Learning texture similarity with perceptual pairwise distance

Yan Gao  
Nanyang Technological University, Singapore, yg904@uowmail.edu.au

Lei Wang  
University of Wollongong, leiw@uow.edu.au

Kap Luk Chan  
Nanyang Technological University, Singapore

Wei-Yan Yau  
Institute for Infocomm Research Singapore

Publication Details
Learning texture similarity with perceptual pairwise distance

Abstract
In this paper, we demonstrate how texture classification and retrieval could benefit from learning perceptual pairwise distance of different texture classes. Textures as represented by certain image features may not be correctly compared in a way that is consistent with human perception. Learning similarity helps to alleviate this perceptual inconsistency. For textures, psychological experiments were shown to be able to construct perceptual pairwise distance matrix. We are going to show how this distance information could be utilized in learning similarity by Support Vector Machines for efficient texture classification and retrieval.

Keywords
learning, texture, similarity, perceptual, distance, pairwise

Disciplines
Engineering | Science and Technology Studies

Publication Details

This conference paper is available at Research Online: http://ro.uow.edu.au/eispapers/657
Abstract

In this paper, we demonstrate how texture classification and retrieval could benefit from learning perceptual pairwise distance of different texture classes. Textures as represented by certain image features may not be correctly compared in a way that is consistent with human perception. Learning similarity helps to alleviate this perceptual inconsistency. For textures, psychological experiments were shown to be able to construct perceptual pairwise distance matrix. We are going to show how this distance information could be utilized in learning similarity by Support Vector Machines for efficient texture classification and retrieval.

1. Introduction

Texture is one of the important visual properties that has received intense studies [8, 9]. Many texture descriptors describing textural properties both in the spatial image domain and frequency domain have been reported in literature. Similarly, in the study of human perception, psychological experiments on textures were also reported [10]. There have been some exciting findings about texture perception. It has been identified that periodicity, directionality and structural complexity [10] are the most dominant factors in discriminating different texture classes. However, it is believed that none of the computational image features we have at our disposal directly measure or correlate with the factors that affect human perception, even though there are the image features extracted from multichannel Gabor filters being thought of as a close modeling of the early human vision. It is of imminent importance and of great interests to bridge this so-called perceptual gap.

Most of recent research focused on the terminology of learning similarities. Its objective is that given human labeled textures of different classes or groups of similar classes, a classifier that is in accordance with human perception (in terms of classes) can be learned and such a trained classifier can be used to perform classification, and ideally the results should be in agreement with human perception. Essentially, this is a supervised learning process with labeled samples. In [8], self-organized maps (SOMs) were used to learn the classifier. In [5], multi-class Support Vector Machines (SVMs) were used. In their methods, texture classes were grouped into clusters of similar textures and classification into these different clusters was performed by the respective learning machine. However, human perception of similarities of texture is not simply ‘discrete’ classification. We always have a sense of the degree of similarity. For example, Class A may be more similar to class B than Class C is. Learning similarity in terms of classification may only be considered the preliminary exploration in filling up the perceptual gap.

In [6] and [7], findings in psychological studies were borrowed into the field of perceptual consistent texture space learning. Given the pairwise distance information of 60 texture classes obtained from psychological experiments, Multidimensional Scaling (MDS) was used to find a low dimensional embedding and Support Vector Regression was used to construct a mapping from the original feature space to the low dimensional perceptual space. If such a mapping is found, texture classification and retrieval could be performed in the perceptual space and the results obtained should be more perceptually consistent with human perception. However, this method involves two stages of mapping, firstly from pairwise distance to low dimensional perceptual space and secondly from feature space to perceptual space. Especially, in the second stage of mapping, one Support Vector regression is needed for each dimension in the perceptual space.

In this paper, we also make use of the results from psychological studies. In particular, we use the results obtained in [10] where the pairwise perceptual distance of 30 texture classes from Brodatz album [1] was obtained. Moreover, we propose a way to seamlessly integrate the perceptual distance information into the Support Vector Machines which is a popular state-of-the-art learning machine that possesses many elegant properties. In the process, in contrast to many existing methods, model selection for the
kernel used (i.e. optimal parameter selection for a given kernel function used) in SVMs is also simultaneously solved. In addition, we propose a method to assess the perceptual quality of image retrieval.

2. Human Perception of Texture

Various psychological experiments have been done to study human perception of textures. In [10], a subset of 30 images from Brodatz album were used and this subset was chosen to capture the variations in the texture across the album and was a representative subset of the original 112 pictures. Twenty subjects participated in the study. A bottom-up sorting procedure was adopted because it enabled the sorter to create hierarchically structured groups of similar items. Each subject was asked to group the 30 pictures into as many classes as desired. The subjects then repeatedly regrouped the pictures into smaller numbers of classes. For each of the initial groupings of the 30 textures by the 20 subjects, a lower triangular similarity matrix was constructed in which the matrix elements were filled with ones for two textures within the same group and they were filled with zeros for textures in different groups. These individual subject matrices were summed across all 20 subjects to give a pooled similarity matrix. A graphical illustration of the results of the study is given in Appendix A.

Hence, a distance matrix of $30 \times 30$ measuring the pairwise perceptual distance of the 30 texture classes could be constructed. This matrix represents human perception of the 30 texture classes. We shall denote this matrix $D_p$ and we are going to show later how this matrix can be learned to improve texture classification and retrieval. In addition, this matrix can also be converted to a ranking matrix of $30 \times 30$ in which column $i$ is an ordered list of classes similar to class $i$. Hence, we can say that we have obtained the ground truth ranking of the image classes, with which we are going to show how the perceptual quality of image retrieval could be assessed.

3. Support Vector Machines for Learning Similarity

Let $\mathcal{D}$ be a training data set and $\mathcal{D} = \{(x, y)\} \in (\mathbb{R}^n \times \mathcal{Y})^{\lvert \mathcal{D} \rvert}$, where $\mathbb{R}^n$ denotes an $n$-dimensional input space, $\mathcal{Y} = \{\pm 1\}$ denotes the label set of $x$, and $\lvert \mathcal{D} \rvert$ is the size of $\mathcal{D}$. Through the kernel trick, SVM finds an optimal separating hyperplane in the kernel space which classifies the two classes by the minimal expected test error. A kernel, $k$, is defined to be $k(x, y) = \langle \phi(x), \phi(y) \rangle$, where $\phi(\cdot)$ is the associated mapping from a feature space, $\mathbb{R}^n$, to a kernel space, $\mathcal{F}$. This mapping is often nonlinear, and the dimensionality of $\mathcal{F}$ can be of high or even infinite dimensions. The nonlinearily separable patterns in $\mathbb{R}^n$ can become linearly separable in $\mathcal{F}$ with higher probability. Let $\langle w^*, \phi(x) \rangle + b^* = 0$ denote this hyperplane, where $w^*$ and $b^*$ are normal vector and bias, respectively. $w^*$ and $b^*$ can be found by minimizing

$$\Phi(w) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{\lvert \mathcal{D} \rvert} \xi_i$$

subject to: $y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i$, $i = 1, \ldots, \lvert \mathcal{D} \rvert$

(1)

where $\xi_i$ ($\xi_i \geq 0$) is the i-th slack variable and $C$ is the regularization parameter controlling the trade-off between function complexity and training error. The decision function is

$$f(x) = \sum_{i=1}^{\lvert \mathcal{D} \rvert} \alpha_i y_i \langle \phi(x_i), \phi(x) \rangle + b^* = \sum_{i=1}^{\lvert \mathcal{D} \rvert} \alpha_i y_i k(x_i, x) + b^*$$

(2)

where $\alpha_i$ is a non-negative coefficient.

In the test phase, a test sample is labelled as $\text{sgn} [f(x)]$, where $\text{sgn}[\cdot]$ denotes the sign function. The standard SVMs are designed for binary classification. For the multi-class classification problem it is commonly solved by a decomposition into several binary problems. Popularity used approaches include one-against-one and one-against-all decompositions. The final classification is usually determined by voting. In some scenarios, probability estimates are desired. Probability estimates could be obtained by combining all pairwise comparisons (see [2] for an example).

The commonly used kernel functions in SVMs are Gaussian Radial Basis Function (RBF) kernel, polynomial kernel, and sigmoid kernel. Gaussian RBF kernel is generally preferred for its superior capability in many real-world applications. SVMs possess many desired properties as a learning machine in terms of efficiency of training, efficiency of testing, overfitting and parameter tuning [3]. In spite of all these, the kernel used in SVMs is crucial and if the kernel is not selected and tuned to fit the learning task, very poor performance of the SVMs could be resulted. Although many fast model selection approaches have been proposed, cross validation is still considered as the most reliable approach though it is very time-consuming. In real-time applications such as image retrieval, cross validation becomes unacceptable as an on-line learning step.

4. The Proposed Perceptual Similarity Learning Approach

4.1. The basic idea

In order to introduce our proposed method of learning similarity, we need to first discuss more about the kernel used in SVMs. As mentioned in the preceding section, kernel is the soul of SVMs. The kernel function is the dot
product of two data points \(\langle \phi(x_i), \phi(x) \rangle\) in the implicitly induced kernel space. The dot product can also be considered as a measure of proximity of the two data points in the kernel space. For the Gaussian RBF kernel, a simple transformation will make this relationship clearer.

The Gaussian RBF kernel is defined as \(k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)\), where the Gaussian width \(\sigma\) is the kernel parameter. This kernel ranges from 0 to 1 and \(k(x_i, x_j) = 1\) only when \(x_i = x_j\). Under this kernel, the squared Euclidean distance between two data points in the kernel space is

\[
||\phi(x_i) - \phi(x_j)||^2 = \langle \phi(x_i), \phi(x_i) \rangle + \langle \phi(x_j), \phi(x_j) \rangle - 2\langle \phi(x_i), \phi(x_j) \rangle
\]

\[
= k(x_i, x_j) + k(x_j, x_j) - 2k(x_i, x_j)
\]

\[
= 2 - 2k(x_i, x_j)
\]

(3)

Let \(K_g\) denote the kernel matrix, and \(D_g\) the corresponding matrix of pairwise squared Euclidean distance. From Eq. (3), we get

\[
D_g = 21 - 2K_g
\]

(4)

where \(I\) is a square matrix of ones of the same size as \(K_g\).

Therefore, the kernel is directly related to distance in the kernel space. Since SVMs actually perform classification in the kernel space, the “goodness” of the kernel space determines how well the classifier could be. The kernel space is fully described by the kernel matrix, \(K_g\), or equivalently, the distance matrix \(D_g\). Now, we shall see how our knowledge of perceptual pairwise distance comes in and the interesting interpretation of it as an ideal distance matrix.

Suppose we have a mapping \(\psi\) that could map the feature vector of an image \(x_i\) to a perceptual representation \(\psi(x_i)\) that is in perfect agreement with human perception. Hence, it is expected that in this perceptual representation, feature vectors of different texture classes should be able to resemble the pairwise distance matrix \(D_p\) obtained from psychological study. Here we notice that \(\psi\) can be interpreted as the implicit mapping \(\phi\) in the SVMs, and \(D_p\) can be interpreted as the ideal distance matrix \(D_g\). In the induced kernel space, it means that images from the same class should be clustered around the same location, while images from different classes are separated with a between-class distance equal to the perceptual distance.

At this point, we are ready to explore how to incorporate the learning of human perception seamlessly into the SVMs for learning similarities for the tasks of texture classification and retrieval. The kernel, or the induced kernel space, provides a natural mechanism that embeds the distance information into the learning process of SVMs. In fact, we could learn a specially designed kernel whereby the given distance information can be optimally embedded into it. Recent work in kernel learning has achieved some promising results (e.g., [4]) but in their work, they only learn from class information but not perceptual distance. In this paper, instead of learning a specially designed kernel, we use a pre-defined kernel, such as the commonly used Gaussian RBF kernel, to learn from the perceptual distance matrix by optimizing the kernel parameters. Hence, the problem of learning similarity from distance is transformed to learning kernels, and further simplified to a kernel tuning problem.

### 4.2. The proposed learning approach

We need a criterion that measure how close the distance matrix of the kernel used is to the ideal distance matrix \(D_p\). We can write down a general form of the criterion denoted as \(J\)

\[
J = J(D_p, D_g; \Omega_j)
\]

where \(\Omega_j\) is the parameter set of the criterion \(J\).

This criterion measures how good the employed kernel is in terms of resembling the ideal distance matrix in its kernel space. The optimal kernel parameter \(\sigma\) of a Gaussian RBF kernel is hence given by:

\[
\sigma^* = \arg \min_{\sigma \in \Theta} J
\]

(6)

where \(\Theta\) is the parameter space of \(\sigma\).

As an implementation of the general criterion, we shall use a simple sum-of-squared-error as the criterion \(J\). \(J\) becomes:

\[
J = \sum_{i,j} [w_{i,j}(\langle D_p \rangle_{i,j} - \langle D_g \rangle_{i,j})^2]
\]

(7)

where \(w_{i,j}(\geq 0)\) is a weight that could incorporate our knowledge about the importance of each error term. If for some reason, we are more confident with some of the entries in the perceptual distance matrix, we can readily assign larger values to the corresponding \(w_{i,j}\) so that the optimization is biased towards the most important ones. In addition, the weight could also be used to normalize the errors against the number of samples of different classes when the class sizes are not balanced, which is usually the case in image retrieval. Hence \(w\) can be defined as:

\[
w_{i,j} = \frac{1}{n_{y_i} \times n_{y_j}}
\]

(8)

where \(n_{y_i}\) and \(n_{y_j}\) are the number of samples in the texture class \(y_i\) and \(y_j\), respectively. Following Eq. (6), we have the optimal \(\sigma^*\):

\[
\sigma^* = \arg \min_{\sigma \in \Theta} \sum_{i,j} [w_{i,j}(\langle D_p \rangle_{i,j} - \langle D_g \rangle_{i,j})^2]
\]

(9)

and the gradient of this criterion is given by

\[
\frac{dJ}{d\sigma} = \sum_{i,j} [-2w_{i,j}(\langle D_p \rangle_{i,j} - \langle D_g \rangle_{i,j}) \frac{d\langle D_g \rangle_{i,j}}{d\sigma}]\]

(10)
It is straightforward to verify that $J$ has continuous first and second derivatives with respect to $\sigma$. The minimization can be solved by applying a nonlinear optimization technique. The Broyden-Fletcher-Goldfarb-Shanno (BFGS) Quasi-Newton method is often favored for less number of iterations for convergence. Although optimizing the criterion is not a convex optimization problem in general, our experiments show that the criterion function is always characterized by an obvious global minimum. The computational overhead in each iteration is largely due to evaluating $D_j$, which involves calculating $D_j $, and the complexity is $O(|D|^2)$ for a given visual feature vector. Hence, it is expected that the optimization process will not take much time, and it will not significantly slow down the response required in real-time applications.

5. Experimental Results

5.1. Image database and visual features

As mentioned in Section 2, the Brodatz album was used. The same 30 texture classes as in [10] were selected (see Appendix A). Each of the image was scanned in $512 \times 512$ pixels resolution and each of the large texture image was subdivided into 16 non-overlapping images of size $128 \times 128$ as in [9]. Therefore we had altogether 16 samples from each class.

The texture feature we used was Gabor texture feature [9]. In [7], it has been shown that Gabor feature is the most perceptual consistent feature among the several types of texture feature tested. Following [9], Gabor feature at 4 scales and 6 orientations were extracted and mean and variance were used to represent each band. Therefore we obtained a 48 dimensional feature vector representation of the texture image.

5.2. Texture classification

For classification, we used 8 images from each class as training data and the other 8 images as test images. Random splitting of the 16 images from each class into training set and test set was performed. Totally 20 such combinations of training set and test set were generated and experimental results were averaged over these 20 sets for robust comparison. In order to assess the performance of the proposed approach in the handling of small training set which usually occurs in image classification and retrieval, the number of training samples from each class was gradually reduced from 8 to 2 while the test set remained unchanged.

The classification was basically formulated as a multi-class paradigm. The BFGS Quasi-Newton method was used to find the $\sigma$ that minimized the proposed criterion. The initial value of $\sigma$ was set to $\sigma_0 = 2.0$, and the stop criterion was set to $|J(\sigma_{i+1}) - J(\sigma_i)| \leq 10^{-6}J(\sigma_i)$. It was observed in the experiments that the criterion did not exhibit any local minima problem. A plot of the criterion in one of the retrieval sessions is given in Figure 1.

![Figure 1. The criterion curve](image)

To assess the performance of the proposed approach, we also used the cross validation approach to tune the kernel and obtained the corresponding classification performance. 5-fold cross validation was performed on the training data of the first 5 sets out of the total 20. When the best kernel parameter set was obtained, it was used for training and classification on all the 20 sets and average results were recorded.

The classification accuracy of the proposed approach and the cross validation approach are shown in Figure 2. It can be observed that our method achieves a comparable performance and an even better performance for very small training sizes. When the training set is small, the capability of cross validation falls as it relies on representative training samples to generalize well on the test data. In contrast, our criterion is less demanding on the size of the training set.

Table 1 shows the optimization time taken by the two methods. It can be seen that the proposed approach takes very little time while the cross validation approach incurs a very long computation overhead. The optimization time is critical in real-time applications such as image retrieval, where cross validation approach would become unrealistic.

5.3. Texture retrieval

When SVMs are used for image retrieval, the decision value $f(x)$ (refer to Section 3) is usually used as a measure for ranking the retrieved images. For multi-class retrieval, we followed a procedure similar to that in [5]. The LIB-SVM tool we used could output a probability estimate of
Table 1. Average optimization time ($t_o$) for training set of different number of samples from each class ($N_{train}$). PA: the Proposed Approach. CV: the Cross Validation approach

<table>
<thead>
<tr>
<th>$N_{train}$</th>
<th>$t_o$ (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.2424</td>
</tr>
<tr>
<td>7</td>
<td>0.1953</td>
</tr>
<tr>
<td>6</td>
<td>0.1551</td>
</tr>
<tr>
<td>5</td>
<td>0.1222</td>
</tr>
<tr>
<td>4</td>
<td>0.0907</td>
</tr>
<tr>
<td>3</td>
<td>0.0644</td>
</tr>
<tr>
<td>2</td>
<td>0.0477</td>
</tr>
</tbody>
</table>

Figure 2. Texture classification accuracy versus number of training samples

Figure 3. The ground truth ranking of retrieved images when the query is from class $c_1$. $c_2$ is the 2nd most similar class to $c_1$ and similarly for $c_3$ to $c_{30}$. It means that among the ordered 239 retrieved images, the top 7 matches should be from $c_1$, and the next 8 top matches should be from class $c_2$, and so on.

Hence, we could measure the quality of perceptual ranking of the retrieved images as the number of images correctly retrieved according to the ground truth at the check points at the 7th, 15th, 23rd · · · retrieved image. For example, we might have 6 images from class $c_1$ in the top 7 matches, and 10 images from classes $c_1$ and $c_2$ in the top 15 matches. In this way, we could obtain a Perceptual Recall curve with the horizontal axis being the check points according to the ground truth and the vertical axis the number of correct matches at those check points.

Figure 4 shows the average perceptual recall using the proposed method as well as using cross validation. The ground truth is also drawn which is always at 45 degrees. Again our method achieves a comparable performance with that of cross validation. The perceptual consistency of retrieval could be measured by how close the perceptual recall curve is to the ground truth curve. Hence, the perceptual re-
call curve also serves as a measure for comparison for further studies on perceptual ranking of images.

![Figure 4. Perceptual Recall curve](image)

6. Summary and Conclusions

In this paper, a method of learning similarity to learn from the perceptual distance of textures obtained from psychological experiments is proposed. The learning process is seamlessly integrated into training of Support Vector Machines. At the same time, the well-known problem of model selection in SVMs is also solved. The proposed approach achieves a comparable performance with that of cross validation, while the optimization time is significantly reduced. It is worth noting that this is only a preliminary work and its capability is limited because of the pre-defined kernel function. Its performance is bounded by the best possible results obtained by a Gaussian RBF kernel. For the future work, combination of different kernels and specially designed kernels will be explored. For texture retrieval, SVMs may not be an optimal choice since retrieval is a ranking process while SVMs are designed for classification. Other approaches such as Support Vector Regression may produce better perceptual ranking results.

References


A. Appendix

In [10], after constructing the similarity matrix from psychological experiments, Multi-dimensional Scaling was then applied to obtain a low dimensional embedding of the 30 texture classes. It was found that embedding in a 3-dimensional space achieved a stress of only 0.045. The 30 texture classes selected in the study and their locations in 3-dimensional space is shown in Figure 5. The 3 dimensions are found to be corresponding to periodicity, directionality and structural complexity.

![Figure 5. The 30 texture classes plotted with data from [10] in 3D](image)