LEARNING PSEUDO METRICS FOR SEMANTIC IMAGE CLUSTERING

DIAN-HUI WANG\textsuperscript{1}, YONG SOO KIM\textsuperscript{2}

\textsuperscript{1}Department of Computer Science and Computer Engineering, La Trobe University, Melbourne, VIC 3086, Australia
\textsuperscript{2}College of Software, Kyungwon University, Seongnam, Gyeonggi-Do, 405-760, South Korea
E-MAIL: csdhwang@ieee.org

Abstract:
While people cluster images in terms of semantics, computers can not do too much on this job due to the use of lower-level features, which is the common way to computer workers to deal with the image recognition. To measure the closeness between feature vectors, similarity metrics play a key role and they directly impact the clustering performance. This paper develops a framework of learning pseudo metrics (LPM) for semantic image clustering practice. Multilayer perceptron (MLP) is employed to model the LPM with a set of criteria on evaluating the LPM quality and availability. Using a standard k-Mean clustering technique, a comparative study is carried out to demonstrate the significance of our proposed LPM in semantic image clustering. Experiments show that the LPM-based similarity metric can produce better clustering results in terms of both impurity and robustness.

Keywords:
Semantic images; clustering; learning pseudo metrics; neural networks

1. Introduction

Clustering is a typical task that involves partitioning a given data set into subsets based on certain similarity measure [2]. For semantic image clustering (SIC), the objective is to identify groups of images in a collection such that images with the same set of semantics are assigned to the same group [7]. Two images are said to be associative in a sense of semantic similarity if they share the same concept. That is, images in different clusters deal with relatively distinct subjects. The SIC techniques are useful to speed up the retrieval process in content-based image searching systems [8, 11]. If a large image database can be successfully grouped into different semantic clusters, a query may be quickly narrowed to a specific category. The SIC techniques can also help to perform visual data classification tasks [1,5]. With given examples and a proper similarity metric, some prototypes (cluster centers) can be generated firstly in a feature space. Then, images in the database are compared with these prototypes to determine the most appropriate class. This approach results in a knowledge-based classification system [5].

Each image in a database can be mapped into a point in a high-dimensional feature space. The similarity between two images is determined based on a metric function defined over the feature space of images, such as Euclidean distance, Manhattan distance, histogram intersection, or any other appropriate measurement method. However, the low-level features, such as color, shape and texture, sometimes do not precisely represent the semantics of the images. For instance, the feature vectors of some semantically relevant images may be located far away in the feature space if the Euclidean norm is employed as a similarity measure. Notice that the similarity nature for semantic images lies in relativity and subjectivity. Thus, any forms of metrics that are not derived from the data itself may be deficient to perform the similarity measure task well in terms of semantic relevance using low-level features. One promising approach to bridge the gap between low-level visual features and high-level semantics is to build a similarity measure directly from the data itself using machine learning techniques [5,10]. We termed such a distance-like functions learning similarity metric (LSM) [10], which is typically used in k-NN rule for classification tasks.

The remainder of this paper is organized as follows. Section 2 presents some mathematical preliminaries. Section 3 gives a mathematical definition on our proposed LPM concept with some remarks on implementation issues. Section 4 outlines our approaches for clustering quality assessment. Experiments are reported in Section 5, and Section 6 concludes this work.

2. Mathematical preliminary

This section gives some necessary mathematical concepts for supporting the paper development.

Definition 1 [6] Let $\mathcal{X}$ be a set of points, and $d$ be a real-valued function defined over the set $\mathcal{X} \times \mathcal{X}$. The
function $d$ is called a pseudo metric on $\chi$ if it satisfies the following three axioms, i.e., for all $x, y, z \in \chi$ there hold:

1. $d(x, x) = 0$ (the axiom of reflexivity),
2. $d(x, y) = d(y, x)$ (the axiom of symmetry),
3. $d(x, z) \leq d(x, y) + d(y, z)$ (the axiom of triangle inequality).

A pseudo metric space is just like a metric space, except that it is possible in a pseudo metric space for two points to be distinct and yet be zero distance apart. Every metric is also a pseudo metric; but not every pseudo metric space is a metric space.

Definition 2 [6] An equivalence relation on a set $\chi$ is a binary relation on $\chi$ that is reflexive, symmetric and transitive, i.e., if the relation is written as ~ it holds for all $x, y, z \in \chi$ that

1. $x \sim x$ (Reflexivity),
2. if $x \sim y$ then $y \sim x$ (Symmetry),
3. if $x \sim y$ and $y \sim z$ then $x \sim z$ (Transitivity).

Given a set $\chi$ and an equivalence relation ~ over $\chi$, an equivalence class is a subset of $\chi$ of the form ${x \in \chi | x \sim a}$ where $a$ is an element in $\chi$. This equivalence class is usually denoted as $[a]$; it consists of precisely those elements of $\chi$ which are equivalent to $a$. The following shows a way to define a pseudo metric through the existing equivalence classes.

Proposition 1. Given a set $\chi$ and a set of equivalence classes $\{[a_j], j = 1,2,..., p\}$ which are derived from an equivalence relation ~ over $\chi$. The function $f$ defined by (1) below is a pseudo metric over $\chi$.

$$f(x, y) = \begin{cases} 1 & \text{if } x \in [a_i] \text{ and } y \in [a_j], i \neq j \\ 0 & \text{if } x \in [a_i] \text{ and } y \in [a_j], i = j \end{cases}$$

3. Learning pseudo metric using neural networks

Let $S = \{S_1, S_2, ..., S_p\}$ be a set of visual data, $S_i$ be a subset of $S$, which represents a class of data characterized by a specific semantic concept, for instance, airplane, hours, woman. Denoted by $M$ an extractor which maps $S_i$ into its feature space, namely $F_i$, with a finite of dimension $n$, i.e.,

$$F_i = \{x = M(e) \in R^n : e \in S_i\}, k = 1: p.$$

Define a pattern pool as follows:

$$D = \{(x, y) \mapsto \delta_y : (x, y) \in F_i \times F_j, i, j = 1: p\},$$

where $\delta_y$ represents the Dirichlet sign function, and $\times$ is the Cartesian product operator.

Using the data from the pool $D$, learning machines, such as neural networks, can be trained to perform a generalization of the characteristic function of the equivalence relationship over $F = \{F_1, F_2, ..., F_p\}$. Obviously, the quality of this generalization depends on the richness and quality of the learning examples from the pattern pool. Denoted by $\sigma(x, y)$ the output (its value is constrained in $[0, 1]$) of a specific learning machine model for the input pair $(x, y)$. According to Definition 1 and Proposition 1 above, we now state our LPM concept.

Definition 3 Given a pattern pool $D$, a learning function $\sigma(x, y)$ is called as a LPM over $F$ if it satisfies the following conditions for a small positive value of $\mu$:

1. $\sigma(x, y) \leq \mu$ as $x$ and $y$ belong to a same class
2. $\sigma(x, y) \geq 1 - \mu$ as $x$ and $y$ belong to different classes
3. $|\sigma(x, y) - \sigma(y, x)| \leq \mu$ for any $x$ and $y$
4. $\sigma(x, y) \leq \sigma(x, z) + \sigma(y, z)$ as $x$ and $y$ belong to different classes, and for any $z$

Remark In Definition 3 above, it is not necessary to constrain the triangle inequality above for points from the same class. This is because we are concerned with conceptual similarity, rather than measuring the closeness between image features associated to the same semantic class.

Remark Although many existing models [3], such as feed-forward neural networks or support vector machines, can be employed to perform the modeling task, it seems not so easy to apply them in practice because of the large volume of training data. Data decomposition and modular modeling techniques are useful to resolving this problem. Generalization capability is a measure to evaluate the predictability of a model or a system for unseen inputs, which relates to the quality of knowledge extension. The generalization power can be measured by counting the percentage of the testing data pairs that satisfy the criteria in Definition 3.

4. Clustering quality assessment

This section discusses issues on the evaluation of clustering performance. In this paper, we consider two evaluation aspects for clustering system, that is, impurity and its robustness with respect to the model parameters.

Impurity function is referred to an error measure for testing the importance of a selected attribute in classification tree generation [4]. In this paper, we employ
this concept to characterize the quality of clustering performance for labeled data sets. The well-known impurity functions are the entropy function and the Gini index, which are respectively defined by
\[
\varphi_e(x_1, x_2, ..., x_p) = - \sum_{i=1}^{p} x_i \log_2 x_i ,
\]
\[
\varphi_g(x_1, x_2, ..., x_p) = 1 - \sum_{i=1}^{p} x_i^2 ,
\]
where \( x_i \) represents a ratio between the number of examples (covered by a cluster) from class \( i \) and the total number of examples covered by the local cluster.

Correspondingly, we define two clustering performance indexes by using weighted impurity measures (WIM) as follows:
\[
P_e(k) = \sum_{i=1}^{k} w_i \varphi_e(x_1, x_2, ..., x_p) ,
\]
\[
P_g(k) = \sum_{i=1}^{k} w_i \varphi_g(x_1, x_2, ..., x_p) ,
\]
where \( k \) represents the cluster number (provided by designer), \( w_i = n_i / N \), \( n_i \) is the number of examples covered by the \( i \)-th cluster, and \( N \) is the total number of examples in the database; \( \varphi^e_i \) and \( \varphi^g_i \) represent the entropy function and Gini index for the \( i \)-th cluster, respectively.

The use of various similarity metrics in clustering algorithms will result in different outcomes. It is desired that the value of the WIM can be as smaller as possible and insensitive to the LPM model parameter perturbations simultaneously. The latter is related to the system robustness or model reliability issue. Robustness can be defined as the ability of a system to maintain its performance when subjected to noise in external inputs, changes to internal structure and/or shifts in parameters [9]. Higher model reliability implies a relaxed requirement to solution constraint. Conversely if the model reliability is weak, then the variation scope of the parameters becomes limited. This makes the process of achieving a feasible solution more complicated or difficult. For the LPM, the model parameters are the weights of the neural networks, the solution refers to a set of specified weights obtained through learning, and the constraint may be learning rate or terminal conditions. To investigate the model's reliability, we generate a random matrix, namely, \( M_{\text{noise}} \), whose size equals the weight matrix and its elements are uniformly distributed in (-1,1). Then, perturbed weight matrix can be obtained by
\[
W_{\text{noise}} = (I + \delta M_{\text{noise}})^{\ast} W ,
\]
where \( I \) is an identity matrix with compatible size, \( W \) is the obtained weight matrix, and the \( \delta \) varies from different levels.

Finally, we calculate the WIM from the perturbed LPM models with different noise levels.

5. Experimental results

The proposed techniques were evaluated by an image database with 335 items that belong to the following 11 classes: Air Show, Bald Eagle, Cheetah, Desert, Elephant, Field, Horse, Mountain, Night Scene, Sunset and Tiger. For each class, there are around 30 images. There are 51 visual features extracted from the images. Of these features, there are 9 color moments for the three-color channels, 42 local statistics, i.e., the means and standard deviations, of the coefficients of discrete cosine transform (DCT) for three-color channels defined over 7 sub-blocks as depicted in Figure 1.

Figure 1. Segmentation of the DCT Transform Coefficients

Multilayer perceptrons (MLPs) with architecture 102-50-1 were employed to model the LPM. The MLPs were trained by using a scaled conjugate gradient (SCG) algorithm [3], and the learning programs were terminated after 1200 epochs. A 10-fold cross validation scheme (each fold contains 70% randomly selected examples as the training data set and the remaining 30% examples as the test set) was adopted in our experiments for performance evaluation.

To evaluate the goodness of the modeling, we randomly selected a pair of points from the training data sets and the test data sets respectively. Then, they were fed into the neural network models to calculate the degree of the semantic relevance similarity for this pair. In this experiment, the \( \mu \) was taken as 0.3 and resulted in a LPM model for the test data sets. Table 1 gives a statistics for modeling performance. The figures given in percentage represent the ratios between the number of testing pairs that met the corresponding criteria (denoted by C1-C4) in Definition 3 and the total number of testing pairs (i.e., the number listed in the first column of Table 1). The results demonstrate that the MLPs performed at reasonable
accuracy for this data set.

Table 1. LPM model verification (in %)

<table>
<thead>
<tr>
<th>Pairs No.</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>78.80</td>
<td>95.50</td>
<td>90.10</td>
<td>93.30</td>
</tr>
<tr>
<td>4000</td>
<td>79.03</td>
<td>95.78</td>
<td>91.63</td>
<td>92.35</td>
</tr>
<tr>
<td>8000</td>
<td>79.56</td>
<td>95.55</td>
<td>91.54</td>
<td>92.91</td>
</tr>
<tr>
<td>12000</td>
<td>79.68</td>
<td>95.56</td>
<td>91.70</td>
<td>93.01</td>
</tr>
<tr>
<td>16000</td>
<td>79.59</td>
<td>95.51</td>
<td>91.28</td>
<td>92.89</td>
</tr>
<tr>
<td>20000</td>
<td>79.71</td>
<td>95.52</td>
<td>91.43</td>
<td>93.04</td>
</tr>
</tbody>
</table>

Using the obtained LPM, we ran a standard k-Mean clustering algorithm [4] for 10 times. Table 2 and 3 show the clustering performance (average figures of 10-fold cross validation runs) for both training and test sets respectively, where $k$ in these tables represents the number of clusters and was given by designers. Obviously, our LPM-based k-Mean approach produced much better results for this database. This reveals that the Euclidean norm, as a commonly used similarity metric, is not an appropriate candidate to perform semantic or conceptual data clustering tasks although it has been widely employed in literature.

Table 2. Performance comparison for the training sets

<table>
<thead>
<tr>
<th>Pairs No.</th>
<th>$LPM$ (k)</th>
<th>$P_e$ (k)</th>
<th>$E$ (k)</th>
<th>$P_e$ (k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.2309</td>
<td>0.5989</td>
<td>0.7321</td>
<td>2.1885</td>
</tr>
<tr>
<td>4000</td>
<td>0.1906</td>
<td>0.5061</td>
<td>0.7089</td>
<td>2.1044</td>
</tr>
<tr>
<td>8000</td>
<td>0.1658</td>
<td>0.4499</td>
<td>0.7111</td>
<td>2.1285</td>
</tr>
<tr>
<td>12000</td>
<td>0.1624</td>
<td>0.4408</td>
<td>0.6902</td>
<td>2.0201</td>
</tr>
<tr>
<td>16000</td>
<td>0.1275</td>
<td>0.3659</td>
<td>0.6681</td>
<td>1.9427</td>
</tr>
<tr>
<td>20000</td>
<td>0.1245</td>
<td>0.3589</td>
<td>0.6856</td>
<td>1.9932</td>
</tr>
</tbody>
</table>

Table 3. Performance comparison for the test sets

<table>
<thead>
<tr>
<th>Pairs No.</th>
<th>$LPM$ (k)</th>
<th>$P_e$ (k)</th>
<th>$E$ (k)</th>
<th>$P_e$ (k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.4621</td>
<td>1.2431</td>
<td>0.6975</td>
<td>2.0341</td>
</tr>
<tr>
<td>4000</td>
<td>0.4471</td>
<td>1.2037</td>
<td>0.6768</td>
<td>1.9154</td>
</tr>
<tr>
<td>8000</td>
<td>0.4284</td>
<td>1.1455</td>
<td>0.6500</td>
<td>1.8405</td>
</tr>
<tr>
<td>12000</td>
<td>0.4304</td>
<td>1.1468</td>
<td>0.6615</td>
<td>1.8502</td>
</tr>
<tr>
<td>16000</td>
<td>0.4051</td>
<td>1.0768</td>
<td>0.6336</td>
<td>1.7231</td>
</tr>
<tr>
<td>20000</td>
<td>0.3967</td>
<td>1.0450</td>
<td>0.6386</td>
<td>1.7669</td>
</tr>
</tbody>
</table>

Table 4 reports the performance on robustness with respect to model parameter changes, where data used in this experiment are the whole data set (i.e., the training and the test sets). The given figures are the average values over different numbers of clusters (i.e., $k = 10:15$) and for 10 runs (i.e., 10 sets of noisy weights), the value of the $\delta$ in (8) took 10 different values from 0.01 to 0.1. From the results, it can be seen that the LPM-based k-Mean clustering algorithm demonstrates a nice robust property. This implies that the LPM model implemented using neural networks is reliable to perform clustering tasks for this data set.

Table 4. Performance robustness comparison

<table>
<thead>
<tr>
<th>$\delta$</th>
<th>LPM approach</th>
<th>Euclidean norm</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
<td>0.2720</td>
<td>0.8052</td>
</tr>
<tr>
<td>0.02</td>
<td>0.2750</td>
<td>0.8183</td>
</tr>
<tr>
<td>0.03</td>
<td>0.2860</td>
<td>0.8659</td>
</tr>
<tr>
<td>0.04</td>
<td>0.2881</td>
<td>0.8671</td>
</tr>
<tr>
<td>0.05</td>
<td>0.2856</td>
<td>0.8616</td>
</tr>
<tr>
<td>0.06</td>
<td>0.2938</td>
<td>0.8993</td>
</tr>
<tr>
<td>0.07</td>
<td>0.3050</td>
<td>0.9322</td>
</tr>
<tr>
<td>0.08</td>
<td>0.2961</td>
<td>0.9207</td>
</tr>
<tr>
<td>0.09</td>
<td>0.3329</td>
<td>1.0240</td>
</tr>
<tr>
<td>0.10</td>
<td>0.3328</td>
<td>1.0508</td>
</tr>
</tbody>
</table>

6. Conclusions

In this paper, we contribute a framework of learning pseudo metric for measuring the similarity of semantic image objects. Two new clustering quality assessments associated with the LPM based k-Mean clustering technique are proposed. The methodology developed in this paper is useful to deal with conceptual relevance similarity for multimedia objects. Experimental results demonstrated that the LPM-based clustering method outperforms the widely used Euclidean distance based method in terms of both the impurity and robustness for our testing image database.

References


4976