Feature Learning by Multidimensional Scaling and its Applications in Object Recognition

Quan Wang   Kim L. Boyer
Signal Analysis and Machine Perception Laboratory
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180, USA
wangq10@rpi.edu   kim@ecse.rpi.edu

***Abstract*—We present the MDS feature learning framework, in which multidimensional scaling (MDS) is applied on high-level pairwise image distances to learn fixed-length vector representations of images. The aspects of the images that are captured by the learned features, which we call MDS features, completely depend on what kind of image distance measurement is employed. With properly selected semantics-sensitive image distances, the MDS features provide rich semantic information about the images that is not captured by other feature extraction techniques. In our work, we introduce the iterated Levenberg-Marquardt algorithm for solving MDS, and study the MDS feature learning with IMage Euclidean Distance (IMED) and Spatial Pyramid Matching (SPM) distance. We present experiments on both synthetic data and real images — the publicly accessible UIUC car image dataset. The MDS features based on SPM distance achieve exceptional performance for the car recognition task.**

***Keywords*-Feature learning; image distance measurement; multidimensional scaling; spatial pyramid matching**

I. Introduction

To represent an image by a fixed-length feature vector, there are generally two groups of approaches, often referred to as hand-designed features and feature learning, respectively. In this section, we briefly review several commonly used methods from each group, and relate the proposed MDS feature learning to these existing methods.

A. Hand-Designed Features

Most hand-designed features, or sometimes called handcrafted features, focus on capturing the color, texture and gradient information in the image. Generally, these features have a closed form to be computed, without looking at other images. Some popular yet simple hand-designed image features include color-histogram, wavelet transform coefficients [1], scale-invariant feature transform (SIFT) [2], color-SIFT [3], speeded up robust features (SURF) [4], histogram of oriented gradients (HOG) [5] and local binary patterns (LBP) [6]. To represent an image with one fixed-length feature vector, there are generally three ways: (1) First, these features can be computed for the entire image, but the resulting feature vector will fail to embed the spatial relationship between different objects or different locations in the image. (2) Second, the image can be first uniformly divided into $M \times N$ blocks. Then these features can be computed for each block, and can be concatenated to make a long feature vector. (3) Further, the division of the image does not have to be uniform, but can be arbitrary. We can just put random rectangular or circular masks onto the image, and compute features for each mask (or "patch"), then concatenate. To do this, the division must be consistent for all images.

The divide-and-concatenate methods will result in very large feature vectors. Given a large dataset, PCA can be used to reduce the dimensionality.

B. Feature Learning

Feature learning has often been used as a synonym of deep learning, especially in recent years, and often refers to recent techniques such as sparse coding [7], [8], auto-encoder [9], convolutional neural networks [10], restricted Boltzmann machines [11], and deep Boltzmann machines [12]. However, we believe this interpretation of feature learning is literally imprecise. Feature learning should be more generally defined as the opposite to hand-designed features — it should refer to any technique that learns a fixed-length vector representation of each image in the dataset by utilizing the pattern distribution of the entire dataset, or optimizing a target function that is defined on the entire dataset. Any technique that can generate a feature representation of each image without looking at the entire dataset should fail to fall into this category.

We further categorize existing feature learning methods into two subgroups: feature learning with raw intensities, and feature learning with hand-designed features. The proposed MDS feature learning falls into a third new subgroup: feature learning with image distance measurement.

1) Feature Learning with Raw Intensities: This subgroup of methods treat the feature learning problem as a dimensionality reduction problem, where the original high-dimensional data are the image intensities, either gray-level or RGB values. Efforts on data dimensionality reduction have a long history [13], dating from the early work on PCA [14] and its nonlinear form, kernel PCA [15], to the recent work on sparse coding and deep learning [7]–[12]. In all these methods, high dimensional data, such as an image, is represented by a low dimensional vector. Each entry of this vector describes one salient varying pattern of all images within the training set.

Assume we have a dataset $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)^\mathsf{T}$, where each $\mathbf{x}_i$ ($1 \leq i \leq N$) is one data point. We briefly review several dimensionality reduction methods below.

- **PCA** linearly projects vector $\mathbf{x}$ to $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ is obtained by performing eigenvector decomposition on the covariance matrix $\mathbf{S}_{\mathbf{x}} = \mathbf{X}\mathbf{X}^{\mathsf{T}}$.
- **Kernel PCA** first constructs a kernel matrix $\mathbf{K}$, where each entry of this matrix is obtained by evaluating the kernel function $k(\cdot, \cdot)$ on two data points:

$$\mathbf{K}_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j). \tag{1}$$

  Then the Gram matrix is constructed as

$$\widetilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N\mathbf{K} - \mathbf{K}\mathbf{1}_N + \mathbf{1}_N\mathbf{K}\mathbf{1}_N, \tag{2}$$

  where $\mathbf{1}_N$ is the $N \times N$ matrix with all elements equal to $1/N$. Next the eigenvector decomposition problem $\widetilde{\mathbf{K}}\mathbf{a}_l = \lambda_l N \mathbf{a}_l$ is solved ($\mathbf{a}_l$ is eigenvector and λ_l is eigenvalue) and the projected vector $\mathbf{y}$ is computed by

$$y_l = \sum_{i=1}^{N} a_{li} k(\mathbf{x}, \mathbf{x}_i). \tag{3}$$

- **Auto-encoders** first normalize all $\mathbf{x}_i$'s to $[0, 1]$, and map them to $\mathbf{y}_i = s(\mathbf{W}\mathbf{x}_i + \mathbf{b})$, where $s(\cdot)$ is a sigmoid function. A reconstruction is computed by $\mathbf{z}_i = s(\mathbf{W}'\mathbf{y}_i + \mathbf{b}')$. The weight matrices $\mathbf{W}$ and $\mathbf{W}'$, and the bias vectors $\mathbf{b}$ and $\mathbf{b}'$ are obtained by minimizing the average reconstruction error, which can be defined as either traditional square error or cross-entropy.

In PCA and kernel PCA, different entries of $\mathbf{y}$ correspond to eigenvectors of different importance, while in auto-encoder, they are equivalently important.

These techniques have been shown effective on problems such as face recognition [16], [17] and even concept recognition [18]. However, most of these methods require all input data to have exactly the same size. If the input is an image, then the image has to be cropped and resized to be consistent with other images in the dataset. However, cropping the image means loss of information, and resizing the image means change of aspect ratio, which will result in distorted object shapes.

2) Feature Learning with Hand-Designed Features: One popular method that falls into this subgroup is the bag-of-visual-words (BOV) method [19]–[21]. This method first divides the image into local patches or segments the image into distinct regions, and then extracts hand-designed features for each patch/region. Rather than being concatenated, these feature vectors make an unordered set, or also referred to as "bag". By performing clustering on the union of all those unordered sets for all training images, a visual vocabulary is established. Now the set of feature vectors previously extracted from each image can be transformed into a "word-frequency" histogram by simply counting which cluster (visual word) is assigned to each patch/region. The "word-frequency" histogram can be optionally normalized to generate the final fixed-length vector.

One extension of BOV is the Fisher Vector (FV) method [22], [23]. Rather than simply counting the word frequency, which can be viewed as the 0-order statistics, FV encodes higher order statistics (up to the second order) about the distribution of local descriptors assigned to each visual word. Another extension is the Spatial Pyramid Matching [24], which gives different weights to features in different image division levels, and defines an image similarity measurement using the pyramid matching kernel [25].

II. Method

In this section, we first review the basics of MDS and its existing solutions, and then introduce our own solution — the iterated Levenberg-Marquardt algorithm (ILMA). Next, we discuss and compare some popular image distance measurement techniques in recent literature.

A. Multidimensional Scaling: Problem Definition

As a statistical technique for the analysis of data similarity or dissimilarity, multidimensional scaling (MDS) has been well applied to areas such as information visualization [26] and surface flattening [27], [28]. Here we briefly review the basic concepts and definitions of MDS. For convenience, we will use the word "image" instead of "data" or "object" in the context, but we keep in mind that MDS is a technique for general purposes.

Suppose we have a set of N images $\Omega = \{I_1, I_2, ..., I_N\}$, and there is a distance measurement $d(I_i, I_j)$ defined between each pair of images I_i and I_j. Note that $d : \Omega \times \Omega \to \mathbb{R}_{\geq 0}$ is only a measurement of image dissimilarity, not necessarily a metric on set Ω in the strict sense, since the subadditivity triangle inequality does not necessarily hold. Multidimensional scaling is the problem of representing each image $I_i \in \Omega$ by a point (vector) in a low dimensional space $\mathbf{x}_i \in \mathbb{R}^m$, such that the interpoint Euclidean distance in some sense approximates the distance between the corresponding images [29]. In Section II-C we will discuss how to define the image distance/dissimilarity measurement. Here we focus on the mathematical definitions related to MDS.

For a pair of images I_i and I_j, let their low dimensional (m-d) representations be $\mathbf{x}_i$ and $\mathbf{x}_j$. The representation error is defined as $e_{ij} = d(I_i, I_j) - ||\mathbf{x}_i - \mathbf{x}_j||$, where $||\cdot||$ denotes the L^2-norm. The *raw stress* is defined as the sum-of-squares of the representation errors:

$$\text{Stress}^* = \sum_{1 \leq i < j \leq N} e_{ij}^2, \tag{4}$$

while the *normalized stress* (also known as *Stress-1*) is defined as

$$\text{Stress-1} = \sqrt{\frac{\sum\limits_{1 \leq i < j \leq N} e_{ij}^2}{\sum\limits_{1 \leq i < j \leq N} ||\mathbf{x}_i - \mathbf{x}_j||^2}}. \tag{5}$$

MDS models require the interpoint Euclidean distances to be "as equal as possible" to the image distances. Thus we can either minimize the raw stress or normalized stress. We compactly represent the image distances by an $N \times N$ symmetric matrix $\mathbf{D} = [d(I_i, I_j)]_{N \times N}$ with all diagonal values equal to 0, and represent the low dimensional vectors by an $N \times m$

matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\mathsf{T}$. Using the raw stress as the loss function, the MDS problem can be stated as:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{1 \le i < j \le N} \left(d(I_i, I_j) - ||\mathbf{x}_i - \mathbf{x}_j||\right)^2 . \qquad (6)$$

B. Solutions for Multidimensional Scaling

There are lots of existing methods for solving Eq. (6), such as Kruskal's iterative steepest descent approach [29] and de Leeuw's iterative majorization algorithm (SMACOF) [30]. In 2002, Williams demonstrated the connection between kernel PCA and metric MDS [31], thus metric MDS problems can also be solved by solving kernel PCA.

In our work, we introduce an iterative least squares solution to the MDS optimization problem. We note that in Eq. (6), the raw stress is minimized with respect to $\mathbf{X}$, which has $N \times m$ entries in total. Thus, when N is large, this nonlinear optimization problem becomes computationally intractable if we attempt to solve for all entries in one step. Inspired by the iterated conditional modes (ICM) method [32], which was developed to solve Markov random fields (MRF), we introduce the two-stage *iterated Levenberg-Marquardt algorithm (ILMA)*. The basic idea of this algorithm is to repeatedly minimize the raw stress with respect to one $\mathbf{x}_i$ while holding all other $\mathbf{x}_i$'s fixed. For this purpose, we maintain a constraining set of the indices of the $\mathbf{x}_i$'s to be fixed. In the initialization stage, indices of all images are selected into the constraining set in a random order. In the adjustment stage, we repeatedly adjust all $\mathbf{x}_i$'s in a randomly permuted order. By doing so, each time we only need to minimize the raw stress with respect to m variables, instead of $N \times m$, which greatly reduces the complexity of the problem. The subproblem can be viewed as a least squares problem, and can be solved by the standard Levenberg-Marquardt algorithm [33], [34]. Since the total raw stress is monotonically non-increasing through time, the convergence of the adjustment is guaranteed. The details of the two-stage algorithm are given in Algorithm 1. We will call the low dimensional vectors $\{\mathbf{x}_i\}$ as *MDS features* or *MDS codes* in the context.

One advantage of our method is that we provide a unified framework for both MDS model training and new data encoding. In MDS model training, pairwise image distances are measured within the training set Ω_{train}, and Algorithm 1 is applied to encode each training image $I_i \in \Omega_{\text{train}}$ to its MDS code $\mathbf{x}_i$. Now given a new image $\widetilde{I}$, we measure the distance from this image to all training images $d(\widetilde{I}, I_i)$, and find its MDS code $\widetilde{\mathbf{x}}$ by:

$$\min_{\widetilde{\mathbf{x}}} \sum_{I_i \in \Omega_{\text{train}}} \left(||\widetilde{\mathbf{x}} - \mathbf{x}_i|| - d(\widetilde{I}, I_i)\right)^2, \qquad (7)$$

which can be directly solved as a least squares problem using the standard Levenberg-Marquardt algorithm. We follow this practice for the training and testing of MDS models in the experiment in Section III-B.

Algorithm 1: The two-stage iterated Levenberg-Marquardt algorithm (ILMA).

```
input  : Distance matrix $\mathbf{D} = [D_{i,j}]$, $1 \le i, j \le N$;
         Max number of iterations $M$;
output: MDS codes $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^\mathsf{T}$;
begin initialization stage:
Randomly choose a non-diagonal entry $D_{i_0,j_0}$ in $\mathbf{D}$;
Set $\mathbf{x}_{i_0} = (0, 0, \ldots, 0)^\mathsf{T}$;
Set $\mathbf{x}_{j_0} = (D_{i_0,j_0}, 0, \ldots, 0)^\mathsf{T}$;
Initialize the constraining set $A = \{i_0, j_0\}$;
while $|A| < N$ do
    Randomly choose $j^* \notin A$;
    Use the standard Levenberg-Marquardt algorithm to
    find the $\mathbf{x}_{j^*}$ that minimizes:
        $\sum_{i \in A} (||\mathbf{x}_{j^*} - \mathbf{x}_i|| - D_{j^*,i})^2$;
    Add $j^*$ to $A$, set $\mathbf{x}_{j^*}$ in $\mathbf{X}$;
end
begin adjustment stage:
for $t \leftarrow 1$ to $M$ do
    Generate a random permutation $(p_1, p_2, \ldots, p_N)$ of
    integers 1 to $N$;
    for $s \leftarrow 1$ to $N$ do
        Take $p_s$ out of $A$;
        Use the standard Levenberg-Marquardt algorithm
        to find the $\mathbf{x}_{p_s}$ that minimizes:
            $\sum_{i \in A} (||\mathbf{x}_{p_s} - \mathbf{x}_i|| - D_{p_s,i})^2$;
        Add $p_s$ back to $A$, update $\mathbf{x}_{p_s}$ in $\mathbf{X}$;
    end
    if $\Delta\text{Stress}^*/\text{Stress}^* < \epsilon$ then break;
end
done
```

C. Image Distance Measurement

The measurement of the similarity or dissimilarity between two images is of essential significance in content-based image retrieval [35], [36]. There are some very simple forms of image distances, such as the traditional Euclidean distance on raw image intensities, and the earth mover's distance (EMD) on image color histograms [37]. Here, we briefly describe two popular image distance measurement methods: the IMage Euclidean Distance (IMED) [38] and the Spatial Pyramid Matching (SPM) distance [24]. These distances will be evaluated in our experiment on real images in Section III-B.

1) IMED: The IMED is a generalized form of the traditional Euclidean distance on raw image intensities. Give two gray-level images I_1 and I_2 of the same size, the traditional Euclidean distance is defined as the square root of the sum-of-squares of intensity difference at each corresponding image location:

$$d^2_{\text{Euclidean}}(I_1, I_2) = \sum_{(r,c)} \left(I_1^{(r,c)} - I_2^{(r,c)}\right)^2 , \qquad (8)$$

where $I_1^{(r,c)}$ denotes the intensity at row r and column c in image I_1. In contrast, IMED also counts for the intensity difference at different locations, but assigns a weight to it, which is a function of the Euclidean distance of the two locations:

$$d^2_{\mathrm{IMED}}(I_1, I_2) = \sum_{(r,c)} \sum_{(r',c')} \left(I_1^{(r,c)} - I_2^{(r,c)} \right) \cdot \\ g(r,c,r',c') \cdot \left(I_1^{(r',c')} - I_2^{(r',c')} \right), \quad (9)$$

where

$$g(r,c,r',c') = f(\sqrt{(r-r')^2 + (c-c')^2}), \quad (10)$$

and $f(\cdot)$ is a continuous monotonically decreasing function, usually the Gaussian function. An interesting observation by Wang *et al.* [38] is that the IMED (9) on two images is equivalent to the traditional Euclidean distance (8) on a blurred version of the two images. The blur operation is called standardizing transform (ST) by the authors.

Although IMED has shown promising performance on some recognition experiments in [38], we can see that it is still a low-level image distance measurement, based on the raw intensities, without embedding any semantic information. Another disadvantage of IMED is that it is only defined on images of the same size. We will apply MDS on IMED distances for the experiment in Section III-B, where we use Gaussian function for $f(\cdot)$ in Eq. (10) and set $\sigma = 1$, and we call this method IMED-MDS.

2) SPM Distance: The spatial pyramid matching (SPM) [24] is based on Grauman and Darrell's work on pyramid matching kernel [25], which measures the similarity of two sets of feature vectors by partitioning the feature space on different levels and taking the sum of weighted histogram intersection functions. Lazebnik *et al.*'s spatial pyramid matching is an "orthogonal" approach — it performs pyramid matching in the 2-d image space, and uses k-means for clustering in the feature space (edge points and SIFT features). With a visual vocabulary of size M (number of clusters), and L partition levels, spatial pyramid vectors of dimensionality $M\frac{1}{3}(4^{L+1} - 1)$ are generated, and spatial pyramid matching similarities $K^L(I_i, I_j)$ between images I_i and I_j are measured. Authors of [24] recommend parameter setting of $M = 200$ and $L = 2$.

The similarity value $K^L(I_i, I_j)$ lies in $[0, 1]$, where 1 is for most similar, and 0 for least similar. We have many ways to define image distances using the similarities, such as:

$$\begin{aligned} d_{\mathrm{SPM1}}(I_i, I_j) &= 1 - K^L(I_i, I_j), & (11) \\ d_{\mathrm{SPM2}}(I_i, I_j) &= -\ln((1-\epsilon)K^L(I_i, I_j) + \epsilon), & (12) \end{aligned}$$

where ϵ is a small value. We set $\epsilon = 0.001$ in (12) for our experiment in Section III-B.

Unlike IMED, SPM distance is based on hand-designed features such as SIFT and edge points, instead of raw intensities. It models the spatial co-occurrence of different feature clusters, and thus is more semantics-sensitive. Besides, SPM distance does not require the size of images to be the same. We will apply MDS on the two SPM distances defined by Eq. (11) and Eq. (12), and we call them SPM1-MDS and SPM2-MDS, respectively.

III. Experiments

We present two experiments. The first one is on synthetic data, and is to evaluate the running time performance of different MDS algorithms, and to compare different initialization strategies of our iterated Levenberg-Marquardt algorithm. The second one is a real image object recognition task, in which we compare MDS features with PCA features and kernel PCA features. In the second experiment, we use the UIUC car dataset[1], and follow a five-fold cross validation to report the classification precision and recall under different feature dimensions.

A. Synthetic Data Experiment

In this experiment, we use MDS for curved surface flattening [27] on the manually created *Swiss roll* data, which was introduced in [39], and is known to be complicated due to the highly non-linear and non-Euclidean structure [40]. The Swiss roll surface contains 591 points in $\mathbb{R}^3$, as shown in Fig. 1. We measure the pairwise interpoint geodesic distances to construct a 591×591 distance matrix, and re-embed the Swiss roll surface into $\mathbb{R}^3$ by applying MDS on the geodesic distance matrix.

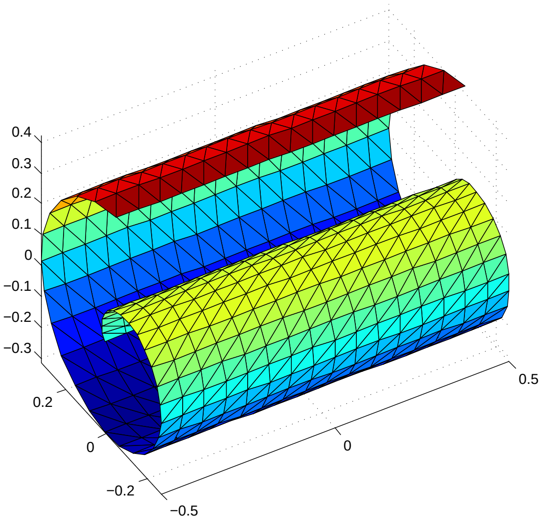


Fig. 1. The Swiss roll surface with 591 points.

1) Running Time: First, we would like to evaluate the running time performance of the proposed iterated Levenberg-Marquardt algorithm and compare with Bronstein's implementation of the SMACOF algorithm and its variants, including SMACOF with reduced rank extrapolation (RRE) and SMACOF with multigrid [40]–[43]. The results are given in Fig. 2, where each number in this plot is averaged on 20 independent repeated experiments, and the running time is reported on a

[1] http://cogcomp.cs.illinois.edu/Data/Car/

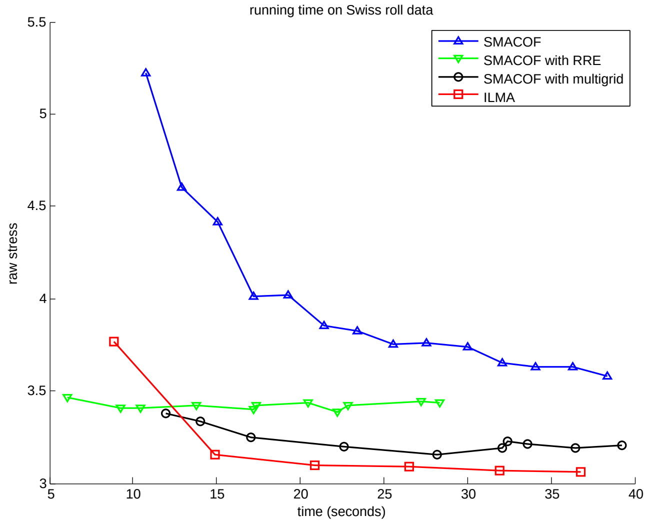


Fig. 2. The raw stress *vs.* running time plot of the SMACOF algorithm, its variants, and the proposed iterated Levenberg-Marquardt algorithm (ILMA) on the Swiss roll geodesic distance matrix.

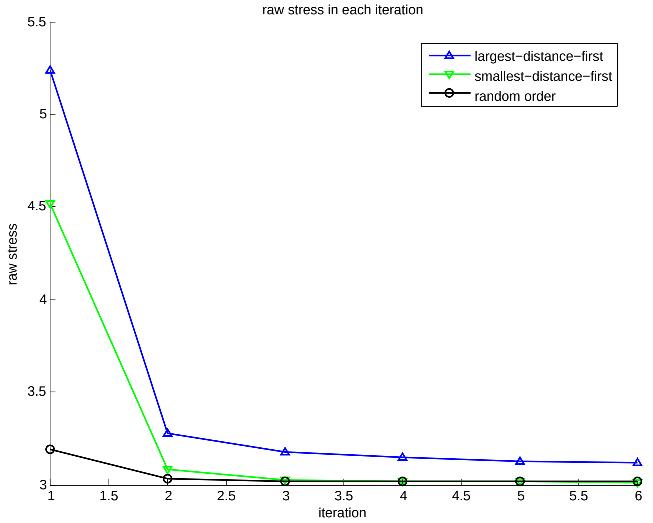


Fig. 3. The raw stress in each iteration of the iterated Levenberg-Marquardt algorithm with different initialization strategies.

Mac Pro with 2 × 2.4 GHz Quad-Core Intel Xeon CPU. From Fig. 2, we can see that our ILMA is an efficient solution, which runs faster and converges to a smaller raw stress value than other methods. The unrolled surfaces by ILMA in different iterations are shown in Fig. 4.

2) Initialization Strategies: Further, we study some modifications to Algorithm 1. The original algorithm uses a **random order strategy** in the initialization stage, but we can modify it to:

- **Largest-distance-first strategy:** For Algorithm 1, in line 2 we choose the largest non-diagonal entry D_{i_0,j_0} in $\mathbf{D}$ instead of a random one; in line 7, we find the $i^* \in A$ and $j^* \notin A$ that maximize D_{i^*,j^*} rather than a random $j^* \notin A$.
- **Smallest-distance-first strategy:** For Algorithm 1, in line 2 we choose the smallest non-diagonal entry D_{i_0,j_0} in $\mathbf{D}$; in line 7, we find the $i^* \in A$ and $j^* \notin A$ that minimize D_{i^*,j^*}.

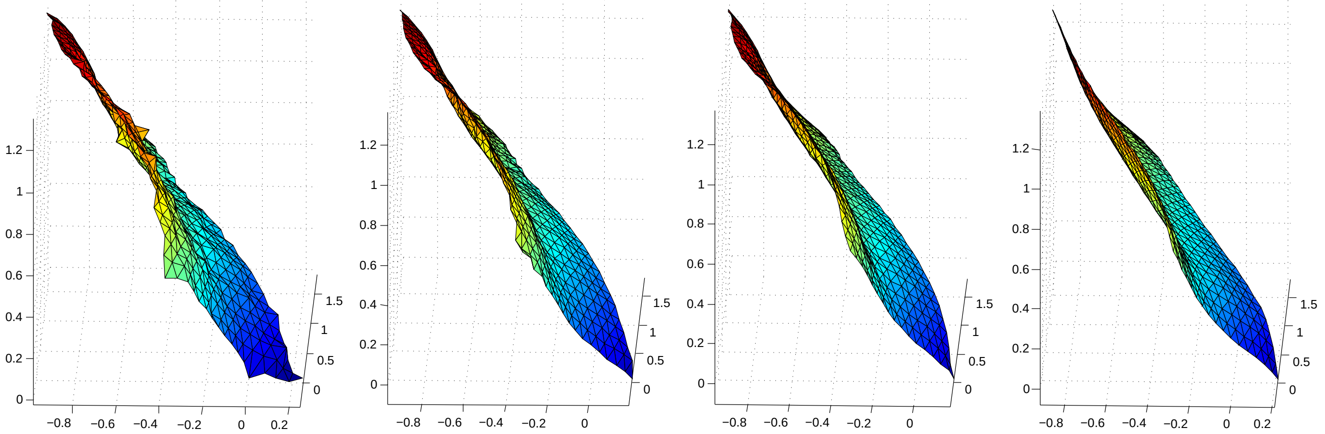
If we assume that the data to be encoded are comprised of clusters, then an intuitive interpretation of the largest-distance-first strategy is that representatives of each cluster are first encoded, and they are expected to be scattered in the multidimensional space; similarly, the smallest-distance-first strategy encodes all data in one cluster first, and then moves to the nearest cluster.

We have been using the three initialization strategies to solve the MDS problem on the Swiss roll geodesic distance matrix, and it turns out that the random order strategy converges faster than the other two, as shown in Fig. 3. Again, each number in this plot is averaged on 20 independent repeated experiments.

B. Car Recognition Experiment

Now we would like to compare the performance of MDS features to the most standard and popular dimensionality reduction algorithms — PCA [14] and kernel PCA [15] on raw pixel intensities. We use the UIUC car image dataset [44], which contains 550 car and 500 non-car gray-level images of size 40 × 100 (Fig. 5). We can observe that all car images are side-view images, but can be either side, and can be partly occluded. We divide the total of 1050 images into five subsets, each containing 110 car images and 100 non-car images, and each time we use four subsets as training set and one as testing set. We use the following methods to generate fixed-length feature vectors for the images:

1) **PCA** We represent each 40 × 100 gray-level image by a 4000-d vector, and perform standard PCA on such vectors of the training set to get eigenvectors and low dimensional representations of the training images. Then we use the eigenvectors to get the low dimensional representations of the testing images.
2) **kPCA Gaussian** Similar to the above method, but we use Gaussian kernel PCA instead of standard PCA. We follow the automatic parameter selection strategy in [45] to determine the σ.
3) **kPCA poly** Similar to the above two methods, but we use third-order polynomial kernel PCA instead of standard PCA.
4) **IMED-MDS** We first measure the IMED between each pair of training images, and run Algorithm 1 to learn the low dimensional MDS features of each training image. Then we measure the IMED from each testing image to each training image, and solve Eq. (7) to obtain the MDS features of each testing image.
5) **SPM1-MDS** Similar to the above method, but we use SPM1 distance (11), instead of IMED, where the SPM parameters are $M = 200$ and $L = 2$.
6) **SPM2-MDS** Similar to the above method, but we use SPM2 distance (12).
7) **pyramid PCA** Instead of computing MDS features from



(a) after initialization
(b) after 1 adjustment iteration
(c) after 2 adjustment iterations
(d) after 10 adjustment iterations

Fig. 4. Flattened Swiss roll surface by applying MDS with iterated Levenberg-Marquardt algorithm on the geodesic distance matrix.

Fig. 5. Example car (first row) and non-car (second row) images in UIUC car image dataset.

SPM distances, we can also directly perform PCA on the obtained $M\frac{1}{3}(4^{L+1}-1)$-dimensional spatial pyramid vectors without measuring similarities. In our experiment, we set $M = 200$ and $L = 2$, and the spatial pyramid vectors are 4200-d. Evaluating this method will allow us to observe whether the MDS on SPM distance measurement captures semantics beyond the spatial pyramids.

After we have obtained the fixed-length features of all images, we use the features of training images to learn a binary RBF kernel SVM [46], [47], and use it to classify the features of testing images. Each dimension of the feature vector is normalized to 0-mean and unit standard deviation. In the radial basis function $\exp(-||\mathbf{u}-\mathbf{v}||^2/\gamma)$, we set γ as the feature vector length. The experiment is repeated for different feature vector lengths from 1 to 20. We show the precision, recall and accuracy in Fig. 6. We also provide the feature scatter plots of different methods for feature length $m = 2$ in Fig. 7.

In Fig. 6, we can observe that IMED-MDS method performs slightly but not significantly better than directly applying PCA or kernel PCA on raw gray-level intensities, and the superiority of IMED-MDS is more obvious when feature dimension is low. Spatial pyramid based methods do perform much better than other methods. Especially, SPM1-MDS and SPM2-MDS methods outperform all other methods, including pyramid PCA, at all feature dimensions. While the precision and recall of PCA, kernel PCA and IMED-MDS methods saturate at 98% and 96% respectively, the precision and recall of SPM1-MDS and SPM2-MDS saturate at 100% and 99% respectively. At low feature dimensions ($m \leq 5$), the accuracy of PCA and kernel PCA are very low, but the SPM1-MDS and SPM2-MDS perform almost as equally well as at very high dimensions.

In Fig. 7, we can also see that SPM1-MDS and SPM2-MDS separate car and non-car images with very clear class boundary curves in 2-d feature space.

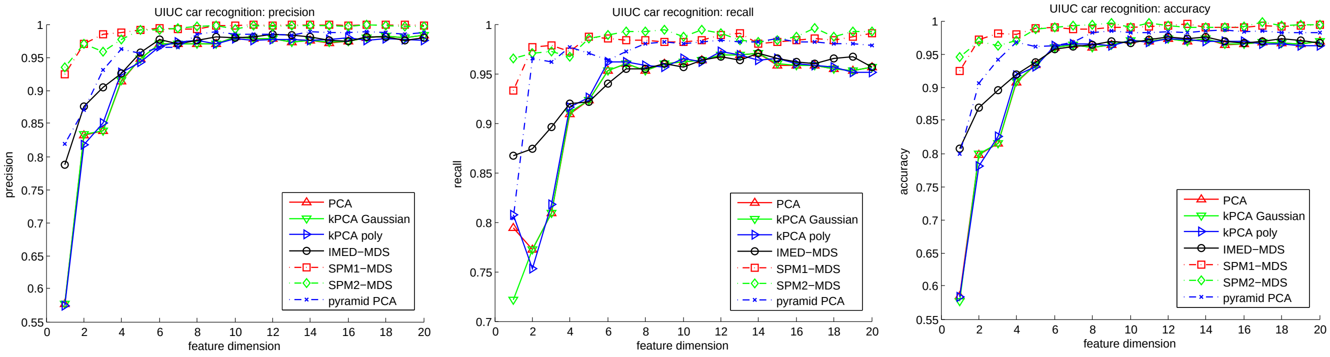
IV. Conclusions and Future Work

In this paper, we have presented a feature learning framework by combining multidimensional scaling with image distance measurement, and compared it with a number of popular existing feature extraction techniques. To the best of our knowledge, we are the first to explore MDS on image distances such as IMage Euclidean Distance (IMED) and Spatial Pyramid Matching (SPM) distance.

We have introduced a unified framework for both MDS model training and new data encoding based on the standard Levenberg-Marquardt algorithm. Our two-stage iterated Levenberg-Marquardt algorithm for MDS model training is an efficient solution, and has shown good running time performance compared with other off-the-shelf implementations (Fig. 2).

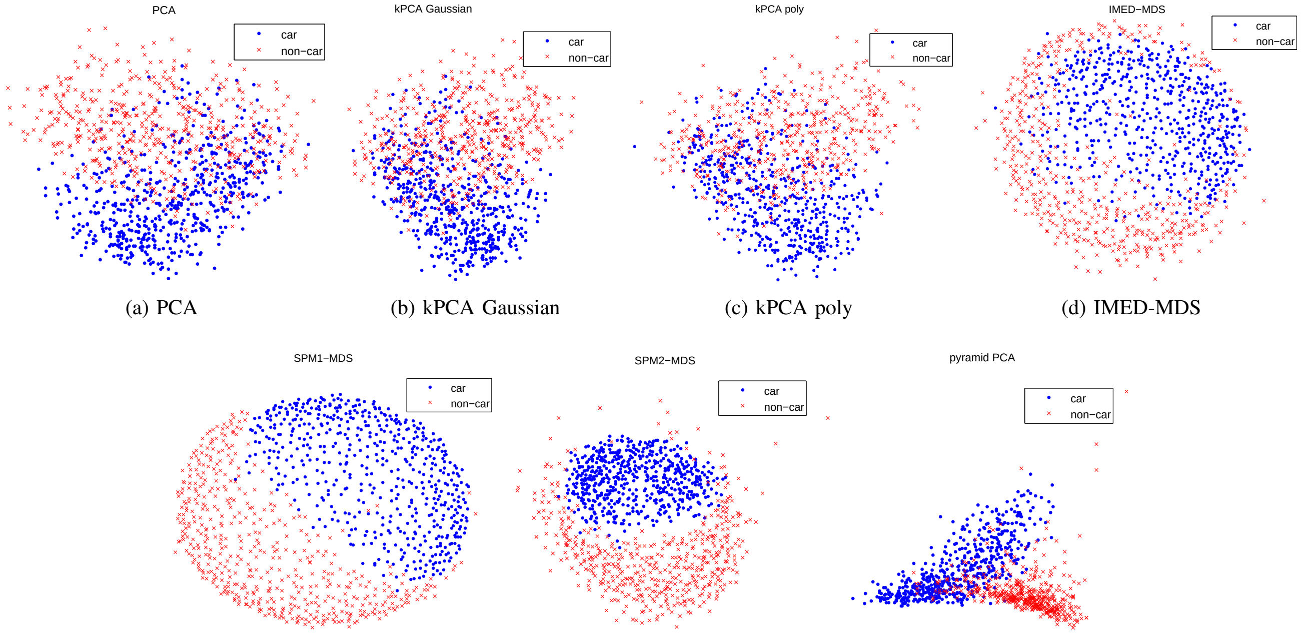
In the car recognition experiment, we have demonstrated the power of MDS features. MDS features learned from SPM distances achieve the best classification performance on all feature dimensions. The good performance of MDS features attributes to the semantics-sensitive image distance, since it captures very different information from the images than traditional feature extraction techniques. The MDS further embeds such information into a low-dimensional feature space, which also captures the inner structure of the entire dataset. The MDS embedding is a very necessary step, since in Fig. 6 we can see the performance of MDS features learned from SPM distances is significantly better than simply running PCA on spatial pyramid vectors.

Our ongoing work on this method explores these directions:



(a) precision (b) recall (c) accuracy

Fig. 6. UIUC car recognition performance.



(a) PCA (b) kPCA Gaussian (c) kPCA poly (d) IMED-MDS

(e) SPM1-MDS (f) SPM2-MDS (g) pyramid PCA

Fig. 7. 2-d feature scatter plots of UIUC car image dataset with different features.

1) We study more image distance measurements, such as the Integrated Region Matching (IRM) distance, which was originally designed for semantics-sensitive image retrieval systems [48]. Performance of MDS codes learned from such distances can be evaluated and compared with the SPM-MDS method in this paper.
2) Our MDS feature learning framework can be validated on larger datasets with many categories of color images of different sizes. For example, we can validate the methodology on the popular Caltech-101 dataset [49] or the COREL dataset [48], [50].
3) In Eq. (7), rather than using the entire training set, we can also use only a subset of the training images to encode new data. It would be interesting to see how the performance varies by applying different subset selection strategies and different sizes of the subset.
4) Currently the two-stage iterated Levenberg-Marquardt algorithm is implemented in MATLAB[2]. We are also recoding it in C/C++ with the lmfit library[3], which will be more computationally efficient.

References

[1] I. Daubechies *et al.*, *Ten lectures on wavelets*. SIAM, 1992, vol. 61.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] K. E. A. Van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[2] Code is available at https://sites.google.com/site/mdsfeature/

[3] http://joachimwuttke.de/lmfit/

[4] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2005, pp. 886–893.

[6] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[7] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 12, pp. 607–609, 1996.

[8] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems 19*, 2007, pp. 801–808.

[9] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[12] R. Salakhutdinov and G. E. Hinton, "Deep boltzmann machines," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 5, no. 2. MIT Press Cambridge, MA, 2009, pp. 448–455.

[13] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik, "Dimensionality Reduction: A Comparative Review," Tilburg University, Tech. Rep., 2009.

[14] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 2, pp. 559–572, 1901.

[15] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel pca and de-noising in feature spaces," in *Advances in Neural Information Processing Systems 11*. MIT Press, 1999, pp. 536–542.

[16] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[17] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[18] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng., "Building high-level features using large scale unsupervised learning," in *International Conference on Machine Learning*, June 2012.

[19] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Ninth IEEE International Conference on Computer Vision, 2003*. IEEE, 2003, pp. 1470–1477.

[20] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, 2004, p. 22.

[21] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005*, vol. 2. IEEE, 2005, pp. 524–531.

[22] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007*. IEEE, 2007, pp. 1–8.

[23] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," *Computer Vision–ECCV 2010*, pp. 143–156, 2010.

[24] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006*, vol. 2. IEEE, 2006, pp. 2169–2178.

[25] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Tenth IEEE International Conference on Computer Vision, 2005*, vol. 2. IEEE, 2005, pp. 1458–1465.

[26] I. Borg and P. J. F. Groenen, *Modern Multidimensional Scaling: Theory and Applications (Springer Series in Statistics)*, 2nd ed. Springer, 2005.

[27] E. L. Schwartz, A. Shaw, and E. Wolfson, "A numerical solution to the generalized mapmaker's problem: flattening nonconvex polyhedral surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 9, pp. 1005–1008, 1989.

[28] A. Elad and R. Kimmel, "On bending invariant signatures for surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1285–1295, 2003.

[29] J. Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," *Psychometrika*, vol. 29, pp. 1–27, 1964.

[30] J. de Leeuw, "Applications of convex analysis to multidimensional scaling," in *Recent Developments in Statistics*, J. Barra, F. Brodeau, G. Romier, and B. V. Cutsem, Eds. Amsterdam: North Holland Publishing Company, 1977, pp. 133–146.

[31] C. K. Williams, "On a connection between kernel pca and metric multidimensional scaling," *Machine Learning*, vol. 46, no. 1, pp. 11–19, 2002.

[32] J. Besag, "On the Statistical Analysis of Dirty Pictures," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 48, no. 3, pp. 259–302, 1986.

[33] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.

[34] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[35] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys (CSUR)*, vol. 40, no. 2, p. 5, 2008.

[36] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1349–1380, 2000.

[37] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[38] L. Wang, Y. Zhang, and J. Feng, "On the euclidean distance of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1334–1339, Aug. 2005.

[39] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[40] M. M. Bronstein, A. M. Bronstein, R. Kimmel, and I. Yavneh, "Multigrid multidimensional scaling," *Numerical linear algebra with applications*, vol. 13, no. 2-3, pp. 149–171, 2006.

[41] G. Rosman, A. M. Bronstein, M. M. Bronstein, A. Sidi, and R. Kimmel, "Fast multidimensional scaling using vector extrapolation," *SIAM J. Sci. Comput*, vol. 2, 2008.

[42] G. Rosman, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Topologically constrained isometric embedding," in *Human Motion*. Springer, 2008, pp. 243–262.

[43] A. M. Bronstein, M. Bronstein, M. M. Bronstein, and R. Kimmel, *Numerical geometry of non-rigid shapes*. Springer, 2008.

[44] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, Nov. 2004.

[45] Q. Wang, "Kernel principal component analysis and its applications in face recognition and active shape models," *arXiv preprint arXiv:1207.3538*, 2012.

[46] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[47] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[48] J. Wang, J. Li, and G. Wiederhold, "Simplicity: semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 9, pp. 947–963, Sep. 2001.

[49] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *Computer Vision and Pattern Recognition Workshop, 2004*. IEEE, 2004, pp. 178–178.

[50] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1088–1099, July 2006.