Semi-Supervised Object Recognition Based on Connected Image Transformations

Steven Van Vaerenbergh[a,*], Ignacio Santamaria[a], Paolo Emilio Barbano[b]

[a] *Dept. Communications Engineering, University of Cantabria, Santander, Spain*
[b] *Dept. of Applied Mathematics and Theoretical Physics, University of Cambridge, UK*

Abstract

We present a novel semi-supervised classifier model based on paths between unlabeled and labeled data through a sequence of local pattern transformations. A reliable measure of path-length is proposed that combines a local dissimilarity measure between consecutive patters along a path with a global, connectivity-based metric. We apply this model to problems of object recognition, for which we propose a practical classification algorithm based on sequences of "Connected Image Transformations" (CIT). Experimental results on four popular image benchmarks demonstrate how the proposed CIT classifier outperforms state-of-the-art semi-supervised techniques. The results are particularly significant when only a very small number of labeled patterns is available: the proposed algorithm obtains a generalization error of 4.57% on the MNIST data set trained on 2000 randomly chosen patterns with only 10 labeled patterns per digit class.

Keywords: semi-supervised classification, object recognition, connectivity, deformation models, low-density separation

1. Introduction

In many object recognition problems, obtaining labeled data is a time-consuming and expensive task, whereas large unlabeled data sets are usually available. This is particularly true in problems involving high-dimensional

*Corresponding author: Tel: +34 942200919 ext 802, Fax: +34 942201488.
Email addresses: steven@gtas.dicom.unican.es (Steven Van Vaerenbergh), ignacio@gtas.dicom.unican.es (Ignacio Santamaria), p.e.barbano@damtp.cam.ac.uk (Paolo Emilio Barbano)

data, such as handwritten digit recognition, text categorization [1], protein classification [2] or hyper-spectral data classification [3]. In such scenarios it is desirable to develop semi-supervised learning techniques, as these allow to exploit the available unlabeled data concurrently with the labeled training data. We focus on recognition problems in which many instances of each object are available for training, and each instance differs only slightly from another instance of the same object. This is typically the case in handwritten digit and face recognition systems, but it occurs more generally in a wide variety of image recognition problems where series of spatially or temporally related patterns are available. In this case, the available instances of an object usually relate to each other by transformations such as rotations, scalings and small nonlinear axis deformations.

A large number of semi-supervised learning techniques have been proposed in the last years, for instance [4, 5, 6, 7, 8, 9, 10, 11, 12]. The success of these techniques relies mainly on two key assumptions: i) the data lie on a manifold of much lower dimensionality than the data dimension itself (manifold assumption) [12]; and ii) data points belonging to the same high-density region are likely to belong to the same class (cluster assumption) [11]. Both assumptions can be interpreted in terms of data similarity and distances. In this sense, the manifold assumption states that *local* variations in the data should only involve variations of a small number of parameters. This property is illustrated in Fig. 1, which shows a number of handwritten instances of the number 3: although the data dimensionality is high, most local variations can be described by few parameters, such as line thickness, skew and rotation. Therefore, the manifold assumption leads naturally to the concept of a *local* distance between patterns. Several algorithms exploit the manifold assumption, e.g. [13], by estimating the marginal distribution underlying the data and training a classifier on the manifold itself.

The cluster assumption states that two data points should belong to the same class if they can be connected by a path that lies exclusively in a region of high density. This assumption, which was exploited for instance in [11, 14], allows to define a *global* distance measure between patterns that lie further apart. Specifically, the global distance between two points is measured as the length of the path between them, in which each connection is measured as a local dissimilarity between two intermediate patterns. Therefore, while the manifold assumption refers to a local dissimilarity, the cluster assumption refers to a global distance.

The proposed semi-supervised method uses small pattern transformations

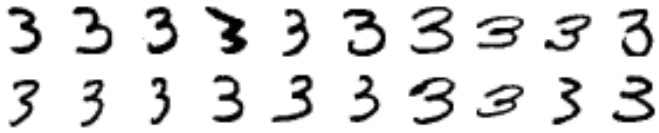


Figure 1: Handwritten instances of the number 3, from the MNIST data set.

as the local dissimilarity measure. They are accumulated along a path using a connectivity distance to obtain a robust and reliable global distance, and a simple nearest-neighbor technique is finally used for classification. Despite of its fairly simple formulation, the new algorithm outperforms state-of-the-art semi-supervised classification algorithms when tested on standard benchmark image data sets. Some preliminary results of the proposed method appeared in [15]. Here, we extend the experimental study of the algorithm, and we formulate the out-of-sample classification procedure. The algorithm has quadratic time and memory complexity in terms of the number of training points, which becomes impractical if large data sets are used. In order to reduce the out-of-sample classification cost, we also propose a prototype-based approximation procedure.

The paper is organized as follows: in Section 2, we provide some related literature and previous work. In Section 3, we review the semi-supervised classification setting and state the main assumptions on which the proposed method is based. Section 4 introduces the local and global dissimilarity measures, which form the basis of the proposed semi-supervised classifier, and it describes the proposed technique. An out-of-sample extension is discussed in Section 5, including a strategy for dealing with large-scale data sets. Section 6 illustrates the obtained performance in comparison to other state-of-the-art techniques on four typical databases. Finally, we summarize the main conclusions of this work in Section 7.

2. Previous work on semi-supervised classification

Much of the recent effort in semi-supervised learning has been centered around the problem of finding a reliable method to infer a global distance measure from local dissimilarities [7, 8, 9, 10, 12, 13], which is also the main problem addressed in the present contribution. Most of these techniques start by constructing an undirected weighted graph (or, equivalently, an affinity

matrix) on the labeled and unlabeled data points, where the edge weights measure the pairwise dissimilarities. Then, they apply different approaches to design a global classifying function with desirable properties (e.g., smoothness, robustness, etc.). For instance, [7] and [8] use a probabilistic approach in which the graph weights (local dissimilarities) are viewed as transition probabilities and the global dissimilarities are established through a random walk or a diffusion process on the graph, respectively. The local dissimilarity metric in [7],[8], however, is computed by the standard Gaussian kernel, which makes the estimate of the shortest path length more sensitive to noise. Instead of considering just the shortest path, these algorithms integrate the volume of all paths between two data points, hence effectively de-noising the global metric.

Closely related approaches that eliminate the dependency of [7] and [8] with respect to the diffusion time are the harmonic Gaussian field classifier described in [9] and the consistency method in [10]. These methods estimate a global metric on the weighted graph (i.e., the semi-supervised classifier) by repeatedly applying the Laplacian matrix (or some of its normalized versions) over a matrix of labels which is consistent with the training data. Over iterations, label information is propagated through the graph and, after reaching a stable state, the unlabeled patterns are assigned to the classes from which they have received more information. Again, these methods use the conventional Gaussian kernel as the local similarity function for computing the affinity matrix. The smoothness constraint imposed by the Laplacian is in this case responsible for de-noising the global metric.

In [12] Belkin et al. proposed a framework that exploits the geometry of the underlying marginal distribution, which can be estimated from unlabeled data, to regularize the data manifold. This principle was used to design a semi-supervised classifier, denoted as the Laplacian Support Vector Machine (LapSVM). The resulting classifier has the interesting property of providing a natural out-of-sample extension. In order to lower the cubic training complexity of LapSVM, a training algorithm in the primal was recently proposed in [16].

While many other graph-based approaches for semi-supervised classification have been proposed over the past years, all of them use for the local dissimilarities a function of the Euclidean distance with exponential decay, typically the Gaussian kernel, regardless of the particular application considered. Their emphasis is on how a suitable global metric or function for semi-supervised learning should be estimated from a graph, and to this end

they proposed quite sophisticated methods. Departing from that trend, in this work we demonstrate that better results can be obtained by translating most of the complexity to the computation of the local dissimilarity measure. This metric should be problem-dependent to better characterize the data manifold structure at a local scale. In doing so, we can simplify the global metric as a shortest path which can be implemented using Dijkstra's algorithm, as we will show below. This conceptually simple procedure provides very good results in different scenarios.

3. Problem formulation and assumptions

We consider a multi-class classification problem with N classes $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$. In a semi-supervised classification setting, we are given a training data set consisting of $n = l + u$ patterns, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_l, \mathbf{x}_{l+1} \ldots, \mathbf{x}_n\} = \mathcal{X}_l \cup \mathcal{X}_u$, represented as Euclidean vectors of dimension *dim*. The first l patterns in this set correspond to labeled data, with class labels $\{y_1, \ldots, y_l\}$, while the remaining u patterns constitute the available unlabeled data. We assume that all input patterns $\mathbf{x}_i$ have been drawn independently and identically distributed (i.i.d.) from some unknown marginal data distribution $P(\mathbf{x})$. This is the conventional setting assumed in most semi-supervised classification techniques described in the literature [5, 17]. Furthermore, in this paper we are interested in semi-supervised classification problems where l is a very small fraction of the total number of available patterns, n.

We now make the following assumptions:

Assumption 1. *For each pattern $\mathbf{x}_i$ in the data set there exist close patterns $\mathbf{x}_j$ of the same class ($y_i = y_j$) that can be obtained by small transformations of the given pattern. The norm of these transformations is measured by some intrinsic dissimilarity measure.*

Assumption 2. *For any two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ that belong to the same class ($y_i = y_j$), there exists a sequence of k transformations*

$$\mathbf{x}_j = T_k \circ T_{k-1} \circ \cdots \circ T_2 \circ T_1(\mathbf{x}_i) \tag{1}$$

which is both short *(i.e. the total number of transformations k is small), and* well connected *(i.e. the norm of the transformation between two consecutive patterns along the path is also small). These sequences are referred to as* consistent*. Accordingly, if two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different classes*

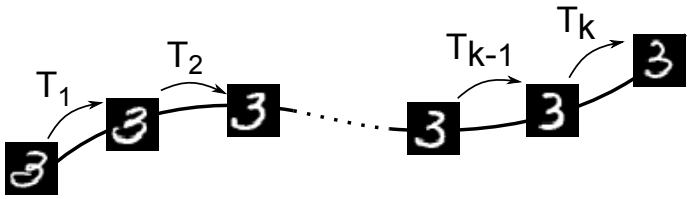


Figure 2: By applying a sequence of small transformations it is possible to transform any pattern into another pattern of the same class. All patterns of the sequence belong to the same low-dimensional manifold.

($y_i \neq y_j$), all possible sequences of transformations between $\mathbf{x}_i$ and $\mathbf{x}_j$ are either very long (i.e. $k \gg 1$), or are not well connected (i.e. the connecting path contains at least one weak link formed by two distant patterns).

The first assumption is built upon the standard manifold assumption. In particular, it supposes that most local intra-class variations can be covered by a small number of parameters. The second assumption is equivalent to the standard cluster assumption, and its formulation is similar to the concepts of local and global consistency discussed in [10].

The applied transformations T_i of a consistent sequence should have limited flexibility, since otherwise any two patterns could be transformed one into another by a single transformation, thus losing the idea of a connected path. As we will see, the idea of *connected transformations* brings a new perspective on how the standard assumptions should be exploited in an efficient way, especially when only a few labeled data are available.

4. Classification through connected image transformations

According to the model of connected transformations given by Eq. (1), a *global* path-based distance measure should be computed by taking into account the whole sequence of *local* deformations or dissimilarities starting from an unlabeled pattern and reaching a labeled one. Therefore, the proposed classifier requires defining three stages or blocks: i) a local pairwise dissimilarity metric, ii) a global distance that measures the length of the path through all connected image transformations, and iii) a final classification step based on the proposed global distance.

In this paper we mainly focus on the first two stages that provide us with a robust density-based metric for semi-supervised classification. In particular,

once a suitable distance has been computed, any of the nearest-neighbor based techniques can be used for classification. For simplicity we choose the 1-NN classifier that selects the class of the closest (in the sense of the proposed path-based metric) labeled example.

In the following we review the used local dissimilarity measure and the global path-based distance.

4.1. Dissimilarity based on local deformations

While the Euclidean distance between patterns is often used as a dissimilarity measure in graph-based semi-supervised classification problems (see for instance [7, 8, 9, 10, 12, 13, 17]), it is not necessarily the most suitable local dissimilarity. Especially when using nearest-neighbor classifiers in high-dimensional data sets, it is well-known that all pairwise Euclidean distances seem to be similar. This observation is sometimes referred to as the "concentration phenomenon" in the pattern recognition literature [18], or the "sphere-hardening effect" in other fields [19].

Furthermore, in image classification problems it is necessary to use distances that are invariant to certain transformations of the input. In this sense, the limitations of the Euclidean distance can be illustrated with a simple example. Suppose we are given two images that are identical except for the fact that one image is shifted one or more pixels to the right. Although these images are visually very similar, the Euclidean distance between vector representations of these images will report a high dissimilarity. More generally, we are interested in a dissimilarity measure that allows to compensate for small geometric intra-class variations while retaining the larger inter-class differences.

The literature on image deformation models is vast, ranging from elastic matching techniques [20] to shape contour models [21]. In addition to being flexible enough (but not too flexible), the chosen transformation model should be computationally efficient. As a good tradeoff between all these requirements, we use the image distortion model (IDM) proposed by Keysers et al. in [22]. This model has a very simple implementation and has been applied successfully in supervised handwritten character recognition, showing a generalization error of 0.54% on the complete MNIST benchmark data set.

To formally describe the IDM measure we adhere to the notational convention from [22]. Specifically, let us denote two images taken from the complete data set $\mathcal{X}$ as $\mathbf{a} = \{\mathbf{a}_{pq}\}$ and $\mathbf{b} = \{\mathbf{b}_{pq}\}$. The pixel positions are

input: images $\mathbf{a}$ and $\mathbf{b}$, parameter w .
Obtain all superpixels $\mathbf{a}_{pq}$ and $\mathbf{b}_{rs}$ of $\mathbf{a}$ and $\mathbf{b}$.
initialize
  $d = 0$
for $p = 1, 2, \ldots, P$ **do**
  for $q = 1, 2, \ldots, Q$ **do**
    $d = d + \min_{\substack{r \in \{1,\ldots,P\} \bigcap \{p-w,\ldots,p+w\} \\ s \in \{1,\ldots,Q\} \bigcap \{q-w,\ldots,q+w\}}} \|\mathbf{a}_{pq} - \mathbf{b}_{rs}\|^2$
  end for
end for
output: $d_{idm}(\mathbf{a}, \mathbf{b}) = d$.

Algorithm 1: Calculation of the image distortion model (IDM) dissimilarity measure from [22]. P and Q denote the image width and height in pixels, respectively.

indexed by $(p, q), p = 1, \ldots, P; q = 1, \ldots, Q$, where P and Q are the image width and height in pixels. In case the images differ in size, a scaling is taken into account and pixel correspondence is based on the closest position (see [22] for details). In general, $\mathbf{a}_{pq}, \mathbf{b}_{pq} \in \mathcal{R}^h$ are vectors (or *superpixels*) that can represent grey values ($h = 1$), color values ($h = 3$), the horizontal and vertical local image gradients ($h = 2$), or a larger pixel context. In our experiments we have always used superpixels of length $h = 18$, obtained from the horizontal and vertical image gradients on a 3×3 patch as computed by the horizontal and vertical Sobel filters, respectively (see [22]).

Algorithm 1 outlines the calculation of the IDM measure. Specifically, for each superpixel $\mathbf{a}_{pq}$ of image $\mathbf{a}$, it aims to find the optimally corresponding superpixel $\mathbf{b}_{rs}$ of image $\mathbf{b}$ within a local neighborhood limited by a warp range w. The IDM dissimilarity is then calculated as the conventional Euclidean distance between the superpixels of image $\mathbf{a}$ and the corresponding superpixels of image $\mathbf{b}$. Since the optimal warping of one superpixel in IDM does not affect the optimal warping of its neighboring superpixels, IDM is referred to as a zero-order model. Models of first and second order take into account one or two levels of neighboring pixels, which guarantees a smoother warping and, in consequence, reduces its matching flexibility. However, they typically introduce a much higher computational burden.

An important observation on IDM is that it is not a symmetric dissimilarity measure. Furthermore, as preliminary experiments on the MNIST

data set pointed out, the differences between $d_{idm}(\mathbf{a}, \mathbf{b})$ and $d_{idm}(\mathbf{b}, \mathbf{a})$ can be substantial for certain pairs of examples, which indicates that IDM is not an equally reliable measure for all patterns in a data set. In order to avoid differences between $d_{idm}(\mathbf{a}, \mathbf{b})$ and $d_{idm}(\mathbf{b}, \mathbf{a})$ that are caused by an excess of flexibility in IDM, we choose to use a symmetrized "worst case" IDM as a local dissimilarity, which returns the higher dissimilarity in case of doubt,

$$d_l(\mathbf{x}_k, \mathbf{x}_l) = \max\left(d_{idm}(\mathbf{x}_k, \mathbf{x}_l), d_{idm}(\mathbf{x}_l, \mathbf{x}_k)\right). \tag{2}$$

This measure will be referred to as "symmetric" IDM in the following, and it is the local dissimilarity measure used throughout this paper.

4.2. The ρ-connectivity global distance measure

The proposed global distance is calculated as the shortest-path length between two patterns $\mathbf{x}_i$ and $\mathbf{x}_j$, in which the length of each segment of the path is measured by the local dissimilarity (2).

Formally, we define $p_{i,j}$ to be a path of length $|p_{i,j}|$ that connects patterns $\mathbf{x}_i$ and $\mathbf{x}_j$ through an arbitrary sequence of intermediate patterns: $p_{i,j} = \{\mathbf{x}_i, \ldots, \mathbf{x}_k, \ldots, \mathbf{x}_j\}$. All patterns composing a given path are in $\mathcal{X}$ without distinguishing between labeled and unlabeled patterns. We use the notation p_k to indicate the k-th pattern in the current path $|p|$, therefore $\mathbf{x}_{p_1} = \mathbf{x}_i$ and $\mathbf{x}_{p_{|p|}} = \mathbf{x}_j$.

The length of each segment $\{\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}\}$ of the path is measured as a weighted version of the local deformation between the two patterns that delimit the segment,

$$d_l^\rho(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}) = e^{\rho d_l(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}})}. \tag{3}$$

The parameter ρ controls how these small deformations are weighted along the path. We then define the global distance as the length of the shortest weighted path between $\mathbf{x}_i$ and $\mathbf{x}_j$,

$$d_g(\mathbf{x}_i, \mathbf{x}_j) = \min_{p \in P_{i,j}} \sum_{k=1}^{|p|-1} d_l^\rho(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}). \tag{4}$$

This measure can be calculated efficiently by using Dijkstra's algorithm on the transformed local dissimilarity (3).

The global distance (4) is inspired by the ρ-connectivity distance proposed in [14], which is defined as

$$d_c(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{\rho} \ln \left(1 + d_g(\mathbf{x}_i, \mathbf{x}_j) - |p|\right), \tag{5}$$

in which $|p|$ is the optimal-path length. This measure has the interesting property that for $\rho \to 0$ it reduces to the shortest path along all deformations without any weighting,

$$d_c(\mathbf{x}_i, \mathbf{x}_j)|_{\rho \to 0} = \min_{p \in P_{i,j}} \sum_{k=1}^{|p|-1} d_l(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}),$$

while for $\rho \to \infty$ only the worst link produced along the path is considered,

$$d_g(\mathbf{x}_i, \mathbf{x}_j)|_{\rho \to +\infty} = \min_{p \in P_{i,j}} \max_{1 \le k \le |p|} d_l(\mathbf{x}_{p_k}, \mathbf{x}_{p_{k+1}}),$$

which was originally proposed in [11] as a means to improve clustering algorithms. A proper selection of ρ yields a trade-off between these two extremes. In particular, a value of $\rho < \infty$ allows to de-noise the metric at a global scale and increases its robustness against outliers and bridge points. Since the global distance (4) and the ρ-connectivity distance (5) differ only by a monotonically increasing transformation, they yield the same result when used in nearest-neighbor searches. Hence, while we will use Eq. (4) in the following, the interpretation of the parameter ρ is the same as for the ρ-connectivity distance (5).

Finally, while IDM is not a metric, the global measure (4) obtained by combining the symmetric IDM local dissimilarity with the path-based global distance is a pseudo-metric. This is discussed in more detail in Appendix A.

4.3. Classification of unlabeled training data

Given the ρ-connectivity distance measure between image pairs, we apply the nearest-neighbor classifier that selects the class of the closest labeled example: an unlabeled pattern $\mathbf{x}_j$ is assigned to the class to which its closest labeled neighbor $\mathbf{x}_j^*$ belongs, which is found as

$$\mathbf{x}_j^* = \arg\min_{\mathbf{x}_i \in \mathcal{X}_l} d_g(\mathbf{x}_i, \mathbf{x}_j), \tag{6}$$

1: **input:** data set $\mathcal{X}$ and labels $\{y_1, \ldots, y_l\}$
2: **for each** $\mathbf{x}_i \in \mathcal{X}$, $\mathbf{x}_j \in \mathcal{X}_u$, **do**
3: Calculate IDM dissimilarity $d_s(\mathbf{x}_i, \mathbf{x}_j)$ with (2).
4: **end for**
5: **for each** $\mathbf{x}_j \in \mathcal{X}_u$, **do**
6: Retrieve the shortest path from $\mathbf{x}_j$ to a labeled pattern using (4). Store the corresponding label y_j^* and distance $d_g(\mathbf{x}_j^*, \mathbf{x}_j)$.
7: **end for**
8: **output:** Estimated labels $\{y_{l+1}^*, \ldots, y_n^*\}$ and corresponding distances.

Algorithm 2: Training of the Connected Image Transformations (CIT) classifier.

Table 1: Variables used for calculating the computational cost.

l	Total number of labeled patterns.
u	Total number of unlabeled patterns.
dim	Image dimension (number of pixels).
w	IDM warp range.

The corresponding label is estimated as $\hat{y}_j = y_j^*$ and the distance to this labeled pattern is $d_g(\mathbf{x}_j^*, \mathbf{x}_j)$.

An overview of the proposed CIT training algorithm is given in Algorithm 2. A Matlab implementation can be obtained from http://gtas.unican.es/people/steven.

4.4. Complexity analysis

The computational costs corresponding to the different operations of the CIT algorithm are represented in Table 2. Taking into mind the adopted notations, summarized in Table 1, the computational complexity can be broken down as follows:

1. The pattern matching procedure of IDM (the *for*-loop containing line 3 of Alg. 2) requires to calculate the IDM measure between all pairs of training data. Taking into account that the local dissimilarity is never computed between pairs of labeled data, the total amount of data pairs in this step is $u^2 + 2ul$. Each of these calculations requires to search a grid of $(2w + 1)^2$ neighbors for each of the dim superpixels

Table 2: Computational cost of the different operations (with references to the line numbers of Alg. 2).

IDM pixel matching	(line 3)	$O((u^2 + 2ul) \cdot dim \cdot w^2)$
Compute shortest paths	(line 6)	$O(u^2 + l)$

of the reference image. Since we are interested in semi-supervised scenarios in which $u \gg l$, the computational complexity of this step can be approximated as $\mathcal{O}(u^2 \cdot dim \cdot w^2)$. Simply put, the computational complexity is quadratic in terms of the number of available unlabeled training images, u, and linear in the image dimension, dim.

2. The calculation of the shortest paths with the ρ-connectivity distance (the *for*-loop containing line 6) assumes an implementation based on Dijkstra's algorithm [23], yielding a complexity of $\mathcal{O}(u^2+l)$ for a graph of u fully connected unlabeled patterns that are all connected to all l labeled patterns. Remember that the CIT classifier only requires computing the shortest path from any unlabeled pattern to all training patterns. In case the graph is not fully connected, the calculation of the ρ-connectivity distance measure can be sped up by implementing it as a binary or Fibonacci heap [14].

5. Out-of-sample extension

Given a pattern $\mathbf{x}_k$ that was not included in the training data set, we denote the closest labeled pattern as $\mathbf{x}_k^*$. It can be retrieved efficiently as $\mathbf{x}_k^* = \mathbf{x}_j^*$, where $\mathbf{x}_j^*$ is the labeled pattern closest to the the training pattern j identified by

$$j = \underset{j \in [1,\ldots,n]}{\arg\min} \left(d_g(\mathbf{x}_j^*, \mathbf{x}_j) + d_l^\rho(\mathbf{x}_j, \mathbf{x}_k) \right). \tag{7}$$

For j corresponding to labeled patterns, the distance $d_g(\mathbf{x}_j^*, \mathbf{x}_j)$ is zero, while for unlabeled data the distances $d_g(\mathbf{x}_j^*, \mathbf{x}_j)$ in Eq. (7) have been obtained and stored by the training algorithm. Nevertheless, the local dissimilarities to all u unlabeled training data, $d_l^\rho(\mathbf{x}_j, \mathbf{x}_k)$, still require to be calculated. This calculation can represent an important computational burden in case many testing data are provided. Specifically, in order to classify one out-of-sample datum, $2u$ calculations of IDM are required (see Eq. (7)). In the following we study an approximation that allows to speed up the out-of-sample extension.

5.1. Prototype-based approximation

The bottleneck operation of the proposed method is the computation of the IDM measure between data points. The number of IDM calculations required amounts to $u^2 + 2ul$ for training the CIT classifier and $2u$ for classifying one out-of-sample pattern. Our optimized C subroutine of IDM requires 0.4ms to calculate one IDM dissimilarity between 28×28 images of the MNIST database on a dual core 3GHz Pentium IV PC running Matlab under Windows 7. For $n = 2000$ training patterns, with $l = 100$ and $u = 1900$, training the entire algorithm takes 1464s, out of which 1448s correspond to calculating IDM and 9.4s are required by the Dijkstra algorithm. Testing one out-of-sample datum takes 1.6s.

While designing a speedup for the training procedure of CIT is not a straightforward task, a speedup in the out-of-sample classification can be obtained by implementing a very small adjustment to the original CIT training algorithm. In particular, a sparsification procedure can be obtained as follows. We assume that the data distribution has a simple underlying structure. We wish to identify a representative subgraph characterized by a reduced number of training patterns, or *prototypes*, that allow to represent the structure of this subgraph sufficiently well, according to a suitable criterion. Since the proposed method is based on connectivity, it is reasonable to use connectivity as the criterion to construct the representative subgraph as well. A straightforward way to measure the degree of connectivity of each training pattern is the number of times it is used in a path during training of the original graph, which can be obtained by slightly extending the Dijkstra algorithm underlying the path retrieval. In other words, patterns that are part of many paths are considered well-connected, and we denote them as prototypes. Patterns that are only part of only one or few paths are considered badly-connected and they can be pruned. It is unlikely that they will be needed in out-of-sample classification, and they only raise the computation time. In summary, the entire graph is calculated and the subgraph is obtained from it by stripping off the patterns that are part of the fewest paths. Depending on the specific requirements of the classification problem, we can either aim to obtain a fixed-complexity classifier, in which case only a fixed number of the best-connected patterns is retrieved, or we can aim for a certain classification precision, in which case we choose a threshold for the number of connections and we select only prototypes that exceed this threshold. Once a set of suitable prototypes is obtained, the out-of-sample classification (7) only requires to calculate IDM between the test points and

the selected prototypes.

To illustrate that the number of prototypes, n_p, can be significantly lower than the number of training data for sufficiently rich data sets, consider the following example. We take 2000 patterns from the MNIST database and randomly choose 10 labeled instances per class. We train the CIT algorithm on these data to obtain the connectivity graph and the number of paths that go through each pattern. The algorithm's parameter is chosen as $\rho = 20$, which is a standard value used throughout the experiments (see Section 6). The second column of Table 3 shows the number of paths that pass through each training pattern. Interestingly, more than half of the data only contribute to one connection in the calculated paths. More specifically, the only path they form part of is the path that connects them with the closest labeled patterns. In the third column we display the results for a second experiment with $n = 5000$, where similar conclusions can be drawn. Therefore, if a speedup in the out-of-sample classification test phase of the algorithm is required, these patterns can be removed from the graph. While a small error might be introduced in the performance by doing so, it allows to reduce the testing computation of the algorithm to less than half.

6. Experimental results

6.1. Data sets

In order to understand the scope of the algorithm's applicability, we conducted experiments on four popular image data sets (see Table 4). All patterns used in the experiments are grayscale images.

The modified National Institute of Standards and Technology (MNIST) database is the standard benchmark for handwritten character recognition[1] [24]. This database contains a very large number of training and test data from 10 digit classes, 60000 and 10000 patterns, respectively. Each pattern is a 28×28 gray-valued image that has been preprocessed by normalization and centering. State-of-the-art supervised classification techniques obtain classification errors well below 1% when trained on all available MNIST training data. In particular, the k-NN classifier using IDM dissimilarities [22] obtains 0.54% error rate; the large-convolutional-net based classifier from [25] and the multi-layer perceptron classifier from [26] obtain 0.39% and 0.35%,

[1] MNIST: http://yann.lecun.com/exdb/mnist

Table 3: The number of patterns that are part of a specified number of paths in MNIST, with 10 labeled patterns per class.

# of paths	$n = 2000$	$n = 5000$
1	1138	2919
2	324	747
3	129	380
4	90	216
5	61	125
6:10	125	284
11:25	88	198
26:50	34	64
51:100	9	35
101:200	2	25
201:400	0	7

respectively, both of which use elastic distortions. On the other hand, semi-supervised techniques are designed to exploit the information in the unlabeled data that might be available. As a result, they require far less training data to reach acceptable error rates. For the present experiments only very small subsets of labeled data are used.

The Columbia University Image Library (COIL-20) data set is a collection of pictures of 20 different objects[2] [27]. Each object has been placed on a turntable and an image was obtained at every 5 degrees of rotation. Each picture was cropped to remove black borders and rescaled to 32x32 pixels. Some images of the first class of COIL-20 can be seen in Fig. 3. The UMIST data set is a collection of pictures of 20 individuals [3]. Each individual is shown in a range of poses from profile to frontal views, with variations with respect to facial expressions, gender, appearance and lighting. Finally, the

[2]COIL-20: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[3]UMIST: http://www.sheffield.ac.uk/eee/research/iel/research/face

Table 4: Benchmark data sets used in the experiments.

Data set	Classes	Size	Dimensions
MNIST	10	60000 + 10000	28 x 28
COIL-20	20	1440	32 x 32
UMIST	20	564	27 x 32
ORL face database	40	400	28 x 34

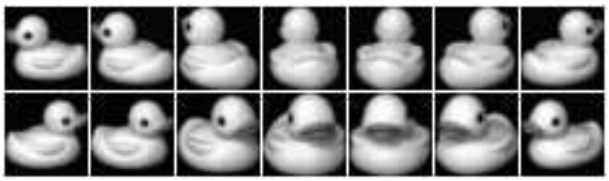


Figure 3: Images of the first class of COIL-20.

ORL face database contains 10 different images of each of 40 individuals[4] [28]. The images were taken at different times and show considerable variations in lighting, facial expressions and facial details (see Fig. 4). While for MNIST separate training and test data sets are available, for the COIL-20, UMIST and ORL face data sets the training and test data were selected from a single pool per database.

6.2. Analysis of CIT

We first describe parameter selection and results of the CIT algorithm on the MNIST data set.

6.2.1. Parameter selection

As discussed earlier, the principal parameter of the CIT classifier is the connectivity-related value ρ. The other two parameters are the superpixel dimension h and the warp-range w, which are only used to calculate the

[4]ORL Face Database: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

Figure 4: Sample images of the ORL face database.

IDM dissimilarity measure. We fix these last two parameters in advance, in particular $h = 18$ and $w = 2$, which are standard values used in [22].

The optimal value of ρ depends on the particular data set. Since we are dealing with scenarios in which only very few labeled training points are available, cross-validation is not a recommended technique to determine ρ as it would lower the number of usable labeled data even more in order to create a validation set. Fortunately, we have observed experimentally that optimal performance is typically achieved for a wide range of values. Fig. 5 shows the classification test error rate versus ρ for the MNIST database, calculated on all 10000 test patterns after training the algorithm with 2000 randomly selected training patterns, consisting of 10 labeled patterns per class and 1900 total unlabeled data. A similar behavior has been observed for other numbers of labeled and unlabeled patterns, and for other databases. In general, higher values of ρ assign more confidence to the weakest link in a path, while lower values can be used to average out several less reliable connections. Since the error rates are similar for any ρ between 10 and 200, we chose to use a generic value of $\rho = 20$ for the MNIST database.

The illustrations of Figures 6 and 7 allow to analyze the decisions taken by the proposed CIT classifier more closely. They show the connected transformation paths that are followed to classify some example digits. Specifically, the first image in each path is the unlabeled image to be classified, and the last image corresponds to the labeled pattern that is calculated to be the closest, according to the proposed metric. Fig. 6 shows the paths followed for a number of correctly classified digits, while several erroneous paths are illustrated in Fig. 7. As a visual inspection confirms, all neighboring images in these sequences are very similar. Interestingly, however, the erroneous connections show the highest dissimilarity in the sequence, which indicates that CIT at least identifies the erroneous connection as the weakest link in these cases.

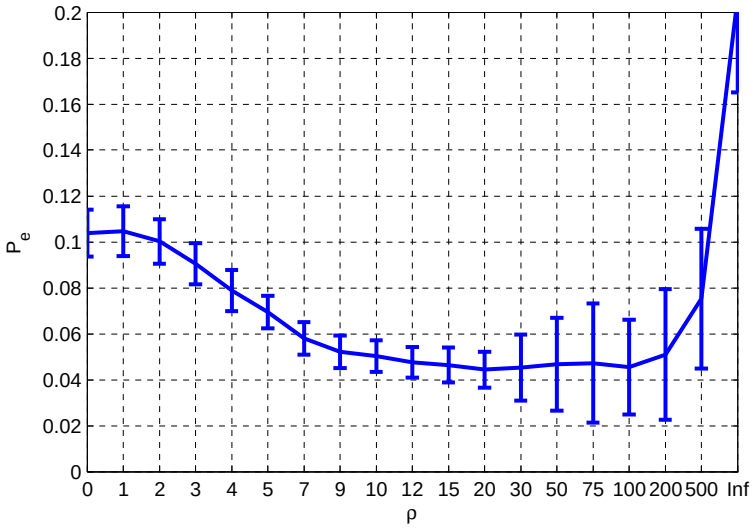


Figure 5: CIT test error probability versus connectivity parameter ρ on the MNIST database with $n = 2000$.

6.2.2. Out-of-sample classification for large data sets

Next, we analyze the prototype-based approximation for out-of-sample classification described in Section 5.1. The previous experiment was repeated, with $\rho = 20$. Out of $n = 2000$ training patterns, we retained different numbers of prototype patterns, n_p, by selecting only the best-connected patterns. Only these prototypes were used to classify the selected test patterns. As Fig. 8 shows, the test error does not increase if the 1000 least-relevant training patterns are discarded (i.e. if $n_p = 1000$), while this more than halves the computation time of the out-of-sample classification. Very similar results are still obtained after discarding 1500 patterns and retaining only $n_p = 500$ prototypes.

In order to analyze these findings we plotted the number of connections that pass through each training point in Table 3, for one experiment with $n = 2000$ and another one with $n = 5000$. Interestingly, more than half of the data only contribute to one connection in the calculated paths. More specifically, the only path they form part of is the path that connects them with the closest labeled point. Since it is unlikely that these data will contribute anything to the algorithm, we will simply discard patterns with only one connection in the following experiments with the MNIST data set. By doing so, the

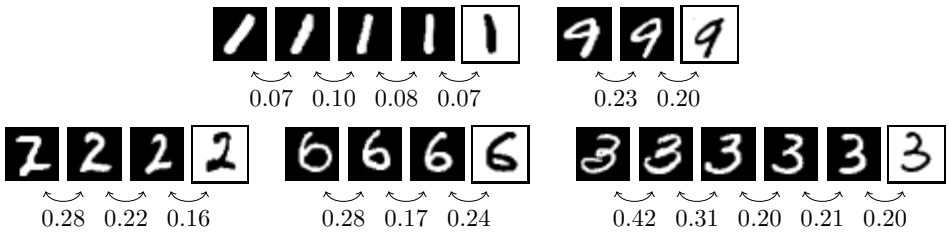


Figure 6: Examples of paths followed for correctly classified digits. Unlabeled and labeled digits are drawn respectively with black and white backgrounds. Under each connection the IDM local dissimilarity between both images is mentioned. Each path starts at the unlabeled image to be classified, and after following several connections (other unlabeled images) it reaches the closest labeled image.

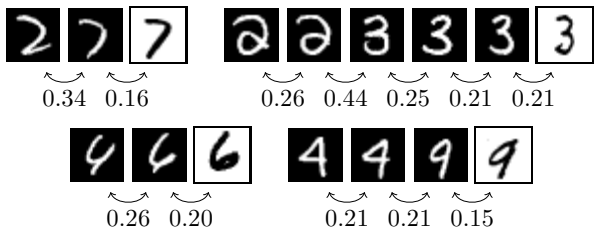


Figure 7: Examples of paths followed for incorrectly classified digits.

testing of the algorithm requires less than half of the computation, and the data manifold modeled by the algorithm becomes somewhat de-noised.

6.3. Performance comparison

We proceed to conduct experiments with the rest of the data sets described in Table 4, comparing the proposed algorithm to three state-of-the-art algorithms from literature.

6.3.1. Algorithm descriptions and parameter selection

The applied algorithms are the following:

1. LDS: The Low Density Separation (LDS) algorithm from [14] introduces the ρ-connectivity distance measure to detect clusters that are separated by regions of low density, using the Euclidean distance as a

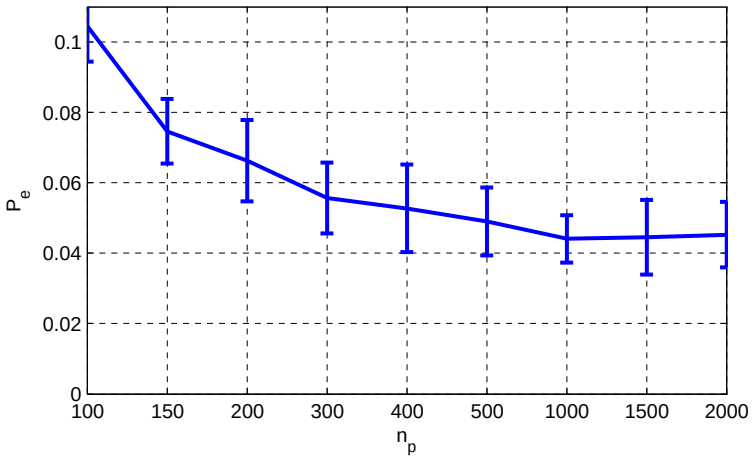


Figure 8: CIT test error probability versus number of prototype patterns (out of 2000 total training patterns).

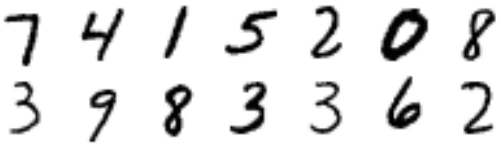


Figure 9: The 12 best-connected prototypes corresponding to the experiment using $n = 5000$ in Table 3.

local dissimilarity measure. On the resulting distance matrix it performs multi-dimensional scaling, and finally classifies unlabeled data using a transductive SVM classifier.

2. LapSVMP: The Laplacian SVM trained in the Primal. This algorithm exploits the geometry of the underlying marginal distribution to regularize the data manifold [16]. We used the implementation in which Newton's method was employed for solving the convex optimization problem, which has cubic complexity, instead of the faster preconditioned conjugate gradient, which is faster but obtains slightly weaker results (see [16]). Since LapSVMP is a binary classifier and all experiments are multi-class classification problems, the one-against-all approach has been performed for this algorithm.

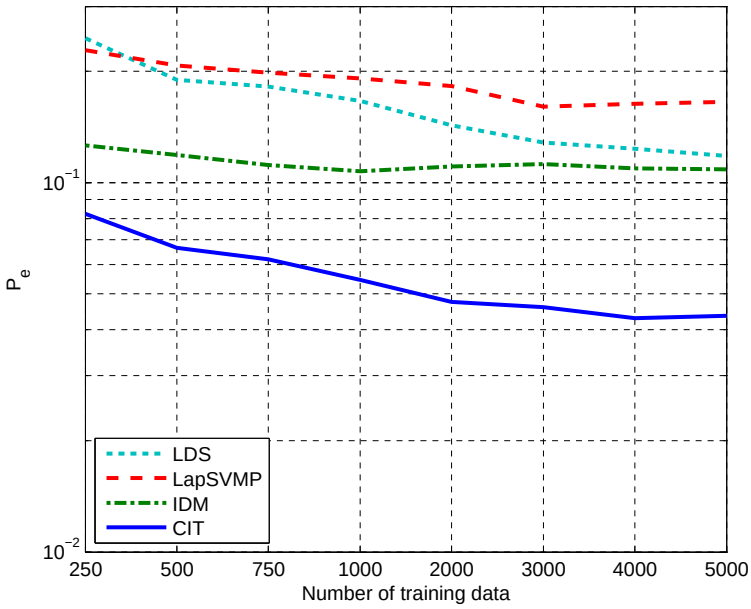


Figure 10: Error probabilities on the unlabeled training data, for different numbers of total training data n.

3. IDM: In [22] the Image Distortion Model (IDM) dissimilarity measure was applied to construct a 1-NN classifier. Although it was proposed as a supervised classifier, we include its results to demonstrate that it is able to outperform some semi-supervised classifiers when only very few labeled patterns are available.
4. CIT: The proposed Connected Image Transformations (CIT) classifier as described in Alg. 2.

The parameters for LDS and LapSVMP were obtained by a grid search over the intervals reported in [14] and [16], respectively, and we used the parameters that minimized their error rates. For reproducibility the selected parameter values are reported in Appendix B.

6.3.2. Influence of the training data size

Fig. 10 illustrates the error probability on the unlabeled training data for all four algorithms on the MNIST data set, for different numbers of training data. 10 labeled data per class were used. The advantage of CIT over the other algorithms is significant. Furthermore, along the entire sweep

Table 5: The number of data points in each split of the selected data set. l and u indicate the labeled and unlabeled training data sizes, respectively, whereas t is the number of out-of-sample test data.

Data set	l	u	t
MNIST	100	1900	10000
COIL-20	40	1000	400
UMIST	40	424	100
ORL Face database	80	220	100

range, IDM performs better than the semi-supervised techniques LDS and LapSVMP. This is a somewhat surprising result, since IDM is applied in a supervised manner and hence ignores the information contained in the unlabeled data. Notice that the results of IDM do not depend on n. Nevertheless, this observation confirms one of the main theses of this paper, in particular that not all the emphasis should be put only on designing a suitable global metric.

6.3.3. Performance comparison

Next, we compare the performance of the four algorithms on all four described data sets. Details of the used data splits are collected in Table 5. The results for these splits are given in Table 6. CIT outperforms all other methods on these data sets.

In order to analyze these results more closely, we extend them by varying the number of labeled training data per class. The number of total training patterns, n, is kept constant. The results are shown in Fig. 11. Clearly, the advantage of using connectivity in addition to the IDM measure is seen in all four experiments as the difference between the performance of the IDM algorithm and the CIT algorithm. In Fig. 11a IDM obtains reasonable results on the MNIST data. Since IDM does not take into account the unlabeled data, this results suggests that the labeled data already sample the manifold sufficiently. By adding connectivity, the CIT algorithm obtains much lower error rates. LDS and LapSVMP obtain fairly bad results. In the case of LapSVMP, this might be partly due to the one-vs-all multi-class strategy.

Results on the COIL-20 data set are represented in Fig. 11b. Unlike the previous experiment, IDM obtains very weak results on these data. This could be explained by the particularities of the data itself: in this set, the main difference between images of the same class is the angle under which each picture was taken (see Fig. 3). Therefore, all data from one class lie close to a one-dimensional manifold that is parameterized by this angle, favoring path-based algorithms such as LDS. The performance gain of CIT over the state-of-the-art semi-supervised classifiers LDS and LapSVMP is worth noting in this experiment.

Fig. 11c presents results on the UMIST data, which consist of pictures of individuals in which the main intra-class differences are due to different camera angles, similar to the COIL-20 data. While the same trend is visible in the results, CIT's advantage is smaller, probably because the number of training data points is about half that of the COIL-20 experiment. LapSVMP and LDS also perform slightly worse here. IDM is not affected by the number of unlabeled training data, and performs slightly better than in the previous experiment.

Finally, Fig. 11d illustrates the results on the ORL face database. With the exception of LDS, all algorithms obtain similar results on this set. This can be explained by the following observations. First, very few images per class are available in the training data, compared to the other data sets (see Table 4). Second, since all individuals were facing the camera for this data set, the inter-class differences in this set are much smaller. This last observation implies that there is no clear low-density separation between classes, which explains the weak results for LDS. As CIT combines parts of LDS with IDM, its improvement over IDM is minimal here. Notice also that, since n is chosen fixed, as the number of labeled patterns goes up there are less unlabeled training patterns available. Hence, as the number of labeled patterns goes towards 7 (for which $l = 280$ and $u = 20$), this experiment becomes predominantly a supervised experiment and the results for CIT and IDM are identical.

7. Conclusions

We have proposed a semi-supervised image classification algorithm that is capable of operating with only very few labeled data available. The algorithm builds upon the assumption that any image can be obtained as a slight transformation of another sufficiently close image of the same class. As

a result, in a semi-supervised classification scenario with enough unlabeled data available, class membership can be determined by considering the labeled point that can be reached by the shortest sequence of such Connected Image Transformations (CIT).

The manner in which the distance is measured through a series of connected image transformation turns out to be a fundamental aspect of the proposed classifier. We have proposed a distance that combines a local dissimilarity measure (the image distortion model proposed in [22]) with a global connectivity-based distance measure (the ρ-connectivity proposed in [14]). In semi-supervised image classification problems, especially when very few labeled data are available, the proposed metric squeezes the distances among patterns belonging to the same class, while leaving them almost undistorted in the low-density zones between classes. In this way, we have a de-noised metric that is robust to outliers or sparsely populated regions of the class manifolds and, in addition, improves the class-separability.

The features of the proposed classifier have been experimentally corroborated on four popular image databases. On each of the tested benchmarks the proposed algorithm obtains excellent results, often significantly outperforming state-of-the-art semi-supervised classifiers.

7.1. Further lines and extensions

Although the proposed classifier is easy to implement, its training phase requires the calculation of the IDM dissimilarity between all pairs of available data points, which can be computationally costly. In order to speed up this procedure an approximate nearest-neighbor technique could be designed, inspired by [29, 30, 31, 32], as discussed in Section 5.1.

Other future work will be dedicated to improving the local dissimilarity measure, first by considering smoother deformations, and, more importantly, by learning the most adequate underlying local metric for a specific problem. To this end, convex optimization techniques on the training data could be employed, as in [33].

Finally, the proposed technique could be extended with an on-line formulation, in order to update the classifier as new data arrive. We will consider these extensions as future research lines.

Appendix A. CIT is a pseudo-metric

In this appendix we prove that the distance measure consisting of the ρ-connectivity distance calculated on the symmetrized IDM similarity measure is a pseudo-metric on the set of all available data patterns, $\mathcal{X} = \mathcal{X}_l \bigcup \mathcal{X}_u$. Formally, the function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a pseudo-metric on the set $\mathcal{X}$ if it fulfills the following conditions for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{X}$ [34]:

1. Non-negativity: $d(\mathbf{x}, \mathbf{y}) \geq 0$.
2. Symmetry: $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$.
3. Triangle inequality: $d(\mathbf{x}, \mathbf{z}) \leq d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{z})$.

In order to be a metric, an additional identity condition is imposed:

4. Identity: $d(\mathbf{x}, \mathbf{y}) = 0 \iff \mathbf{x} = \mathbf{y}$.

First of all, notice that the IDM measure as originally proposed in [14] is not a metric, since it only fulfills the first condition. It is not symmetric, and, as can be seen easily from Fig. A.12, it does not fulfill the triangle inequality. IDM can be symmetrized as in Eq. (2), but the resulting measure still does not fulfill the triangle inequality. However, the proposed measure that combines the local dissimilarity of symmetrized IDM with a global ρ-connectivity distance measure, as given in Eq. (4), does fulfill all conditions to be a pseudo-metric.

Proof. 1. Since symmetrized IDM is nonnegative, each term in the sum (4) is nonnegative. Therefore, the result is also nonnegative.
2. Symmetrized IDM is symmetric by construction. Therefore, the sum (4) is also symmetric.
3. The global distance measure (4) considers the shortest path between any two patterns $\mathbf{x}$ and $\mathbf{z}$. Given a third pattern $\mathbf{y}$, the sum $d_g(\mathbf{x}, \mathbf{y}) + d_g(\mathbf{y}, \mathbf{z})$ is equal to $d_g(\mathbf{x}, \mathbf{z})$ only if $\mathbf{y}$ is part of the optimal path between $\mathbf{x}$ and $\mathbf{z}$. Otherwise it is larger by construction. Therefore we have $d_g(\mathbf{x}, \mathbf{z}) \leq d_g(\mathbf{x}, \mathbf{y}) + d_g(\mathbf{y}, \mathbf{z})$.

This concludes the proof.

Appendix B. Parameters used in the experiments

Table B.13 collects all the parameters selected during the experiments. The warp range w and the superpixel dimension h were chosen fixed in IDM and CIT. The other parameters were optimized.

Acknowledgments

This work was supported by MICINN (Spanish Ministry for Science and Innovation) under grants TEC2010-19545-C04-03 (COSIMA) and CONSOLIDER-INGENIO 2010 CSD2008-00010 (COMONSENS).

References

[1] T. Joachims, Text categorization with support vector machines: Learning with many relevant features, in: European Conference on Machine Learning (ECML), Springer, Berlin, 1998, pp. 137–142.

[2] J. Weston, C. Leslie, D. Zhou, A. Elisseeff, W. S. Noble, Semi-supervised protein classification using cluster kernels, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems (NIPS) 16, MIT Press, Cambridge, MA, 2004, pp. 3241–3247.

[3] S. Rajan, J. Ghosh, M. Crawford, An active learning approach to hyperspectral data classification, IEEE Transactions on Geoscience and Remote Sensing 46 (2008) 1231–1242.

[4] X. Zhu, Semi-Supervised Learning Literature Survey, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[5] I. Cohen, F. Cozman, N. Sebe, M. Cirelo, T. Huang, Semisupervised learning of classifiers: Theory, algorithms, and their application to human-computer interaction, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1553–1566.

[6] F. Wang, C. Zhang, Robust self-tuning semi-supervised learning, Neurocomputing 70 (2007) 2931 – 2939.

[7] T. S. Jaakkola, M. Szummer, Partially labeled classification with markov random walks, in: T. G. Dietterich, S. Becker, Z. Ghahramani (Eds.), Advances in Neural Information Processing Systems (NIPS) 14, MIT Press, Cambridge, MA, 2002, pp. 945–952.

[8] A. D. Szlam, M. Maggioni, R. R. Coifman, Regularization on graphs with function-adapted diffusion processes, Journal of Machine Learning Research 9 (2008) 1711–1739.

[9] X. Zhu, Z. Ghahramani, J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the Twentieth International Conference on Machine Learning (ICML), Washington, DC, USA, pp. 912–919.

[10] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems (NIPS) 16, MIT Press, Cambridge, MA, 2004, pp. 321–328.

[11] B. Fischer, V. Roth, J. M. Buhmann, Clustering with the connectivity kernel, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems (NIPS) 16, MIT Press, Cambridge, MA, 2004, pp. 89–96.

[12] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, Journal of Machine Learning Research 7 (2006) 2399–2434.

[13] M. Belkin, P. Niyogi, Using manifold structure for partially labeled classification, in: S. T. S. Becker, K. Obermayer (Eds.), Advances in Neural Information Processing Systems (NIPS) 15, MIT Press, Cambridge, MA, 2002, pp. 929–936.

[14] O. Chapelle, A. Zien, Semi-supervised classification by low density separation, in: Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS), pp. 57–64.

[15] S. Van Vaerenbergh, I. Santamaria, P. Barbano, Semi-supervised handwritten digit recognition using very few labeled data, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Prague, Czech Republic, pp. 2136–2139.

[16] S. Melacci, M. Belkin, Laplacian support vector machines trained in the primal, Journal of Machine Learning Research 12 (2011) 1149–1184.

[17] O. Chapelle, B. Schölkopf, A. Zien (Eds.), Semi-Supervised Learning, MIT Press, Cambridge, MA, 2006.

[18] D. Francois, V. Wertz, M. Verleysen, The concentration of fractional distances, IEEE Transactions on Knowledge and Data Engineering 19 (2007) 873 –886.

[19] E. Biglieri, Coding for Wireless Channels, Kluwer Academic Publishers, Norwell, MA, 2005.

[20] S. Uchida, H. Sakoe, Eigen-deformations for elastic matching based handwritten recognition, Pattern Recognition 36 (2003) 2031–2040.

[21] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 509–522.

[22] D. Keysers, T. Deselaers, C. Gollan, H. Ney, Deformation models for image recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 1422–1435.

[23] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269–271.

[24] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.

[25] M. Ranzato, C. Poultney, S. Chopra, Y. LeCun, Efficient learning of sparse representations with an energy-based model, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems (NIPS) 19, MIT Press, Cambridge, MA, 2007, pp. 1137–1144.

[26] D. Ciresan, U. Meier, L. Gambardella, J. Schmidhuber, Deep big simple neural nets excel on handwritten digit recognition, Arxiv preprint arXiv:1003.0358 (2010).

[27] S. A. Nene, S. K. Nayar, H. Murase, Columbia object image library (COIL-20), Technical Report CUCS-005-96, Department of Computer Science, Columbia University, New York, 1996.

[28] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, in: Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on, pp. 138 –142.

[29] G. Shakhnarovich, T. Darrell, P. Indyk, Nearest-Neighbor Methods in Learning and Vision: Theory and Practice, The MIT Press, 2006.

[30] J. Bourgain, On lipschitz embedding of finite metric spaces in hilbert space, Israel Journal of Mathematics 52 (1985) 46–52.

[31] C. Faloutsos, K. Lin, Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets, in: Proceedings of the 1995 ACM SIGMOD international conference on Management of data, ACM, pp. 163–174.

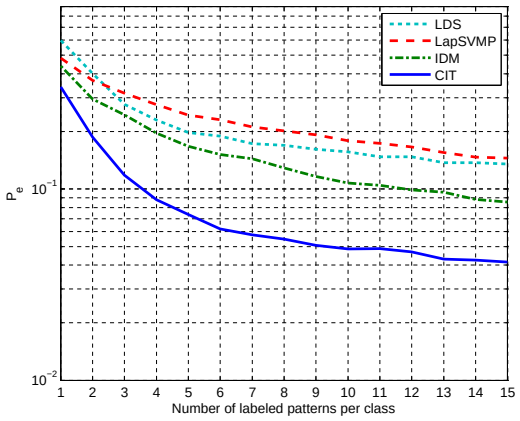
[32] V. Athitsos, J. Alon, S. Sclaroff, G. Kollios, Boostmap: An embedding method for efficient nearest neighbor retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 89 –104.

[33] K. Q. Weinberger, F. Sha, L. K. Saul, Convex optimizations for distance metric learning and pattern classification, IEEE Signal Processing Magazine 27 (2010) 146–158.

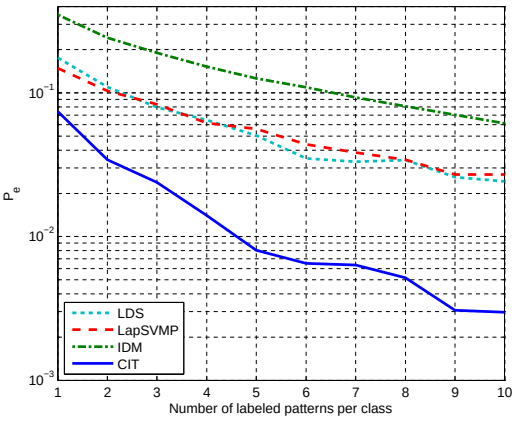
[34] L. Steen, J. Seebach, Counterexamples in Topology, Courier Dover Publications, 1995.

Table 6: Algorithm performance on different data sets, using the split sizes of Table 5. All values are expressed in percentages. Values between parentheses indicate standard deviations. LDS does not allow out-of-sample testing as it is a transductive method. The results for LapSVMP coincide with the results reported in [16], on the COIL-20 set.

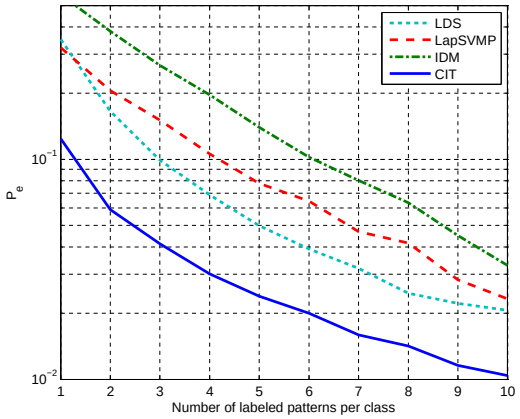
Data set	**Algorithm**	**Training error**	**Test error**
MNIST	LDS	15.62 (1.82)	n/a
	LapSVMP	17.89 (1.92)	19.24 (1.22)
	IDM	10.76 (1.14)	10.02 (0.93)
	CIT	**4.86** (**0.98**)	**4.57** (**0.96**)
COIL-20	LDS	11.04 (1.89)	n/a
	LapSVMP	10.31 (2.32)	11.79 (2.87)
	IDM	24.18 (1.93)	24.50 (2.40)
	CIT	**3.44** (**2.20**)	**3.49** (**2.45**)
UMIST	LDS	16.53 (4.03)	n/a
	LapSVMP	20.56 (3.32)	21.00 (4.26)
	IDM	38.21 (4.07)	38.29 (5.81)
	CIT	**5.90** (**2.45**)	**6.11** (**3.31**)
ORL Face	LDS	14.92 (3.47)	n/a
	LapSVMP	13.61 (2.73)	13.82 (3.99)
	IDM	12.20 (2.66)	12.97 (4.15)
	CIT	**8.57** (**2.76**)	**8.91** (**3.08**)



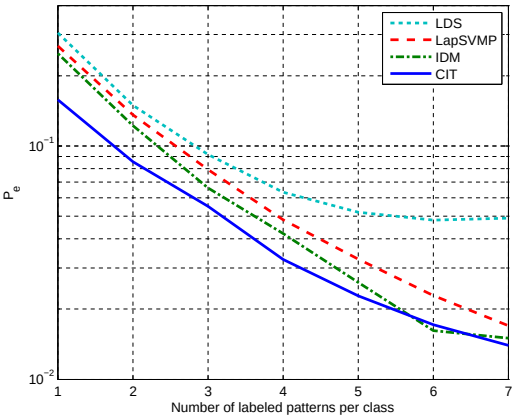
(a) MNIST: $n = 2000$, $N = 10$.



(b) COIL-20: $n = 1040$, $N = 20$.



(c) UMIST: $n = 464$, $N = 20$.



(d) ORL face database: $n = 300$, $N = 40$.

Figure 11: Simulation results on the data sets of Table 4.

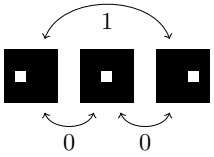


Figure A.12: IDM does not fulfill the triangle inequality: The middle image can be obtained from any of the two other images by shifting the white pixel horizontally over 1 position. Therefore, IDM with warp range $w = 1$ yields zero dissimilarity on these image pairs. However, the differences between the left and right images fall outside of its warp range, yielding a non-zero IDM dissimilarity, here generically represented as 1.

Data set	σ	nn	p	γ_A	γ_I
MNIST	9	10	2	10^{-2}	10^{-1}
COIL-20	0.6	2	1	10^{-6}	1
UMIST	9	5	2	10^{-2}	10^{-6}
ORL face database	9	5	2	10^{-6}	10^{-4}

(a) Parameters used for LapSVMP

Data set	σ	nn	C	ρ
MNIST	∞	100	100	4
COIL-20	∞	0	100	10
UMIST	∞	0	100	10
ORL face database	∞	0	100	5

(b) Parameters used for LDS

Data set	$[h]$	$[w]$
MNIST	18	2
COIL-20	18	2
UMIST	18	2
ORL face database	18	2

(c) Parameters used for IDM

Data set	$[h]$	$[w]$	ρ
MNIST	18	2	20
COIL-20	18	2	50
UMIST	18	2	50
ORL face database	18	2	20

(d) Parameters used for CIT

Figure B.13: Parameters used for each of the algorithms in the different experiments. Parameters in square brackets were chosen fixed.