] and semantic features [24], and exploiting different types of measurements, such as non-metric [5,15] and metric ones [19]. Most common is a geometrical model of similarity that relies on an embedding of objects as points in a suitable (often high-dimensional) vector space, in which similarity between objects is inversely related to their distance. As popular examples of (dis-)similarity measures in this field, let us mention the Euclidean and the Cosine distance [25].

Somewhat surprisingly, another concept of similarity, the one put forward by Tversky [35], appears to be less recognized in this field, in spite of its popularity in psychology and cognitive science. Tversky argued that the geometrical model of similarity is not fully compatible with human perception, and empirically

M.-J. Lesot et al. (Eds.): IPMU 2020, CCIS 1238, pp. 269–280, 2020.
https://doi.org/10.1007/978-3-030-50143-3_21

demonstrated that human similarity assessment does not obey all properties implied by this model, such as minimality, symmetry, and triangle inequality [36]. Due to its cognitive plausibility, we hypothesize that Tversky similarity could provide a suitable basis for mimicking human similarity assessment in domains such as *art images* [20]. For example, an art historian will probably find the copy of a painting more similar to the original than the original to the copy, thereby violating symmetry. Not less importantly, Tversky similarity is a potentially interpretable measure, which does not merely produce a number, but is also able to explain why two objects are deemed more or less similar. In other words, it is able to fill the *semantic gap* between the extracted visual information and a human's interpretation of an image. This feature of *explainability* is of critical importance in many applications [29]. Last but not least, the similarity also exhibits interesting theoretical properties; for example, see [9] for a recent analysis from the perspective of measurement theory.

In this paper, we elaborate on the potential of Tversky similarity as an alternative to existing measures, such as Euclidean and Cosine distance, with a specific focus on the image domain. In particular, we consider the problem of learning Tversky similarity from data, i.e., tuning its parameters on the basis of suitable training data, a problem that has not received much attention so far. As a notable exception we mention [2], where the problem of learning the importance of features (and feature combinations) for a generalized Jaccard measure (a special case of the Tversky measures) on the basis of training data is addressed, so as to achieve optimal performance in similarity-based (nearest neighbor) classification; to this end, the authors make use of stochastic optimization techniques (particle swarm optimization and differential evolution). On the other side, it is also worth noticing that the problem of learning Tversky similarity is in general different from the use of the Tversky measure as a loss function in machine learning [1,27]. Here, the measure serves as a means to accomplish a certain goal, namely to learn a classification model (e.g., a neural network) that achieves a good compromise between precision and recall, whereas is our case, the measure corresponds to the sought model itself.

The paper is organized as follows. After a brief review of related work in the next section, we recall Tversky's notion of similarity based on the ratio model in Sect. 3. In Sect. 4, we then propose a method for learning this similarity from suitable training data. Finally, we present experimental results in Sect. 5, prior to concluding the paper in Sect. 6.

2 Related Work

2.1 Image Similarity

Image similarity and its quantitative assessment in terms of similarity measures strongly depends on the image representation. Numerous approaches have been presented to extract different types of representations based on visual and semantic features of images [12,22]. Most of the state-of-the-art methods for extracting visual features are based on deep neural networks, which produce a variety of features, ranging from low level features in the early layers to more abstract features in the last layers [18,39]. In general, however, similarity is a complex concept that can not be derived from visual features alone. Therefore, some studies have exploited the use of prior knowledge as well as intermediate or high-level representation of features to capture similarity in specific applications [3,11,12].

The similarity or distance on individual features is commonly combined into an overall assessment using measures such as weighted Euclidean or weighted Cosine distance, whereas Tversky similarity is much less used in the image domain in this regard. There are, however, a few notable exceptions. For example, Tversky similarity is used in the context of image retrieval in [28] and for measuring the similarity between satellite images in [33].

2.2 Metric Learning

As already said, the notion of similarity is closely connected to the notion of distance. Whether relationships between objects are expressed in terms of similarity or distance is often a matter of taste, although small differences in the respective mathematical formalizations also exist. In the literature, distance seems to be even a bit more common than similarity. In particular, there is large body of literature on distance (metric) learning [19]. Most distance learning methods focus on tuning the weights of attributes in the Mahalanobis distance or weighted Euclidean distance. As training information, these methods typically use constraints on pairwise or relative relations among objects [7,16].

3 Tversky Similarity

Tversky suggested that humans perceive the similarity between objects based on contrasting (the measure of) those features they have in common and those on which they differ [35]. Moreover, he suggested that more attention is paid to the shared than to the distinctive features. Thus, in his (feature-matching) model, an object is represented as a set of meaningful features, and similarity is defined based on suitable set operations.

3.1 Formal Definition

More formally, consider a set of objects $\mathcal{X}$ and a finite set of features $\mathcal{F} = \{f_1, \ldots, f_m\}$. Each feature is considered as a binary predicate, i.e., a mapping $f_i : \mathcal{X} \longrightarrow \{0,1\}$, where $f_i(\boldsymbol{x}) = 1$ is interpreted as the presence of the i^{th} feature for the object $\boldsymbol{x}$, and $f_i(\boldsymbol{x}) = 0$ as the absence. Thus, each object $\boldsymbol{x} \in \mathcal{X}$ can be represented by the subset of features it exhibits:

$$F(\boldsymbol{x}) = \big\{ f_i \,|\, f_i(\boldsymbol{x}) = 1,\, i \in \{1, \ldots, m\} \big\} \subseteq \mathcal{F}$$

Tversky similarity, in the form of the so-called ratio model, is then defined as a function $S_{\alpha,\beta} : \mathcal{X}^2 \longrightarrow [0, 1]$ as follows:

$$S_{\alpha,\beta}(\boldsymbol{x}, \boldsymbol{y}) = \frac{g(\,|A \cap B|\,)}{g(\,|A \cap B|\,) + \alpha\, g(\,|A \setminus B|\,) + \beta\, g(\,|B \setminus A|\,)}\,, \tag{1}$$

where $A = F(\boldsymbol{x})$, $B = F(\boldsymbol{y})$, and g is a non-negative, increasing function $\mathbb{N}_0 \longrightarrow \mathbb{N}_0$; in the simplest case, g is the identity measuring set cardinality. According to (1), Tversky similarity puts the number of features that are shared by two objects in relation to the number of relevant features, where relevance means that a feature is present in at least one of the two objects—features that are absent from both objects are completely ignored, which is an important property of Tversky similarity, and distinguishes it from most other similarity measures.

The coefficients $0 \leq \alpha, \beta \leq 1$ in (1) control the impact of the distinctive features as well as the asymmetry of the measure. The larger these coefficients, the more important the distinctive feature are. For $\alpha = \beta$, the similarity measure is symmetric, i.e., $S_{\alpha,\beta}(\boldsymbol{x}, \boldsymbol{y}) = S_{\alpha,\beta}(\boldsymbol{y}, \boldsymbol{x})$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$, for $\alpha \neq \beta$ it is asymmetric. Important special cases include the Jaccard coefficient ($\alpha = \beta = 1$) and the Dice similarity ($\alpha = \beta = 1/2$).

3.2 Feature Weighting

The Tversky similarity (TS) measure (1) implicitly assumes that all features have the same importance, which might not always be true. In fact, g is only a function of the cardinality of feature subsets, but ignores the concrete elements of these subsets. In other words, only the number of shared and distinctive features is important, no matter what these features are.

A natural generalization of (1) is a weighted variant of Tversky similarity (WTS), in which each feature f_i is weighted by some $w_i \in [0, 1]$:

$$S_{\alpha,\beta,\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{m} w_i f_i(\boldsymbol{x}) f_i(\boldsymbol{y})}{\sum_{i=1}^{m} w_i \Big(f_i(\boldsymbol{x}) f_i(\boldsymbol{y}) + \alpha f_i(\boldsymbol{x}) \bar{f}_i(\boldsymbol{y}) + \beta \bar{f}_i(\boldsymbol{x}) f_i(\boldsymbol{y}) \Big)}\,, \tag{2}$$

with $\bar{f}_i(\cdot) = 1 - f_i(\cdot)$. Thus, in addition to α and β, this version of Tversky similarity is now parametrized by a weight vector $\boldsymbol{w} = (w_1, \ldots, w_m)$.

3.3 Semantic Similarity

As already said, we believe that Tversky similarity may offer a semantically meaningful and cognitively plausible tool for object comparison, especially in the image domain—provided the f_i are "semantic features", that is, features with a meaningful semantic interpretation. In the image domain, such features are normally not given right away. Instead, meaningful (high-level) features have to be extracted from "low-level" feature information on the pixel level.

Thus, what we envision is an approach to image similarity based on a two-level architecture, in which semantic features are extracted from images in the

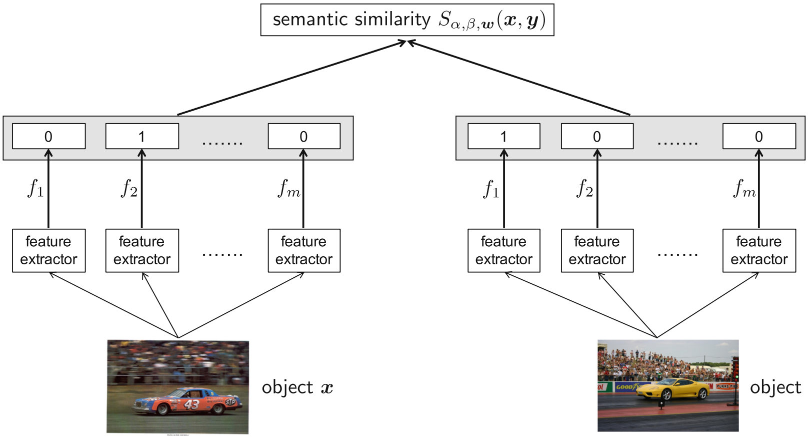

Fig. 1. Two-level architecture for semantic similarity.

first step, and these features are then used to specify similarity in the second step (cf. Fig. 1). While the features could be predefined by a human domain expert, the feature extraction itself should be accomplished automatically based on suitable image processing techniques.

For example, a human expert may know that, in a certain context, the presence of water on an image is a relevant feature. Water might then be included as a feature f_i in $\mathcal{F}$. The function f_i itself is a mapping that assumes an image (represented as a set of pixels) as an input and returns 0 or 1 as an output, depending on whether water is visible on the image or not. It could be realized, for example, by a neural network that has been trained to recognize water on images. Obviously, feature descriptors obtained in this way are never perfect, so that feature descriptions $F(\boldsymbol{x})$ will be "noisy" in practice.

4 Learning Tversky Similarity

In this section, we address the problem of *learning* the Tversky similarity, that is, tuning the parameters of the measure so as the optimally adapt it to a concrete application at hand. To this end, we assume suitable training data to be given, which informs about the similarity or dissimilarity between objects. More specifically, we assume training data of the following form:

$$\mathcal{D} = \left\{ (F(\boldsymbol{x}_n), F(\boldsymbol{y}_n), s_n) \right\}_{n=1}^{N} \subset \mathcal{X} \times \mathcal{X} \times \{0,1\} \tag{3}$$

Each training example is a triplet $(F(\boldsymbol{x}_n), F(\boldsymbol{y}_n), s_n)$, where $s_n \in \{0,1\}$ indicates whether the two objects $\boldsymbol{x}_n$ and $\boldsymbol{y}_n$ are considered similar or not. Thus, instead of precise numerical similarity degrees, the feedback is binary and only provides a rough indication. Note that the objects in the training data are already represented in terms of a feature description, i.e., we disregard the step of feature extraction here and completely focus on specifying the Tversky similarity.

4.1 Contrastive Loss

A common approach to learning similarity or distance functions is to minimize a suitable loss function on the training data $\mathcal{D}$. Here, we make use of the contrastive loss [7], which has been employed quite successfully in the image domain. This loss function compares a "predicted" similarity $\hat{s} = S_{\alpha,\beta,\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) \in [0, 1]$ with given feedback $s \in \{0, 1\}$ as follows:

$$L_m(s, \hat{s}) = s(1 - \hat{s}) + (1 - s) \max\big(m - 1 + \hat{s}, 0\big) , \tag{4}$$

where m is a margin (and a parameter of the loss). Thus, if $s = 1$ (objects $\boldsymbol{x}$ and $\boldsymbol{y}$ are considered similar), the loss is given by $1 - \hat{s}$ and increases linearly with the distance from the ideal prediction $\hat{s} = 1$. If $s = 0$, the loss is specified analogously, except that a loss of 0 only requires $\hat{s} \leq 1 - m$ (instead of $\hat{s} = 0$). Thus, positive and negative examples are treated in a slightly different way: A high (predicted) similarity on negative pairs is penalized less strongly than a low similarity on positive pairs. This could be meaningful, for example, when the feedback s is obtained from class membership in a classification or clustering context (as will be the case in our experiments) with many classes, some of which may (necessarily) overlap to a certain extent.

4.2 Loss Minimization

Training the weighted Tversky similarity (WTS) essentially consists of minimizing the total loss on the training data, i.e., finding

$$(\alpha^*, \beta^*, \boldsymbol{w}^*) = \operatorname*{argmin}_{\alpha,\beta,\boldsymbol{w}} \sum_{n=1}^{N} L_m(s_n, S_{\alpha,\beta,\boldsymbol{w}}(\boldsymbol{x}_n, \boldsymbol{y}_n)) \tag{5}$$

Besides, we consider two restricted variants of this problem: Learning the (unweighted) TS measure comes down to omitting the weight vector $\boldsymbol{w}$ from (5), and enforcing symmetry to optimizing a single parameter α instead of two parameters α and β.

Our learning algorithm (see Algorithm 1) is quite simple and essentially based on gradient descent optimization. Gradients for the parameter updates are computed on mini-batches. To avoid overfitting of the training data, we take a part of the data for validation and apply an early stopping technique. More specifically, we stop the training process as soon as the accuracy on the validation data decreases by more than 0.01 in 20 consecutive iterations.

5 Experiments

In this section, we present the results of our experimental study. Before doing so, we first describe the data we used for the experiments, the experimental setting, and the baseline methods we compared with.

Algorithm 1. Learning Algorithm

```
Input: maximum iteration $MT$, batch size $B$, margin $m$, training data $\mathcal{D}$
Output: $\alpha, \beta, \boldsymbol{w}$

split $\mathcal{D}$ into training data $\mathcal{D}_{train}$ and validation data $\mathcal{D}_{val}$
randomly initialize parameters $\alpha, \beta, \boldsymbol{w}$
while ( $t < MT$) and (stopping criterion = false)
    sample mini-batch uniformly at random from similar and dissimilar pairs
    compute $S_{\alpha,\beta,w}$ according to (2)
    update parameters by minimizing (5)
    test stopping condition on $\mathcal{D}_{val}$
end while
```

5.1 Data

Since the extraction of semantic features (cf. Sect. 3.3) is beyond the scope of this paper, we collected two image datasets for which this information, or at least information that can be interpreted as such, is already given.

The **a-Pascal-VOC2008** data [14] consists a total of 4340 images, which are split into 2113 training and 2227 test images. Each image is labeled with one of 32 class categories, such as "horse". Moreover, each image is described by 64 additional binary attributes, which we interpret as semantic features. Mostly, these features characterize the presence of certain objects on the image, such as saddle, tail, or snout.

The **Sun attributes** dataset [23] includes 14340 images equally distributed over 717 categories, such as "forest road" or "mountain path". Each image is also described by 102 attributes, such as "snow", "clouds", "flowers", etc., which describe images in a semantically meaningful way. These features are attached by 3 human annotators, and we assume the presence of each attribute in an image if it is attached by at least one of them.

5.2 Experimental Setting

We split the Sun attribute dataset into training data, validation data, and test data with a ratio of 70/10/20. The validation data is used to fine-tune parameters like the margin m of the contrastive loss. After each iteration of the training procedure (cf. Algorithm 1), the model is evaluated on validation data, and the model with highest performance is stored. We consider the same setting for the a-Pascal-VOC dataset, except for adopting the predefined train/test split. For this data, we extract 10% of the training data for validation.

Training examples for similarity learning (i.e., triplets of the form $(\boldsymbol{x}, \boldsymbol{y}, s)$) are extracted from the set of images belonging to the training data as follows: A pair of images $\boldsymbol{x}, \boldsymbol{y}$ is considered as similar ($s = 1$) if both images have the same class label, and otherwise as dissimilar ($s = 0$). To facilitate learning and have balanced batches, similar and dissimilar pairs of images are sampled with the same probability of 0.5 (and uniformly within the set of all similar pairs and all dissimilar pairs, respectively).

Once a similarity measure S (or distance measure) has been learned, its performance is evaluated on the test data. To this end, we generate data in the form of triplets $(\boldsymbol{x}, \boldsymbol{y}, s)$ in the same way as we did for training. The similarity measure is then used as a threshold classifier $(\boldsymbol{x}, \boldsymbol{y}) \mapsto [\![S(\boldsymbol{x}, \boldsymbol{y}) > t]\!]$ with a threshold t tuned on the validation data, and evaluated in terms of its classification rate (percentage of correct predictions) and F1 measure. Eventually, the average performance is determined over a very large set of randomly selected triplets (4M in the case of the a-Pascal-VOC2008 data and 6M in the case of Sun attributes), so that the estimation error is essentially 0.

5.3 Methods

We train both variants of the Tversky similarity, the unweighted (TS) and the weighted one (WTS). Since the "ground-truth" similarity in our data is symmetric by construction, we restrict to the respective symmetric versions ($\alpha = \beta$). We train unweighted Tversky similarity and weighted Tversky similarity using stochastic gradient descent with Nestrov's momentum [31] (learnin rate 0.01) and Adagrad [13] (learning rate 0.01), respectively.

As baselines for comparison, we learn two common similarity/distance metrics, namely the weighted Euclidean distance and weighted Cosine distance. Both distances are trained using Adam optimization [17] (with learning rate 0.001), in very much the same ways as the Tversky similarity. To prevent overfitting, the contrastive loss is combined with L1 and L2 regularization terms.

Moreover, we include LMNN [38] and a modified Siamese network (Siamese Net Semantic) based on [15]. The modified Siamese network consists of two blocks: two feature extraction blocks with shared weights for a pair of objects and a non-metric similarity block. A feature extraction block maps the input semantic feature vector into an embedding space, in which similar objects have small and dissimilar objects a larger distance. The non-metric similarity block predicts the similarity score that indicates the degree of similarity or dissimilarity of the input pair.

Since the number of input (semantic) features for the feature extraction block is relatively low in our experiments with the Sun attribute dataset, we only use three fully connected layers of size $[64, 32, 16]$ and activation functions [Relu, Relu, Sigmoid], respectively. Correspondingly, the dimensions of two fully connected layers are $[32, 16]$ with [Relu, Sigmoid] as activation functions for the a-Pascal-VOC2008 dataset. The non-metric network consists of an L1 distance part and two fully connected layers. In the L1 distance part, we calculate the L1 distance between features of the object pair, produced by the feature extraction blocks. The dimensions of two fully connected layers are $[16, 1]$ with [Relu, Sigmoid] as activation functions, respectively. The parameters of the modified Siamese network are learned by minimizing a combination of two common loss functions using back propagation and the Adam optimization method [17] with learning rate 0.01. The contrastive loss is exploited to pull the representation of similar images closer and push the dissimilar ones apart. Moreover, since the evaluation is eventually done on a binary classification task, we combine the contrastive loss with the cross-entropy loss to improve the classification accuracy.

To show the effectiveness of our method in obtaining semantic similarity among images, we also train the modified Siamese network based on only visual features (Siamese Net Visual). The inputs of this network are the original images, and the output is a similarity prediction that indicates whether the two input images are similar or not. In the feature extraction block, we extract high-level features from the pre-trained Inception-V3 [32] followed by a flatten layer, batch normalization, and two fully connected layers with dimensions $[256, 128]$ and activation functions [Relu, Sigmoid]. We also use the same non-metric similarity block and optimization method as explained above with a learning rate 0.01.

5.4 Results

Table 1. Performance on the a-Pascal-VOC2008 dataset.

Method	Classification rate	F1 measure
Weighted Euclidean	0.65	0.74
Weighted Cosine	0.81	0.78
LMNN [38]	0.78	0.81
Siamese Net Semantic [15]	0.82	**0.84**
Tversky (TS)	0.82	0.82
Weighted Tversky (WTS)	**0.83**	**0.84**

Table 2. Performance on the Sun attribute dataset.

Method	Classification rate	F1 measure
Weighted Euclidean	0.76	0.77
Weighted Cosine	0.78	0.80
LMNN [38]	0.79	0.81
Siamese Net Semantic [15]	0.79	0.81
Siamese Net Visual [15]	0.74	0.78
Tversky (TS)	0.80	0.81
Weighted Tversky (WTS)	**0.81**	**0.83**

The results are summarized in Tables 1 and 2. As can be seen, Tversky similarity performs very well and is highly competitive. Somewhat surprisingly, the performance of the simple unweighted version of Tversky similarity is already extremely strong. It can still be improved a bit by the weighted version, but not very much.

The Euclidean (and likewise the Cosine) distance performs quite poorly. As a possible explanation, note that the Euclidean distance is not able to ignore presumably irrelevant features that occur in none of the two objects (images) at hand. Since the distance between the corresponding feature values is 0, such features contribute to the similarity of the objects (e.g., the simultaneous absence of, say, trees on two images contributes to their similarity). This is certainly a strength of the Tversky similarity (and part of its motivation).

6 Conclusion

This paper presents first steps toward learning Tversky similarity, i.e., machine learning methods for tuning the parameters of Tversky's similarity model to training data collected in a concrete application context. To the best of our knowledge, such methods do not exist so far, in spite of the popularity of Tversky's model. The experimental results we obtained for image data so far, even if preliminary, are quite promising.

There are various directions for extending the approach presented in this paper, which ought to be addressed in future work:

- Our learning algorithm implements a rather plain solution and essentially applies a general purpose learning technique (a gradient-based optimization) to a specific loss function. More sophisticated methods, specifically tailored to the Tversky similarity and exploiting properties thereof, promise to improve efficiency and perhaps even boost performance. In this regard, we plan to elaborate on different ideas, such as alternating optimization and correlation analysis to judge feature importance.
- In practical applications, other types of training data may become relevant. An interesting example is relative similarity information of the form "object $\boldsymbol{x}$ is more similar to $\boldsymbol{y}$ than to $\boldsymbol{z}$". Devising methods for learning from data of that kind is another important topic of future work.
- Likewise, other loss functions and performance metrics should be considered, both for training and evaluation. Specifically relevant are ranking measures from information retrieval, because similarity is often used for the purpose of object retrieval (e.g., ranking images stored in a database in decreasing order of their similarity to a query image).
- Further generalizations of the Tversky similarity itself could be considered as well, for example using fuzzy instead of binary features [8,10,34].

Acknowledgements. This work was supported by the German Research Foundation (DFG) under grant HU 1284/22-1. The authors strongly profited from discussions with the collaborators Ralph Ewerth, Hubertus Kohle, Stefanie Schneider, and Matthias Springstein.

References

1. Abraham, N., Khan, N.: A novel focal Tversky loss function with improved attention U-net for lesion segmentation. In: Proceedings ISBI, IEEE International Symposium on Biomedical Imaging (2019)

2. Baioletti, M., Coletti, G., Petturiti, D.: Weighted attribute combinations based similarity measures. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R. (eds.) IPMU 2012. CCIS, vol. 299, pp. 211–220. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31718-7_22
3. Baltrušaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: a survey and taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. **41**(2), 423–443 (2018)
4. Bouchon-Meunier, B., Rifqi, M., Lesot, M.-J.: Similarities in fuzzy data mining: from a cognitive view to real-world applications. In: Zurada, J.M., Yen, G.G., Wang, J. (eds.) WCCI 2008. LNCS, vol. 5050, pp. 349–367. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68860-0_17
5. Chechik, G., Sharma, V., Shalit, U., Bengio, S.: Large scale online learning of image similarity through ranking. J. Mach. Learn. Res. **11**, 1109–1135 (2010)
6. Chen, Y., Garcia, E.K., Gupta, M.R., Rahimi, A., Cazzanti, L.: Similarity-based classification: concepts and algorithms. J. Mach. Learn. Res. **10**, 747–776 (2009)
7. Chopra, S., Hadsell, R., Le Cun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 539–546 (2005)
8. Coletti, G., Bouchon-Meunier, B.: Fuzzy similarity measures and measurement theory. In: Proceedings FUZZ-IEEE, International Conference on Fuzzy Systems, New Orleans, LA, USA, pp. 1–7 (2019)
9. Coletti, G., Bouchon-Meunier, B.: A study of similarity measures through the paradigm of measurement theory: the classic case. Soft. Comput. **23**(16), 6827–6845 (2019)
10. Coletti, G., Petturiti, D., Vantaggi, B.: Fuzzy weighted attribute combinations based similarity measures. In: Antonucci, A., Cholvy, L., Papini, O. (eds.) ECSQARU 2017. LNCS (LNAI), vol. 10369, pp. 364–374. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61581-3_33
11. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: ideas, influences, and trends of the new age. ACM Comput. Surv. **40**(2), 1–60 (2008)
12. Deselaers, T., Ferrari, V.: Visual and semantic similarity in ImageNet. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1777–1784. IEEE (2011)
13. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
14. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1778–1785. IEEE (2009)
15. Garcia, N., Vogiatzis, G.: Learning non-metric visual similarity for image retrieval. Image Vis. Comput. **82**, 18–25 (2019)
16. Hermans, A., Beyer, L., Leibe, B.: In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 (2017)
17. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (NIPS), pp. 1097–1105 (2012)
19. Kulis, B., et al.: Metric learning: a survey. Found. Trends® Mach. Learn. **5**(4), 287–364 (2013)
20. Lang, S., Ommer, B.: Attesting similarity: supporting the organization and study of art image collections with computer vision. Digit. Scholarsh. Humanit. **33**(4), 845–856 (2018)

21. Lesot, M., Rifqi, M., Benhadda, H.: Similarity measures for binary and numerical data: a survey. Int. J. Knowl. Eng. Soft Data Paradigms **1**(1), 63–84 (2009)
22. Liu, Y., Zhang, D., Lu, G., Ma, W.Y.: A survey of content-based image retrieval with high-level semantics. Pattern Recogn. **40**(1), 262–282 (2007)
23. Patterson, G., Xu, C., Su, H., Hays, J.: The sun attribute database: beyond categories for deeper scene understanding. Int. J. Comput. Vision **108**(1–2), 59–81 (2014)
24. Pedersen, T., Patwardhan, S., Michelizzi, J.: Wordnet::Similarity - measuring the relatedness of concepts. In: Proceedings AAAI, vol. 4, pp. 25–29 (2004)
25. Qian, G., Sural, S., Gu, Y., Pramanik, S.: Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In: ACM Symposium on Applied Computing, pp. 1232–1237 (2004)
26. Richter, M.M.: On the notion of similarity in case-based reasoning. In: Della Riccia, G., Kruse, R., Viertl, R. (eds.) Proceedings of the ISSEK94 Workshop on Mathematical and Statistical Methods in Artificial Intelligence. ICMS, vol. 363, pp. 171–183. Springer, Vienna (1995). https://doi.org/10.1007/978-3-7091-2690-5_12
27. Salehi, S.S.M., Erdogmus, D., Gholipour, A.: Tversky loss function for image segmentation using 3D fully convolutional deep networks. In: Wang, Q., Shi, Y., Suk, H.-I., Suzuki, K. (eds.) MLMI 2017. LNCS, vol. 10541, pp. 379–387. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67389-9_44
28. Santini, S., Jain, R.: Similarity measures. IEEE Trans. Pattern Anal. Mach. Intell. **21**(9), 871–883 (1999)
29. Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. IEEE Trans. Pattern Anal. Mach. Intell. **22**(12), 1349–1380 (2000)
30. Strehl, A., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In: Workshop on Artificial Intelligence for Web Search (AAAI), vol. 58, p. 64 (2000)
31. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: International Conference on Machine Learning, pp. 1139–1147 (2013)
32. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826 (2016)
33. Tang, H., Maitre, H., Boujemaa, N.: Similarity measures for satellite images with heterogeneous contents. In: Proceedings Urban Remote Sensing, pp. 1–9. IEEE (2007)
34. Tolias, Y.A., Panas, S.M., Tsoukalas, L.H.: Generalized fuzzy indices for similarity matching. Fuzzy Sets Syst. **120**(2), 255–270 (2001)
35. Tversky, A.: Features of similarity. Psychol. Rev. **84**(4), 327–352 (1977)
36. Tversky, A., Gati, I.: Similarity, separability, and the triangle inequality. Psychol. Rev. **89**(2), (1982)
37. Wang, J., et al.: Learning fine-grained image similarity with deep ranking. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1386–1393 (2014)
38. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. **10**, 207–244 (2009)
39. Yue-Hei Ng, J., Yang, F., Davis, L.S.: Exploiting local features from deep networks for image retrieval. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR), pp. 53–61 (2015)