Good Recognition is Non-Metric

Walter J. Scheirer[a,e,*], Michael J. Wilber[b], Michael Eckmann[c],
Terrance E. Boult[d,e]

[a]Harvard University, 52 Oxford St. NWL 209, Cambridge, MA 02138
[b]University of California San Diego, 9500 Gilman Dr., La Jolla, CA 92093
[c]Skidmore College, 815 North Broadway, Saratoga Springs, NY 12866
[d]Securics, Inc, 1867 Austin Bluffs Parkway, Suite 100, Colorado Springs, Colorado 80918
[e]University of Colorado Colorado Springs, 1420 Austin Bluffs Parkway P.O. box 7150, Colorado Springs, CO 80933-7150

Abstract

Recognition is the fundamental task of visual cognition, yet how to formalize the general recognition problem for computer vision remains an open issue. The problem is sometimes reduced to the simplest case of recognizing matching pairs, often structured to allow for metric constraints. However, visual recognition is broader than just pair-matching: what we learn and how we learn it has important implications for effective algorithms. In this review article, we reconsider the assumption of recognition as a pair-matching test, and introduce a new formal definition that captures the broader context of the problem. Through a meta-analysis and an experimental assessment of the top algorithms on popular data sets, we gain a sense of how often metric properties are violated by recognition algorithms. By studying these violations, useful insights come to light: we make the case for local distances and systems that leverage outside information to solve the general recognition problem.

Keywords: Machine Learning, Metric Learning, Recognition, Computer Vision, Face Recognition, Object Recognition

[*]Corresponding author, Phone number: (610) 657-1538 Fax: (617) 496-5424
Email addresses: `wscheirer@fas.harvard.edu` (Walter J. Scheirer), `mwilber@eng.ucsd.edu` (Michael J. Wilber), `meckmann@skidmore.edu` (Michael Eckmann), `tboult@securics.com` (Terrance E. Boult)

1. Introduction

Recognition is a term everyone in computer vision and machine learning understands – or at least we think we do. Despite multiple decades of research, it may be somewhat surprising to learn that a very basic question remains unresolved: *is recognition metric*? Familiar distance metrics used in computer vision include Euclidean distance and Mahalanobis distance, both computed in feature space. Given one of these metrics, the task of recognizing an unknown object can be approached by finding the class label of its nearest neighbor under that distance metric in a set of training samples. Such an approach provides a recognition function, thus some level of recognition can be accomplished with a metric. However, at a more fundamental level, we would like to know if distance truly captures all that is meant by the term recognition, and if metrics are good approaches to solving complex recognition tasks in computer vision. In this review article, we adopt the convention that a problem is metric if the best solutions to that problem can be achieved by directly applying a distance metric to compute the answer.

An important observation with implications for recognition is that in separable metric space, using a distance metric and the nearest neighbor (NN) algorithm provides useful consistency. As the number of i.i.d. samples from the classes approaches infinity, the NN algorithm will converge to an error rate no worse than twice the Bayes error rate, *i.e.* no worse than twice the minimum achievable error rate given the distribution of the data [3]. To many, this convergence theorem suggests that recognition can always be formulated as NN matching with an appropriate distance metric. However, having to double the error of the optimal algorithm over the same data often does not lead to a particularly good algorithm. This

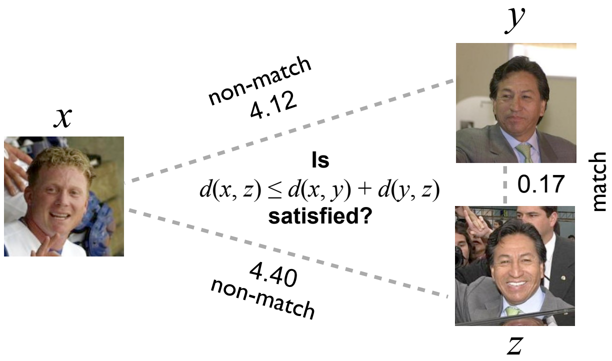

Figure 1: Assumptions are often made about the underlying nature of recognition in computer vision that do not hold true in practice. A common constraint placed upon recognition algorithms is that they must be *metric*, meaning their distance scores adhere to the properties of non-negativity, identity, symmetry and the triangle inequality. At first glance, the scores from many recognition algorithms appear to satisfy these constraints. However, violations can be subtle. For example, the distance scores produced by the top-performing Tom-vs-Pete algorithm [1] for these images from LFW [2] violate the triangle inequality.

becomes apparent when actual error rates are considered during experimentation.

With the recent popularity of metric learning [4, 5, 6, 7, 8, 9, 10, 11, 12, 13] for various recognition tasks, where a metric is learned over given pairs of images that are similar or dissimilar, one might infer that recognition is always a metric process. We note that the NN convergence theorem [3] is true for *any* metric – hence any improvements from the choice of metric, or metric learning, are not about the asymptotic error, but something else such as the error for finite samples and/or the rate of convergence. We will show that while metric learning can produce reasonable results, enforcing metric properties leaves out information, often limiting the quality of recognition with finite data. This is consistent with supporting prior work [14] in pattern recognition that shows increasing discriminative power for non-metric distance measures over visual data.

If the convergence theorem itself is about recognition, then the recognition problem is *assumed* to be formulated in an asymptotic sense with infinite i.i.d. samples. We argue that visual recognition does not rely on either of those assumptions, but rather focuses on maximizing the accuracy for finite, and, unfortunately, opportunistic and hence potentially biased sampling.

A metric function is defined as follows:

Definition 1. (Distance Metric)
A function $d : X \times X \to \mathbb{R}$ *is metric over a set* X *if it satisfies four properties for* $\{x, y, z\} \subseteq X$*:*

1. $d(x, y) \geq 0$ *(non-negativity)*
2. $d(x, y) = 0 \Leftrightarrow x = y$ *(identity)*
3. $d(x, y) = d(y, x)$ *(symmetry)*
4. $d(x, z) \leq d(x, y) + d(y, z)$ *(the triangle inequality)*

Metric functions have useful properties that allow one to show that a particular problem can be formulated as a convex minimization problem, or, as we have stated, that various types of sequences converge in the limit. There are also several cases where one of the properties is excluded. Functions that do not satisfy the triangle inequality are called semimetrics, those that violate symmetry are called quasimetrics, and those missing one or both halves of the identity requirement are called pseudometrics[1]. While the term "distance measure" is sometimes used to mean a distance metric, it is more appropriate to use this term to mean a measurement that provides information about dissimilarity, but may be formally non-metric (our use of the term follows this convention).

Is it reasonable to assume that a distance metric d maps pairs of elements from X into $\mathbb{R}$ during recognition? When a person recognizes an object, do they refer to

[1] Note that without the property of identity, the theorem of NN convergence [3] does not hold. It has also been shown [15] that the optimal distance measure, in the sense of minimal Bayes risk, always violates the identity property and therefore is not metric.

an actual image of the object of interest? A more likely alternative is a comparison to a stored model with a more complex internal representation, not a direct copy of some prior trained input. This view is consistent with prototype theory [16] in cognitive psychology. Thus, at a structural level, recognition in this mode takes an input $x \in X$, and a model M, and hence cannot be metric because it is not even of the proper functional form. It is possible to build a model using just x, and then consider the distance between models in a nearest neighbor fashion. Many instance learning algorithms do just that. However, for many other commonly used recognition algorithms, one cannot induce a proper model from a *single* input[2]. Thus, the general problem of recognition cannot be restricted to just metrics, even though it must include them.

In the core pattern recognition literature, this issue has been raised specifically in the context of Euclidean distance. Pękalska et al. [17] observe that "Non-metric dissimilarity measures may arise in practice *e.g*. when objects represented by sensory measurements or by structural descriptions are compared." Experiments to confirm this have included: comparing distance measures before and after Euclidean transforms are applied [17, 18]; an examination of the parameter space of data for metricity [14]; and an evaluation of dissimilarity representations for classification [19, 20, 18, 21, 22]. In all cases, an enforcement of Euclidean constraints does not help classification performance [23], and non-Euclidean measures are often shown to be better, leading Pękalska et al. to conclude "that non-Euclidean and/or non-metric distances can be informative and useful in statistical learning" [14].

[2] For example, consider support vector machines (SVM): one cannot draw a conceptual decision boundary without both positive and negative samples.

However, even in light of this finding, the research area of metric learning for computer vision remains quite active. A key difference from earlier work in metrics for statistical learning is that recent work in visual learning, with its strong need for data normalization, eschews Euclidean distance in favor of Mahalanobis distance [4]. In our review of the literature, we take a broader look at the many non-Euclidean metric learning approaches that have been proposed since the above studies were conducted.

Beyond statistical learning, it is natural to ask if the human mind, a most successful recognition system, operates in a way that satisfies the key metric properties of symmetry and the triangle inequality. The consensus in the cognitive psychology community is a definitive "no". In seminal work, Tversky [24] showed that human analysis of "similarity" is non-symmetric and is context dependent. One of the visual experiments conducted by Tversky was a simple pair-matching task, where subjects were asked if two block letters were the same or not. A similarity function $\mathcal{S}(p, q)$ indicated the frequency at which subjects noted letter p to be the same as q. The experiment showed that the order of presentation of the letters mattered in a statistically significant way: $\mathcal{S}(p, q) \neq \mathcal{S}(q, p)$. This result, along with others for matching faces, abstract symbols, and the names of countries led Tversky to conclude that "similarity is not necessarily a symmetric relation."

In subsequent work, Tversky and Gati [25] examined if the triangle inequality is satisfied by humans when assessing similarity. Because the triangle inequality can always be satisfied by adding a large constant to the distances between individual points when measuring dissimilarity on an ordinal scale, Tversky and Gati proposed a test that assumes segmental additivity: $d(x, z) = d(x, y) + d(y, z)$. Over numerous pair-matching trials across stimuli, human similarity judgments were found to violate the triangle inequality in a statistically significant manner. Even without the triangle inequality for additive functions, it is still possible to induce metric models with subadditive metrics. However, in experiments where subjects provided subjective probability estimates instead of ordinal numbers, Tversky and Koehler [26] were only able to show that the reported scores are often, *but not always*, subadditive[3].

Linking these findings back to pattern recognition, Duin [28, 29] finds a similar effect for the problem of judging difference between real world objects, and highlights the need for a reconsideration of the assumptions that underlie common distance measures for automated classification. If humans are employing non-metric, non-symmetric similarity measures, do we really want to constrain our recognition algorithms in computer vision to be metric? Addressing this notion in the following sections, we present the following contributions:

- A critical review of the most recent literature in metric learning for visual recognition.
- A new general definition of recognition, which includes provisions for complex models trained over sets of images and assumptions.
- An extensive meta-analysis of metric learning, along with new experiments that give an indication of how often metric constraints are violated.
- A series of useful recommendations, based on our results, for recognition algorithm designs in metric and non-metric spaces.

[3] It is possible to work around the constraint of segmental additivity using a subadditive metric based on Shepard's universal law of generalization to induce a metric from finite sets of data [27], but the result is still not consistent with the human perception findings of Tversky and Koehler [26].

2. A General Definition of Recognition

Surprisingly, a canonical definition of recognition for computer vision has yet to emerge. Many different definitions of recognition can be found in the literature, each addressing particular aspects of the problem. The familiar distance-based approach to recognition [5, 7, 11] compares feature vectors from a test image to one or more feature vectors from known images using a distance measure to indicate similarity. More compatible with recent machine learning-based approaches, statistical learning theory [30] casts recognition as risk minimization over a given loss function and joint probability distribution for a class. Other definitions include the probabilistic formulation described by Shakhnarovich et al. [31], where recognition maximizes the probability that an input distribution matches a probability rule for a single known class, as well as the NN decision rule [3] discussed in Sec. 1.

With many possibilities for class sampling, modeling for training, and strategies for matching, a concise definition that captures all of these aspects is an open issue. The above definitions tend to satisfy the definition of a particular subproblem in recognition, such as pair-matching (1:1 matching), verification (1:1 matching with a claimed class), identification (1:*n* matching), or search (1:*n* matching returning multiple results). However, no current definition captures the general problem encompassing all of them. Further, each definition is missing necessary detail with respect to the information available during matching. For a given class, there is a possibility that assumptions outside any given training examples have been made, which should be incorporated into the overall definition. These assumptions can include side-information [32], regularization terms [33], score normalization [34], or more fundamentally, data used to train a detector that is applied when pre-processing the training and testing images (*e.g.* in the case of face recognition). Another consideration is the possibility of nested or hierarchical classes, where it is necessary to return multiple class labels for a given input. With all of these issues in mind, we introduce the following comprehensive definition:

Definition 2. (The General Recognition Problem) *Given image(s) $I \in \mathbb{R}^v$, where v is the number of pixels, let $F : \mathbb{R}^v \rightarrow \mathbb{R}^D$ extract a D-dimensional feature vector x under a set of feature extractor-specific assumptions ϕ_F:*

$$x = F(I, \phi_F), x \in \mathbb{R}^D \tag{1}$$

The task of a recognition system is to find a ranked set of integer class labels considered to be the best matches to a given input feature vector x_0. For a class labeled $c \in \mathbb{N}$, let X_c be a set of training data $\{x_1, \ldots\}$ composed of m feature vectors, where $m \geq 1$. A class model M_c represents the information learned from X_c, incorporating a set of modeling-specific assumptions ϕ_M. Let R be a matching function that produces a similarity score s_c by comparing x_0 to M_c, taking into account a set of matching-specific assumptions ϕ_R:

$$s_c = R(x_0, M_c(X_c, \phi_M), \phi_R), s_c \in \mathbb{R} \tag{2}$$

For any input x_0, let S be a set of similarity scores $\{s_1, \ldots\}$ generated by n evaluations of R to compare x_0 to n known class models M_c, where $n \geq 1$. Let L be a labeling function that maps S to a ranked set of k class labels $C = \{c_1^*, \ldots\}$, where $k \geq 1$, taking into account any labeling-specific assumptions ϕ_L:

$$C = L(S, \phi_L), C \subsetneq \mathbb{N} \tag{3}$$

where $c_1^* = 0$ is reserved for the non-match label.

Def. 2 is consistent with the four common modes of recognition:

1. For pair-matching, M_c can consist of just features from a single training image $X_c = x_1$, with R a distance measure between vectors and $k = 1, c^* \in \{0, 1\}$ (non-match and match). ϕ_L contains matching criteria (*e.g.* an estimated threshold). M_c can also be a complex model over many images, matching against the image pair as x_0 (see the discussion of LFW in Sec. 3).

2. For verification, we seek to check if an input image belongs to a class c specified *a priori*, with training data defined as above for pair-matching. R could be applied n times in a multi-view setting with multiple models, matching against the set $\{M_{c_1}, \ldots\}$ for class c, where $n \geq 1$. In all cases, $\forall c^* \in C, c^* \in \{0, c\}$ and ϕ_L contains matching criteria.
3. Identification can also make use of the same training strategies as pair-matching, but always applies R over a set of n different class models, where $n \geq 2$. It returns at most one best answer with $k = 1$.
4. Search is similar to identification, but returns multiple labels, *i.e.* $k > 1$.

Next we define what it means for an algorithm to be metric.

Definition 3. (Metric Algorithm) *Let A be an algorithm that solves the recognition problem. Let R be the matching function as defined in Def. 2. A is a metric algorithm if and only if R satisfies all four properties of a metric as stated in Def. 1 for all possible inputs.*

Defs. 2 & 3 serve as general tools for deconstructing the operation of individual recognition algorithms, regardless of the context of recognition mode. Note that many recognition functions fail to satisfy the metric requirement $R : X \times X \rightarrow \mathbb{R}$, making them inherently non-metric. For instance, several of the algorithms considered in our meta-analysis (Sec. 3) make use of an SVM for pair-matching. Because the metric learning problem itself is often framed as pair-matching it may seem intuitive to assume that pair-matching with SVM would be metric. However, when R from Def. 2 is examined for an SVM class model, M_c is a combination over a set of feature vectors X_c from m different images, where $m > 1$. Thus for an SVM, $R : X \times X^m \rightarrow \mathbb{R}$ is not of the appropriate functional form to be metric.

In contrast, consider an algorithm where R is a Mahalanobis distance and where

L in Eq. 3 selects argmax over multiple classes to produce a label. If viewed as a function, the original mapping from input vector to label is not of the form required for Def. 1. However, rephrasing this algorithm in terms of Def. 2 helps us reason about its metricity by splitting the argmax from the matching function, allowing us to conclude algorithms like this are metric, because R is metric which is all that is required by Def. 3. The decomposition in Def. 2 helps us draw out such details.

3. Meta-Analysis of Algorithms for LFW

Our first case study of metric versus non-metric algorithms is Labeled Faces in the Wild [2], a popular current data set for face recognition research. LFW is ideal for testing pair-matching algorithms because it is inherently a pair-matching problem. Using the terminology of Def. 2, each algorithm selects an appropriate feature representation F, a model representation M_c, and a matching function R. Each input is a pair of feature vectors. For consistency with Def. 2, we express this as the concatenation of the two fixed-length input vectors; thus, $x_0 = F(I_1, \phi_F) \,||\, F(I_2, \phi_F)$ where $||$ denotes concatenation. Likewise, each algorithm may train on $X = \left\{x_1^+, \ldots, x_m^+, x_{m+1}^-, \ldots x_{2m}^-\right\}$, a set of m matching pairs and m nonmatching pairs of features. The labeling function $L(S, \phi_L)$ usually checks some likelihood against a threshold τ (learned as part of the labeling-specific assumptions, ϕ_L) to decide whether the pair matches, returning $c^* = 1$ if $s_1 > \tau$ and $c^* = 0$ otherwise, but certain algorithms may instead define something more complicated.

In this analysis, we consider only recent results for the "Image-restricted" setting where outside data was used for feature extraction and in the recognition system, but we briefly mention certain algorithms that take advantage of the unrestricted set. We chose this set of results because it represents several algorithms

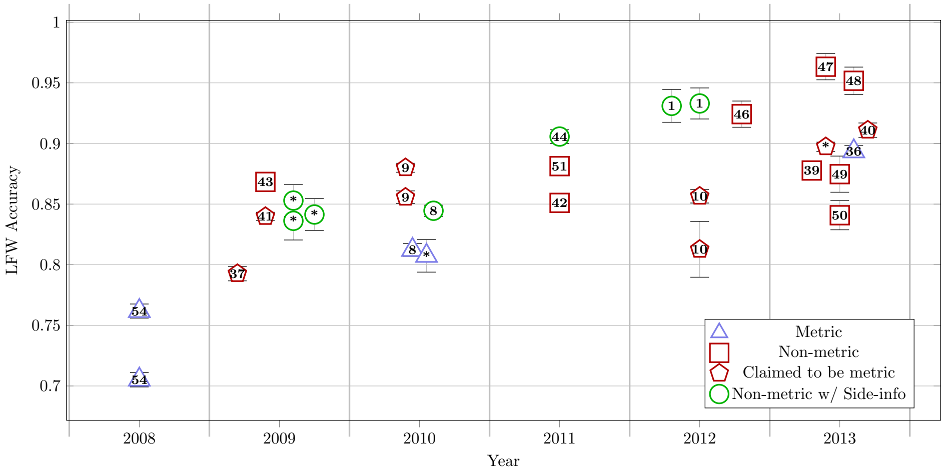

Figure 2: Recognition accuracy of algorithms on LFW. Horizontal axis is year of publication; some cluttered years are slightly separated along the horizontal axis for clarity. "Side-info" refers to algorithms that use outside data in the recognition system beyond feature extraction/alignment. "Claimed to be metric" refers to publications where the algorithm is claimed to be metric, but upon closer inspection, does not meet Def. 3's criteria for a metric learning algorithm. Even for pair-matching, purely metric algorithms are not very competitive. Numbers inside each point correspond to bibliography entries. Points marked with a * are not discussed here; references for them can be found on our companion website: http://www.metarecognition.com/metric-nometric/.

that are both metric and non-metric, allowing us to compare the performance of both. To avoid confirmation bias, we only investigate the results listed on the official LFW results web page at the time of writing [35]. By graphing the accuracy of these results over time, some interesting trends become apparent; see Fig. 2.

First, with the exception of [36], *the non-metric algorithms perform better* than the algorithms that constrain themselves to be completely metric. We investigate specific cases below. Second, *the first results reported on LFW are from metric learning algorithms*, but more recent results are not metric and do not claim to be metric. Note that in Fig. 2, we only consider an algorithm to be "metric" if it satisfies Def. 3. Merely having "distance metrics" or "metric learning" in the paper title is not enough to show this – though many of the papers claim to be metric, upon closer investigation, some of them have a non-metric R or only use metric learning as a part of their overall computation. For example, some techniques define an R that uses a local distance measure to combine information over different neighborhoods, increasing performance while making R globally non-metric.

One example of an algorithm that turns out to be non-metric is [37], which uses a custom logistic discriminant-based metric learning (LDML) approach. The algorithm specifies a nearest-neighbor-like (MkNN) normalization strategy: during testing time, each pair's score is influenced by neighborhoods of matching pairs around the two images being compared. In our words, they define a recognition function $R_{\mathrm{MkNN}}(x_0, M_c(X_c, \phi_M(x_0)), \phi_R)$. Note that $M_c(X_c, \phi_M(x_0))$ now changes at test time: instead of R_{MkNN} being fixed on a particular global model, each model's assumptions $\phi_M(x_0)$ depend on the input testing pair. From this, it is easy to see that LDML-MkNN is not globally metric: R_{MkNN} no longer satisfies symmetry or the triangle inequality because it depends on a model with assumptions that change as a function of the ordering of an image pair being classified. This is important because the extra label information available in the unrestricted set is what allows MkNN to take advantage of the pairs in each neighborhood. This implies that by making the algorithm non-metric, it can take advantage of the extra information in LFW's unrestricted set that is unavailable to the LDML-only algorithm.

Even without the MkNN step, we can make the case that the base implementation of LDML is non-metric. According to Sec. 2 of [37], the R defined by the algorithm is $R(x_0, M_c, \phi_R) = \sigma(b - d_W(F(I_1, \phi_F), F(I_2, \phi_F)))$, where b is a bias term, σ is the sigmoid function, and d_W is the Mahalanobis-like measure. Rather than actual covariance, $W \in \mathbb{R}^{D \times D}$ is a learned matrix, part of model M_c. If W was symmetric and positive-definite, it would result in a metric. However, in Sec. 2.3

of [37], it is stated that no such constraints are placed on W. Thus, this learned distance may not be even pseudometric. Note that later work [38] used these constraints and reported similar results.

Another example of an algorithm that turns out to be non-metric is the Cosine Similarity Metric Learning as presented in [9]. According to Sec. 1.2 of [9], $R_{\text{CSML}}(x_0, M_c(X_c, \phi_M), \phi_R) = \frac{(a_n)^T(b_n)}{\|a_n\|\,\|b_n\|} = \cos\theta$, where a_n and b_n are $T(F(I_{n,1}, \phi_F))$ and $T(F(I_{n,2}, \phi_F))$ for some matrix T, part of model M_c that is learned to minimize the distance between positive pairs and maximize the distance between negative pairs. The algorithm's labeling assumption ϕ_L is a threshold τ over $\cos\theta$, where θ is the angle between a_n and b_n. However, cos is not a distance metric since it may be negative and $s_c = 0$ only implies that a_n and b_n are perpendicular rather than identical, which means cos only satisfies one of the four metric properties (symmetry).

A significant advantage of CSML is that the bounds of $R_{\text{CSML}} \in [-1, 1]$ allows for a fast coarse-to-fine search for optimal parameters. In fact, many algorithms use metric learning precisely for this reason. Here, CSML has found one way to use this property while still performing better than other learning techniques, even though CSML and other algorithms based on it [39, 40] are not actually metric.

Another system that incorporates metric learning as part of a pipeline that is not completely metric is [41], which uses multiple one-shot similarity (OSS). In standard OSS, two models are trained at test time from canonical "negative" examples with each image in the image pair as positives:

$$\begin{aligned} \hat{M}_c(X_c, \phi_M) = \{&M_1(F(I_1^+, \phi_F), x_1^-, \ldots), \\ &M_2(F(I_2^+, \phi_F), x_1^-, \ldots)\} \end{aligned} \tag{4}$$

The scoring function $R_{\text{OSS}}(x_0, \hat{M}_c(X_c, \phi_M), \phi_R)$ uses each model to classify its respective input and averages the two scores. However, there is no clear way for OSS to take advantage of labels when available, so OSS may be biased toward pose, lighting, etc. Multi-OSS improves things by using multiple one-shot scores for multiple labels at test time. Note that neither OSS nor Multi-OSS are metric because each score depends on models created at testing time using different assumptions/examples. This means that none of [41, 42, 43] are metric. However, [41] shows that OSS and Multi-OSS are more effective than a variety of metric techniques. The improvement is attributed to the extra information provided by the class labels – something that the metric techniques cannot take advantage of.

According to Fig. 2, we see that the top scores come from non-metric algorithms, whether the authors intended them to be metric or not. What makes non-metric algorithms better? We emphasize that treating all samples alike may unnecessarily handicap an algorithm. For example, if one classifier is more invariant to pose, that classifier may be better than a generic classifier at handling samples with differing pose. This approach is embraced in [44], where several classifiers are trained across different subsets of the gallery for each pose combination to create a pose-adaptive classification system. Similarly, a top performing algorithm on the LFW unrestricted set, Probabilistic LDA (PLDA) [45], uses a probabilistic model based on the observation that features extracted from an image can change with respect to irrelevant variables such as pose, expression, and illumination. These variables may dwarf the variation created by the actual change in identity in the image pair. A perfect metric system must filter out such unwanted variation completely, which is impossible if all variables can influence score distances. In fact, PLDA is not metric. We show through our own experiments that this algorithm violates the triangle inequality in Sec. 5.

Other probabilistic methods [46] explicitly model the inter-personal and intra-personal variation within and between the face distributions. These methods and others based on them [47, 48, 49] are currently among the top performers on LFW's Image-restricted protocol, but do not satisfy any of Def. 1's metric properties at all. The closed form of their decision function is $R_{\mathrm{JB}}(x_0, M_c(X_c, \phi_M), \phi_R) = a_n^T W_1 a_n + b_n^T W_1 b_n - 2a_n^T W_2 b_n$, where W_1 and W_2 are learned as part of M_c. This function may be negative and is also not symmetric since $a_n^T W_2 b_n \neq b_n^T W_2 a_n$ in general. Identity also only holds when $W_1 = W_2$, a special case equivalent to the Mahalanobis distance. In fact, it can be explicitly shown [46, p.8] that a reduction in performance occurs by forcing the distance measure to converge to Mahalanobis distance, demonstrating that the non-metric algorithm captures more information.

Other non-metric algorithms include APEM [50], which trains a Gaussian mixture model (GMM) on bags of spatial appearance features of every image in the training set. The APEM algorithm adapts the feature selection process *for each face pair.* In the APEM formulation, a new GMM is trained based on the features from both images, which becomes part of the learned model assumptions. Thus, even though APEM is the second-highest performer on the LFW Image-restricted set without outside training data, it is not metric because its model incorporates assumptions learned at test time as a function of the specific image pair.

State-of-the-art deep learning approaches are also worth considering. The algorithm of Pinto and Cox [51] combines layers of several nonlinear filters applied over the original image into each model. A collection of such models is learned and the top-performing models are selected and combined. This algorithm is non-metric for several reasons. For example, each layer includes a thresholding operation to normalize its inputs, which ensures that the distance function is not globally smooth and thus does not always satisfy the triangle inequality.

What about the algorithms that might be metric? Many researchers discuss and formulate their metric learning algorithms in the sense of a globally metric feature space while mentioning, almost as an implementation detail, that they constrain their implementation to be metric only in local neighborhoods [10]. With no good alternatives, this might seem to makes sense. The authors of [10] justify the choice of a local model by arguing that it "is reasonable in the case of learning a metric for the k-NN classifiers since k-NN classifiers are influenced most by the data items that are close to the test/query examples." However, the issue of the propagation of the local/pairwise constraints is never addressed. It is well known that in metric spaces, properties on local sets often have global implications. For example, Menger [52] has shown that the embeddability of a metric into the space l_2^n (*i.e.* the Euclidean norm, with these parameters) is characterized by the embeddability of all subsets of size $n + 3$ into l_2^n. Similarly, there is a wide range of metric embeddings where local subsets imply global properties, with results for exact metrics, and similar properties even for embeddings with distortions [53]. Because the use of only local constraints, not fully propagated, induces distortion into the global metric space, we do not consider algorithms like the one in [10] to be truly metric.

However, several LFW results are unambiguously metric. For example, [54] is a linear combination of two similarity measures learned only from face pairs. Similarly, the recently-proposed PMML algorithm of [36] is a linear combination of Mahalanobis distances learned from different regions of the face. The regularizer encourages the learned matrix to be p.s.d, which makes it metric in both design and implementation. Though it is not as competitive as recent non-metric algorithms,

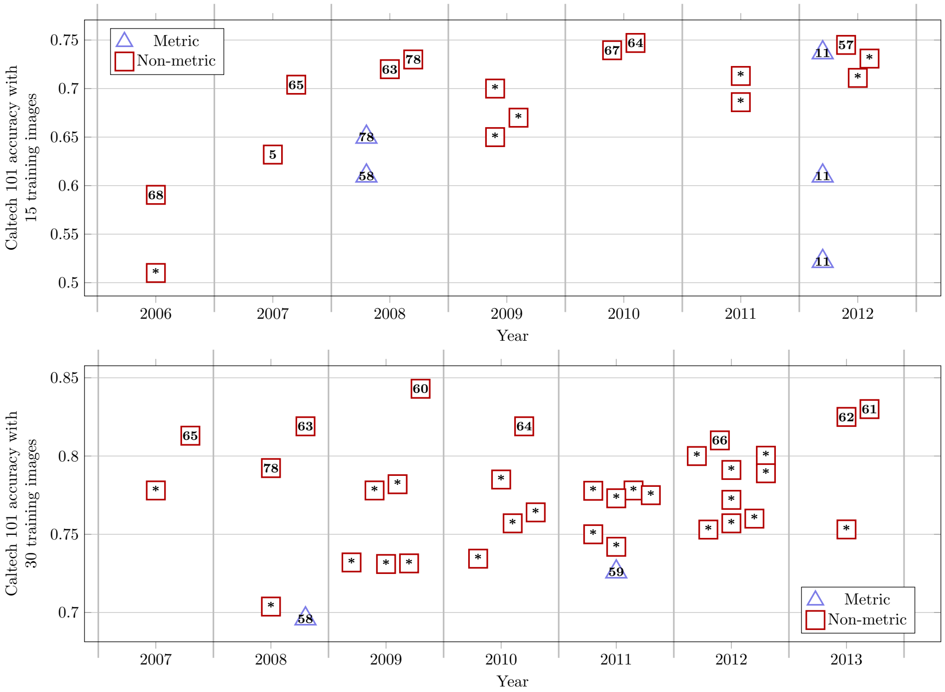

Figure 3: Recognition accuracy of algorithms on Caltech 101, with 15 training images on the top plot and 30 on the bottom plot. The horizontal axis is year of publication; some cluttered years are slightly separated along the horizontal axis for clarity. Note the metric algorithms are generally not as accurate, but are more competitive when fewer images can be used for training. Numbers inside each point correspond to bibliography entries. Note that because not all algorithms reported error bars, we do not show any error bars in this plot. Points marked with a * are not discussed here; references for them can be found on our companion website: http://www.metarecognition.com/metric-nometric/.

this algorithm is the top performing metric algorithm on LFW to date.

4. Meta-Analysis of Algorithms for Caltech 101

Our second case study examines the Caltech 101 data set [55]. Whereas LFW is ideal for analyzing pair-matching algorithms, Caltech 101 is the most well known object recognition set for identification and search scenarios, making it a useful subject of study for these other classes of recognition. We split our meta-analysis into two classes of top performing algorithms: those that use 30 training samples, the most possible, and those that use 15 training samples. To avoid confirmation bias, we only report on the 30-sample algorithms compared in work organized by Lim [56] and the additional algorithms compared in Yang et al. [57] and Jain et al. [11]. Similarly, to compare algorithms that use 15 training samples, we only consider those listed by Lim [56]. Like our analysis of LFW, we can draw some interesting conclusions by considering the plots in Fig. 3.

Notably, there is a general absence of metric methods in Fig. 3. For the algorithms making use of 30 training samples, only [58, 59] are metric. Among the top results [56], there are 33 non-metric algorithms using 30 training samples that have better accuracy than [59], which is metric. Yang et al. [60] achieved accuracy of 84.3% in 2009 with a non-metric algorithm. Some very recent non-metric algorithms [61, 62] come close to achieving that level of accuracy. Other notable non-metric algorithms that were among the best at the time of publication include [63, 64, 65, 66].

For 15 training samples, although several non-metric algorithms [57, 67, 64] do outperform it, the technique of Jain et al. [11] is metric and performs well. In Fig. 3, Jain et al. [11] appears three times. The best performing algorithm (73.7% accuracy) of the three is one from a learned kernel which is the average of a pyramid match kernel (PMK), a spatial PMK and two geometric blur kernels. The other two (61% and 52.2% accuracy) are from a learned correspondence kernel of Zhang et al. [68] and from a learned PMK kernel using SIFT, respectively. Eq. 6 in [11] is the matching function that corresponds to R in Def. 2, which is metric when the chosen kernel function $\kappa_0(x,y)$ is metric. However, the lack of metric approaches with larger amounts of training data suggests that good performance is achieved by exploiting relationships beyond pairs of samples. A common strategy for Caltech

101 is to learn a model for *multiple classes* (often using an SVM with a non-metric kernel) in a 1-vs-All configuration, which is not of the appropriate form to even be considered metric.

Analyzing two specific cases that approach the problem from a metric perspective, we again find clear violations of metric assumptions. Instead of learning a global distance metric, the technique of Frome et al. [5] learns a local distance measure for every feature vector in X_c for all classes c (resulting in a set of assumptions from $\phi_M(X_1, \ldots, X_m)$ that help build M_c) using sets of image triplets incorporating a reference image, matching image, and non-matching image. This approach is non-metric because it intentionally maintains asymmetry; Sec. 3 of [5] states "Let $f_{j,m}$ be the mth feature vector from image j. We assume a basic asymmetric distance from a single feature vector $f_{j,m}$ from one image to the set of features F_i from another." The asymmetry is inherent in computing distance within image triplets that are specific to each reference image $f_{j,m}$.

As another example, Yang et al. [57] refer to kernel metrics throughout their article and while they do use kernel metrics to build models, the overall recognition system is non-metric at a structural level. Like the algorithm of Frome et al. [5], this approach makes use of data dependent local models of groups, as opposed to global models over all of the training data. Relating this back to Def. 2, R includes group-sensitive kernel weights β_i^g (Sec. IV.A.3 of [57]) as part of its matching-specific assumptions $\phi_R(g) = \{\beta_i^g, \ldots, \beta_n^g\}$, where n is the total number of kernels, and g is a specific group. Asymmetry is again inherent in this formulation – by changing the selected group g, there is no guarantee that different weights will yield the same classification result.

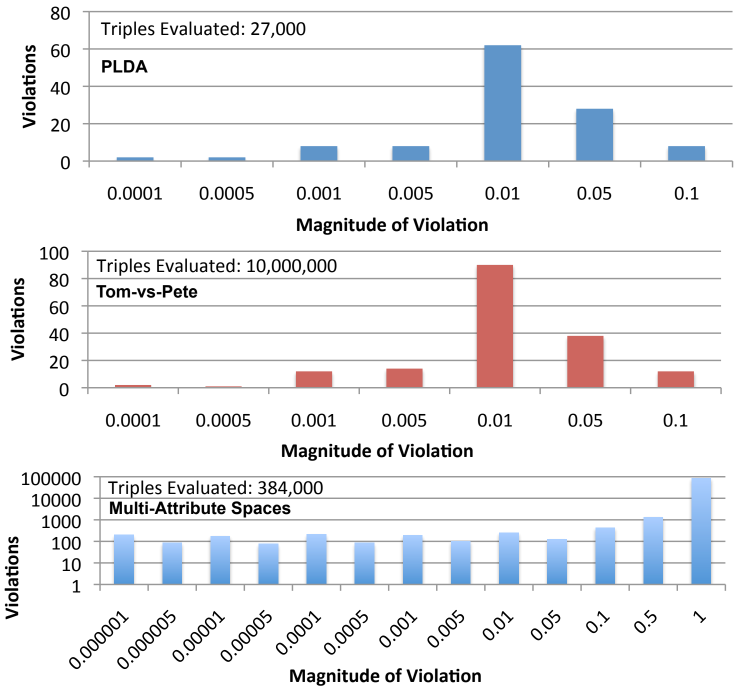

Figure 4: Results showing the distribution of violations of the triangle inequality for three recent face recognition algorithms [1, 34, 45] applied over triplets of images sampled from the LFW [2] data set. "Magnitude of Violation" refers to the difference between the sum of the lengths of two sides of the triangle and the third side that is larger than that sum, divided by the largest side of the triangle, which puts each algorithm on a common basis for comparison. Note that in some cases, it does not take a large sampling of triplets to find violations (PLDA), while in other cases, the occurrences are rare (Tom-vs-Pete), requiring a much larger evaluation. PLDA and Tom-vs-Pete follow the same distribution, suggesting there is some regularity to the pattern of violations.

5. Experimental Assessment of Metric Constraints

To gain a sense of how often the metric conditions are violated by good algorithms on pair-matching tasks that appear to be metric in form, we conducted a series of experiments. We considered three different algorithms applied to data from LFW. The first algorithm is the "Tom-vs-Pete" classification approach of Berg and Belhumeur [1], which learns a large set of identity classifiers, each trained over images for just two people. As of this writing, the "Tom-vs-Pete" algorithm is among the top three algorithms on the LFW Image-restricted Training protocol. The second algorithm is the "Multi-Attribute Spaces" approach of Scheirer et al. [34], where the statistical extreme value theory is leveraged to normalize scores across large sets of attribute classifiers for recognition tasks. The third algorithm is the "Probabilistic LDA" approach of Li et al. [45], which uses a probabilistic generative model to determine if two faces have the same underlying identity cause. It is among the top six algorithms on the LFW Unrestricted Training protocol [35].

Violations of the triangle inequality are subtle, requiring us to perform a large-scale search of the LFW image space. Triplets of images are generated by sampling image combinations from the LFW set, including cases where matches and non-matches occur. Using each algorithm, we calculated the match score for each unique image pair in the triplet, and then checked if the scores satisfied the triangle inequality. To ensure a proper evaluation of distance, the scores $s_1, \dots, s_n$ from the algorithms are processed with a simple transform T that forces a "smaller is better" result: $T(s_i) = s_\ell - s_i$, where s_ℓ is the largest score in the set $\{s_1, \dots, s_n\}$. We were able to find multiple violations for each algorithm[4]; details are provided in Fig. 4. Note that the frequency of violations is a function of the algorithm. In some cases, it does not take a large sampling of triplets to find violations (PLDA), while in other cases, the occurrences are quite rare (Tom-vs-Pete), requiring a much larger evaluation. Further, we see that PLDA and Tom-vs-Pete follow the same distribution when the violations are expressed as magnitudes and binned accordingly. The existence of this distribution suggests that there is some regularity to the pattern of violations across algorithms. However, there is some algorithmic dependence,

[4] Visual examples of these violations can be found on this article's companion website: http://www.metarecognition.com/metric-nometric/

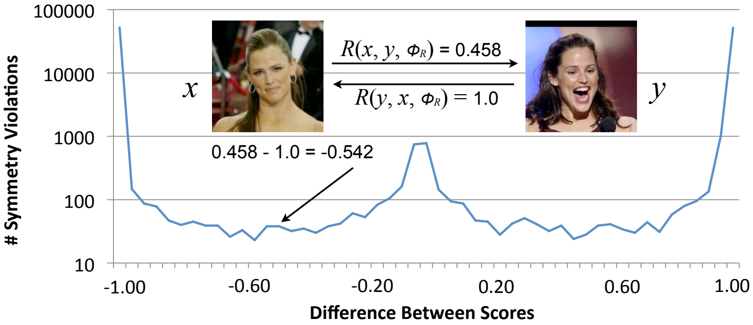

Figure 5: Violations in symmetry for the Multi-Attribute Spaces algorithm [34] applied over the Image-restricted Training Protocol of LFW. For each image pair, we calculated the score for image I_1 matching against image I_2, and vice versa. If the scores subtracted from one another do not equal 0, they are considered a violation. Here we see all violations, organized by value of difference.

since the Multi-Attribute Spaces algorithm follows a different distribution.

Understanding why these violations occur in a seemingly metric scenario is important. Similar to the MkNN algorithm [37] discussed in Sec. 3, the Multi-Attribute Spaces algorithm makes use of a local neighborhood of scores around *one* particular image (bounded from below by a parameter α, and from above by β) during a match, in order to build a good model for its normalization [34]. Thus, if the neighborhood around image x is different from the neighborhood around image y, symmetry is violated in the general case: $\phi_M(\alpha_x, \beta_x) \subseteq \{\forall s \in \mathbb{R} : \alpha_x \leq s \leq \beta_x\} \neq \phi_M(\alpha_y, \beta_y) \subseteq \{\forall s \in \mathbb{R} : \alpha_y \leq s \leq \beta_y\}$. Fig. 5 shows the prevalence of symmetry violations in the Image Restricted Training protocol of LFW for this algorithm. The formulation of Multi-Attribute Spaces also means there is no guarantee that the triangle inequality will be satisfied: the local neighborhoods considered when matching (x, y), (y, z) and (x, z) can be different from one another, often resulting in sets of distances that cause a violation. For the experiments presented here, the neighborhood around the first image is considered for the first two cases, and the neighborhood around the second image is considered for the third case. Even

	PLDA and TvP	TvP and Attributes	PLDA and Attributes
3 Exact Matches	0	0	0
2 Exact Matches	0	1224	380
1 Exact Match	10	12612	9876
3 Identity Matches	0	13	30
2 Identity Matches	10	6207	1172
1 Identity Match	88	14238	22044

Table 1: Number of violations that are common between algorithms computed over the data from Fig. 4. An "exact match" refers to the exact same image involved in the violations, while "identity match" refers to images from the same person. We do not find any of the exact same triplets violating the triangle inequality across algorithms, but there are numerous instances of common images and identities appearing between all three algorithms. This suggests that some similarity judgements are inherently data driven (supported by Tversky's observations on local features and context dependency during matching [24]), even in the case of automated algorithms.

under the weaker constraints of quasimetrics and semimetrics, the algorithm still does not satisfy what is necessary to be considered either. Since the Multi-Attribute Spaces algorithm intentionally exploits similarity around single image targets, it is unclear what advantage, if any, would be provided by enforcing the constraints of symmetry and the triangle inequality.

The statistics for the individual images involved in the violations of the triangle inequality are also interesting. Table 1 summarizes the violations that are common between algorithms computed over the data from Fig. 4. While we do not find any of the exact same triplets violating the triangle inequality across algorithms, there are still numerous instances of common images appearing between all three algorithms. This suggests that beyond algorithmic design as a cause of non-metric behavior, some similarity judgements are inherently data driven. With respect to visual data, Tversky [24] notes that local features such as color, shape, line length and orientation may detract from overall similarity matching in humans. Further, Tversky also emphasizes that a change in scene context also corresponds to a significant change in the measure of the feature space. We found that the violations

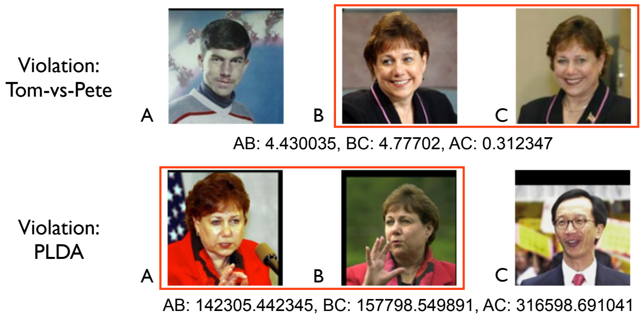
Violation:
Tom-vs-Pete
A
B
C
AB: 4.430035, BC: 4.77702, AC: 0.312347
Violation:
PLDA
A
B
C
AB: 142305.442345, BC: 157798.549891, AC: 316598.691041

Figure 6: An example of a common identity occurring in violations of the triangle inequality across algorithms. Note that each pair of images containing the same identity shows a change in scene context (top: room change; bottom: indoor/outdoor). The numbers below each violation indicate the distance between images in the "triangle". An identity that yields only a subset of local features for matching coupled with a change in context is one possible explanation for this phenomenon [24]. Many more examples of such violations can be found on this article's companion website: http://www.metarecognition.com/metric-nometric/.

often include a change in context across the same identity within a triple (*i.e.* the same person in two different settings) *and* that certain identities appear more frequently than others in the violations. An example is shown in Fig. 6. This can possibly be attributed to a combination of emphasis on local features and scene context during matching for those identities in LFW.

We also conducted a second series of experiments to assess, on a common feature basis, the accuracy and training time of a prevalent metric learning approach for visual recognition tasks versus a typical "off-the-shelf" non-metric supervised method in machine learning. Our meta-analysis provides an indication of general performance, but leaves open the possibility that the metric learning algorithms simply made use of weaker features, and hence did not perform as well as non-metric algorithms leveraging better features. To address this, we selected three very well-studied machine learning benchmark feature sets for visual data: USPS [72], Letter [73], and Satimage [73]. We trained one classifier for each set using the Information-Theoretic Metric Learning (ITML) algorithm [69, 70] and

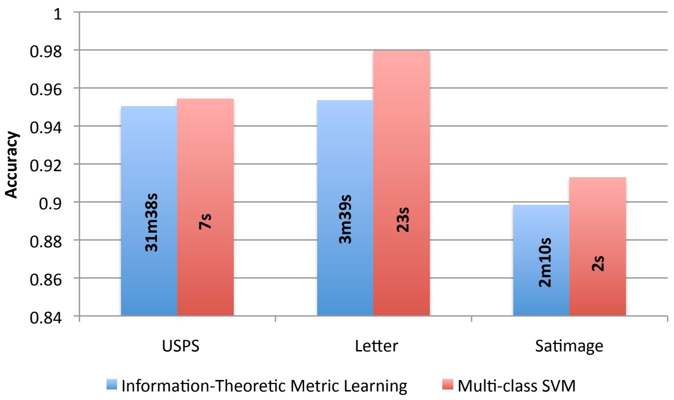

Figure 7: Information-Theoretic Metric Learning versus "1-against-1" Multi-Class SVM. For this experiment, we compared a prevalent metric learning approach [69, 70] to a typical "off-the-shelf" choice for non-metric supervised learning [71] on a common feature basis, which our meta-analysis does not directly provide. Across three very well-known machine learning visual benchmark sets (USPS [72], Letter [73], and Satimage [73]), we observe a clear trend: Information-Theoretic Metric Learning yields lower accuracies at the expense of extra time for training (shown inside each bar).

the "1-against-1" multi-class SVM with an RBF kernel provided by LIBSVM [71]. We set SVM parameters C and γ using the values reported by Nguyen and Ho [74] for USPS, and those reported by Hsu and Lin [75] for Letter and Satimage.

The results for the comparison can be seen in Fig. 7. For all three data sets, ITML yields lower accuracies at the expense of extra time for training. Considering the relative ease of the data sets and that SVM is an older approach, it is somewhat surprising that ITML, a more recent algorithm that is the foundation of a variety of metric learning work [58, 76, 7, 11], is not the best performing approach here. This finding is consistent with previous studies found in the literature using multi-class SVM as a point of comparison [77].

A further note should be made on the dimensionality of feature vectors used for learning. We considered performing a more exhaustive experiment comparing features for Caltech 101 and LFW from the best algorithms, but encountered a problem with respect to the limits of what data is feasible to compute with metric learning. As described by Guillaumin et al. [37], available reference implementations for ITML and LDML (described in Sec. 3) are “intractable when using 600” or more feature dimensions. Even for the most standard off-the-shelf but well-performing object recognition features (*e.g.* HOG), we must consider several thousand dimensions for a data set such as Caltech 101. Thus, metric learning approaches turn to dimensionality reduction to reduce the feature representations before training. Of course, this introduces the risk of discarding information that may be valuable for recognition. Moreover, even though ITML is provably convex, this does not mean an optimal solution can be found in a practical amount of time for a feature set.

6. Discussion

During the course of this work, we found that some problems and their corresponding solutions do not even have the structural form necessary to be metric – they compare input features to more complex models. Similar observations have been made before [17, 78, 29]. In [78] it is proved “that under the Naive-Bayes assumption, the *optimal* distance to use in image classification is the KL “Image-to-Class” distance, and not the commonly used “Image-to-Image” distribution distances.” Moreover, even for the restricted recognition problem of pair-matching, which at least initially looks as if it is metric, the best performing algorithms have a model for “matched pairs” that is non-metric. Metric properties allow some powerful mathematical machinery to be employed and, with effort, any recognition problem’s solution can be “made” metric – the question is if metric constraints improve recognition performance. Our meta-analysis and experimental analysis of top-performing algorithms show violations of symmetry for some and violations of the triangle inequality for others. With so many cases where performance improves as metric conditions are relaxed (an observation supported by the pattern recognition literature [14]), we conclude that, in general, *good recognition is non-metric*.

However, this article should *not* be interpreted as suggesting that metrics have no role in computer vision or that metric learning is not useful for recognition. On the contrary, our analysis has shown that metric learning has provided interesting first cut solutions. Furthermore, many good recognition algorithms use local distance measures as the core of an overall non-metric algorithm. Learning metrics, at least locally, appears to be an effective way to incorporate various types of constraints. In many cases, the original feature space (Eq. 1) is transformed into another locally normalized/metric feature space, before combining data, yielding a non-metric but effective scoring process.

One observation, which can be exploited in other vision work, is why we believe the problem is inherently non-metric. General recognition problems must capture and model the uncertainty in the data and in the class definitions. They must handle local variations in features, in sample density and in labeling. If, as is true in the general setting, the data is not uniformly sampled with uniform error, good recognition algorithms develop local distance measures in a way that may result in asymmetric measures and/or measures that violate the triangle inequality. Thus, even if one chooses to use local metric learning to help normalize the data, one should also look for models that integrate multiple sources of information (including side-information) and use them to model the regional variations and errors.

A good metric-based recognition algorithm would need to have approximately uniform error. If its "learning" could transform an inherently non-uniform biased sampling and errors into a single representation with uniform errors, it would provide a near perfect “whitening” filter correcting the per-class biases and errors. While it is true that in the limit, assuming i.i.d. samples, a metric plus nearest neighbor classification has an error rate no more than twice the Bayes error rate, we note that “in the limit” the infinite i.i.d. sampling requirement is effectively removing any sampling bias and providing uniform error. Most recognition problems do not have the luxury of i.i.d. sampling nor can they wait for the limit of infinite samples. Thus we believe it is important to develop robust features and models of uncertainty/error for more effective recognition algorithms.

We emphasize that this study is ongoing. The rapid evolution of learning algorithms will likely lend new perspectives on this issue as the results reported for Caltech 101 and LFW reach ceiling. We encourage interested readers to submit new algorithms to be included in the meta-analysis through this article’s companion website: http://www.metarecognition.com/metric-nometric/.

References

[1] T. Berg, P. Belhumeur, Tom-vs-pete classifiers and identity-preserving alignment for face verification, in: Proc. of the British Machine Vision Conference, 2012, pp. 129.1–129.11.

[2] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. Rep. 07-49, UMass Amherst (Oct. 2007).

[3] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27.

[4] B. Kulis, Metric learning: a survey, Foundations and Trends in Machine Learning 5 (4) (2013) 287–364.

[5] A. Frome, Y. Singer, F. Sha, J. Malik, Learning globally-consistent local distance functions for shape-based image retrieval and classification, in: Proc. of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[6] S. Ziang, F. Nie, C. Zhang, Learning a mahalanobis distance metric for data clustering and classification, Pattern Recognition 41 (12) (2008) 3600–3612.

[7] B. Kulis, P. Jain, K. Grauman, Metric and kernel learning using a linear transformation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (12) (2009) 2143–2157.

[8] Z. Cao, Q. Yin, X. Tang, J. Sun, Face recognition with learning-based descriptor, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2707–2714.

[9] H. V. Nguyen, L. Bai, Cosine similarity metric learning for face verification, in: Proc. of the Asian Conference on Computer Vision, 2010, pp. 709–720.

[10] Y. Ying, P. Li, Distance metric learning with eigenvalue optimization, Journal of Machine Learning Research 13 (2012) 1–26.

[11] P. Jain, B. Kulis, J. Davis, I. Dhillon, Metric and kernel learning using a linear transformation, Journal of Machine Learning Research 13 (2012) 519–547.

[12] Y. Mu, W. Ding, D. Tao, Local discriminative distance metrics ensemble learning, Pattern Recognition 46 (8) (2013) 2337–2349.

[13] Q. Wang, P. Yuen, G. Feng, Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions, Pattern Recognition 46 (9) (2013) 2576–2587.

[14] E. Pękalska, A. Harol, R. Duin, B. Spillmann, H. Bunke, Non-euclidean or non-metric measures can be informative, in: D.-Y. Yeung, J. Kwok, A. Fred, F. Roli, D. Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Vol. 4109 of Lecture Notes in Computer Science, Springer, 2006, pp. 871–880.

[15] S. Mahamud, M. Hebert, Minimum risk distance measure for object recognition, in: Proc. of the IEEE International Conference on Computer Vision, 2003, pp. 242–248.

[16] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, P. Boyes-Braem, Basic objects in natural categories, Cognitive Psychology 8 (1976) 382–439.

[17] E. Pękalska, R. Duin, S. Gunter, H. Bunke, On not making dissimilarities euclidean, in: A. Fred, T. Caelli, R. Duin, A. Campilho, D. de Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Vol. 3138 of Lecture Notes in Computer Science, Springer, 2004, pp. 1145–1154.

[18] R. Duin, E. Pękalska, A. Harol, W. Lee, H. Bunke, On euclidean corrections for non-euclidean dissimilarities, in: N. Lobo, T. Kasparis, F. Roli, J. Kwok, M. Georgiopoulos, G. Anagnostopoulos, M. Loog (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Vol. 5342 of Lecture Notes in Computer Science, Springer, 2008, pp. 551–561.

[19] E. Pękalska, R. Duin, Dissimilarity representations allow for building good classifiers, Pattern Recognition Letters 23 (8) (2002) 943–956.

[20] E. Pękalska, R. Duin, The dissimilarity representation for pattern recognition: foundations and applications, World Scientific Pub Co Inc, 2005.

[21] R. Duin, E. Pękalska, Non-euclidean dissimilarities: causes and informativeness, in: E. Hancock, R. Wilson, T. Windeatt, I. Ulusoy, F. Escolano (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Vol. 6218 of Lecture Notes in Computer Science, Springer, 2010, pp. 324–333.

[22] Y. Plasencia-Calana, E. Garcia-Reyes, R. Duin, M. Orozco-Alzate, On using asymmetry information for classification in extended dissimilarity spaces, in: L. Alvarez, M. Mejail, L. Gomez, J. Jacobs (Eds.), Proc. of the Iberoamerican Congress on Pattern Recognition, Vol. 7441 of Lecture Notes in Computer Science, Springer, 2012, pp. 503–510.

[23] R. Duin, E. Pękalska, Study on (non)geometricty, Deliverable D3.1, SIMBAD (EU, FP7, FET) (2009).

[24] A. Tversky, Features of similarity, Psychological Review 84 (4) (1977) 327–352.

[25] A. Tversky, I. Gati, Similarity, separability, and the triangle inequality, Psychological Review 89 (2) (1982) 123–154.

[26] A. Tversky, D. Koehler, Support theory: A nonextensional representation of subjective probability, Psychological Review 101 (4) (1994) 547.

[27] F. Jäkel, B. Schölkopf, F. Wichmann, Similarity, kernels, and the triangle inequality, Journal of Mathematical Psychology 52 (5) (2008) 297–303.

[28] R. Duin, Pattern recognition as a human centered non-euclidean problem, in: A. Fred (Ed.), Pattern Recognition in Information Systems, Vol. 73 of Proc. of the International Conference on Enterprise Information Systems, SciTePress, 2010, pp. 3–12.

[29] R. Duin, Non-euclidean problems in pattern recognition related to human expert knowledge, in: J. Filipe, J. Cordeiro (Eds.), Enterprise Information Systems, Vol. 73 of Lecture Notes in Business Information Processing, Springer, 2011, pp. 15–28.

[30] V. Vapnik, Estimation of Dependences Based on Empirical Data, Springer-Verlag, 1982.

[31] G. Shakhnarovich, J. Fisher, T. Darrell, Face recognition from long-term observations, in: Proc. of the European Conference on Computer Vision, 2002, pp. 851–868.

[32] E. Xing, A. Y. Ng, M. I. Jordan, S. Russell, Distance metric learning, with application to clustering with side-information, in: Advances in Neural Information Processing Systems 15, 2002, pp. 505–512.

[33] B. Schölkopf, A. J. Smola, Learning with Kernels, MIT Press, 2002.

[34] W. Scheirer, N. Kumar, P. Belhumeur, T. Boult, Multi-attribute spaces: calibration for attribute fusion and similarity search, in: Proc of the IEEE Conference on Computer Vision and Pattern, 2012, pp. 2933–2940.

[35] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, LFW results, Internet: `http://vis-www.cs.umass.edu/lfw/results.html`, [Accessed: Nov. 17, 2013].

[36] Z. Cui, W. Li, D. Xu, S. Shan, X. Chen, Fusing robust face region descriptors via multiple metric learning for face recognition in the wild, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3554–3561.

[37] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? Metric learning approaches for face identification, in: Proc. of the IEEE International Conference on Computer Vision, 2009, pp. 498–505.

[38] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, Face recognition from caption-based supervision, International Journal of Computer Vision 96 (1) (2012) 64–82.

[39] D. Yi, Z. Lei, S. Li, Towards pose robust face recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3539–3545.

[40] O. Barkan, J. Weill, L. Wolf, H. Aronowitz, Fast high dimensional vector multiplication face recognition, in: Proc. of the IEEE International Conference on Computer Vision, 2013, pp. 1960–1967.

[41] Y. Taigman, L. Wolf, T. Hassner, Multiple one-shots for utilizing class label information, in: Proc. of the British Machine Vision Conference, 2009, pp. 77.1–77.12.

[42] H. J. Seo, P. Milanfar, Face verification using the lark representation, IEEE Transactions on Information Forensics and Security 6 (4) (2011) 1275–1286.

[43] L. Wolf, T. Hassner, Y. Taigman, Similarity scores based on background samples, in: Proc. of the Asian Conference on Computer Vision, 2009, pp. 88–97.

[44] Q. Yin, X. Tang, J. Sun, An associate-predict model for face recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 497–504.

[45] P. Li, Y. Fu, U. Mohammed, J. H. Elder, S. J. Prince, Probabilistic models for inference about identity, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (1) (2012) 144–157.

[46] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: A joint formulation, in: Proc. of the European Conference on Computer Vision, 2012, pp. 566–579.

[47] X. Cao, D. Wipf, F. Wen, G. Duan, A practical transfer learning algorithm for face verification, in: Proc. of the IEEE International Conference on Computer Vision, 2013, pp. 3208–3215.

[48] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3025–3032.

[49] K. Simonyan, O. M. Parkhi, A. Vedaldi, A. Zisserman, Fisher vector faces in the wild, in: Proc. of the British Machine Vision Conference, 2013, pp. 1–12.

[50] H. Li, G. Hua, Z. Lin, J. Brandt, J. Yang, Probabilistic elastic matching for pose variant face verification, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3499–3506.

[51] N. Pinto, D. Cox, Beyond simple features: A large-scale feature search approach to unconstrained face recognition, in: Proc. of the IEEE International Conference on Automatic Face & Gesture Recognition, 2011, pp. 8–15.

[52] K. Menger, New foundation of euclidean geometry, American Journal of Mathematics 53 (4) (1931) 721–745.

[53] M. Charikar, K. Makarychev, Y. Makarychev, Local global tradeoffs in metric embeddings, SIAM Journal on Computing 39 (6) (2010) 2487–2512.

[54] G. Huang, M. Jones, E. Learned-Miller., LFW results using a combined nowak plus MERL recognizer, in: Proc. of the Faces in Real-Life Images Workshop, 2008, pp. 1–2.

[55] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories, in: Proc. of the IEEE CVPR Workshop on Generative-Model Based Vision, 2004, pp. 178–185.

[56] H. W. Lim, Caltech 101 results, `http://zybler.blogspot.com/2009/08/table-of-results-for-famous-public.html`, [Accessed: Nov. 17, 2013] (2012).

[57] J. Yang, Y. Li, Y. Tian, L. Duan, W. Gao, Group-sensitive multiple kernel learning for object categorization, IEEE Transactions on Image Processing 21 (5) (2012) 2838–2852.

[58] P. Jain, B. Kulis, K. Grauman, Fast image search for learned metrics, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[59] A. Coates, A. Ng, The importance of encoding versus training with sparse coding and vector quantization, in: Proc. of the International Conference on Machine Learning, 2011, pp. 921–928.

[60] J. Yang, Y. Li, Y. Tian, L. Duan, W. Gao, Group-sensitive multiple kernel learning for object categorization, in: Proc. of the IEEE International Conference on Computer Vision, 2009, pp. 436–443.

[61] Q. Li, H. Zhang, J. Guo, B. Bhanu, L. An, Reference-based scheme combined with k-svd for scene image categorization, IEEE Signal Processing Letters 20 (1) (2013) 67–70.

[62] D. F. L. Bo, X. Ren, Multipath sparse coding using hierarchical matching pursuit, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 660–667.

[63] S. Todorovic, N. Ahuja, Learning subcategory relevances for category recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[64] F. Li, J. Carreira, C. Sminchisescu, Object recognition as ranking holistic figure-ground hypotheses, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1712–1719.

[65] A. Bosch, A. Zisserman, X. Muoz, Image classification using random forests and ferns, in: Proc. of the IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[66] S. McCann, D. Lowe, Spatially local coding for object recognition, in: Proc. of the Asian Conference on Computer Vision, 2012, pp. 204–217.

[67] A. Kapoor, K. Grauman, R. Urtasun, T. Darrell, Gaussian processes for object categorization, International Journal of Computer Vision 88 (2) (2010) 169–188.

[68] H. Zhang, A. Berg, M. Maire, J. Malik, SVM-KNN: Discriminative nearest neighbor classification for visual category recognition, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2126–2136.

[69] J. V. Davis, B. Kulis, P. Jain, S. Sra, I. S. Dhillon, Information-theoretic metric learning, in: Proc. of the International Conference on Machine Learning, 2007, pp. 209–216.

[70] J. V. Davis, B. Kulis, P. Jain, S. Sra, I. S. Dhillon, Information Theoretic Metric Learning, UT, Austin, http://www.cs.utexas.edu/users/pjain/itml/.

[71] C.-C. Chang, C.-J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011) 27:1–27:27, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[72] J. Hull, A database for handwritten text recognition research, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (5) (1994) 550–554.

[73] D. Michie, D. Spiegelhalter, C. Taylor, Machine Learning, Neural and Statistical Classification, Prentice Hall, 1994.

[74] D. Nguyen, T. Ho, An efficient method for simplifying support vector machines, in: Proc. of the International Conference on Machine Learning, 2005, pp. 617–624.

[75] C. Hsu, C. Lin, A comparison of methods for multi class support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.

[76] P. Jain, B. Kulis, J. Davis, I. Dhillon, Metric and kernel learning using a linear transformation, arXiv:0910.5932.

[77] K. Q. Weinberger, J. Blitzer, L. K. Saul, Distance metric learning for large margin nearest neighbor classification, in: Advances in Neural Information Processing Systems 18, 2006, pp. 1473–1480.

[78] O. Boiman, E. Shechtman, M. Irani, In defense of nearest-neighbor based image classification, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.