A PSO-SVM Approach for Image Retrieval and Clustering

Lianze Ma¹, Lin Lin¹,² and Mitsuo Gen²

¹Dalian University of Technology, Dalian, China
²Fuzzy Logic Systems Institute, Kitakyushu, Japan
mr_maggie@live.cn, lin@dlut.edu.cn, mitsuogen@gmail.com

Abstract

In order to improve the retrieval accuracy of content-based image retrieval systems, several approaches have been designed for sophisticated low-level feature extraction and the 'semantic gap' reduction between the visual features and the richness of human semantics. In this paper, we propose a hybrid approach combining Particle Swarm Optimization (PSO) and Support Vector Machine (SVM) for image retrieval and clustering. Relevance feedback schemes using linear/quadratic estimators have been applied in content-based image retrieval to improve retrieval performance significantly. In this paper, an improved PSO method is proposed with adaptive weight of SVM to improve retrieval performance. In addition, a composite histogram approach is employed which represents a composition of color, luminance, and edge features in the image, to extract low-level feature of an image. As a result, every image can be presented as a composite histogram. The experimental results show that the PSO-SVM approach for image retrieval and clustering problem can achieve higher recognition accuracy and higher recognition speed.

Key words: Image Retrieval, Clustering, Particle Swarm Optimization, Support Vector Machine, Composite Histogram

1. Introduction

In recent years, content-based image retrieval has become a major research area due to the increasing rate of image generated on the internet and in many other fields. One of the most exciting and difficult open problems of visual information retrieval is enabling a machine to recognize objects and object categories in images. A popular way to reducing the “semantic gap” is using machine learning tools to associate low-level features, such as color, texture, shape, with query concepts. There are three fundamental components in these systems: (1) low-level image feature extraction, (2) similarity measures, (3) 'semantic gap' reduction.

Image clustering is an important searching part of content-based image retrieval and image clustering plays an important role in this area. For example, in ref.[1], using binary Bayesian classifier, high-level concepts of natural scenes are captured from low-level image features. Database images are automatically classified into general types as indoor/outdoor, and the outdoor images are further classified into city/landscape. However, there are still difficulties in image clustering that accurate low-level features are hard to get and some high-level concepts are complicate to define.

In this paper, we employ a composite histogram proposed by park et al.[2] for image retrieval. The composite histogram consists of six major colors, five levels of luminance, and three edge bins. Images’ low-level features can be represented by just 14 histogram bins. Furthermore, one can increase the number of bins by defining more specific semantics between the bins in a
hierarchical manner. However, this approach can just retrieval the images whose features are mostly like the query image. When images are belong to the same class but have different features, this approach works not so well. To associate low-level features with high-level semantics, machine-learning is needed.

We proposed a hybrid evolutionary algorithm called PSO-SVM approach for image clustering. Support vector machine (SVM)[3], one kind of supervised machine learning, is often used to learn high-level concepts from low-level image features. However, SVM just supports atomic concepts rather than complex concepts; hence its application is partially limited. In this paper, we propose a hybrid approach combining PSO and SVM to improve the accuracy and ability for image clustering. Particle swarm optimization (PSO) relies on its learning strategy to guide its search direction. Traditionally, each particle utilizes its historical best experience and its neighborhood’s best experience through linear summation. In [4], both traditional PSO algorithm and an improved PSO are introduced. However, it is not referred to image retrieval. In this paper, we employ the traditional PSO algorithm to optimize SVM for image clustering and retrieval to get higher accuracy and speed. In short, in this paper, we proposed a PSO-SVM approach for image clustering to improve accuracy and efficiency. We use SVM to train the samples which are represented by composite histograms and PSO to get largrangians which SVM requires.

2. Low-level Features Extraction

Low-level features extraction is the basis of CBIR systems. Histogram is the most commonly used scheme to represent the global feature composition of an image. In this paper, we employ a composite histogram[1] for image retrieval. As shown in Table 1, the composite histogram consists of six major colors, five levels of luminance, and three edge bins. Therefore, the total number of bins in the histogram is only 14. Let us define the set of the histogram bins H as (1):

\[ H = [h[1], h[2], \ldots, h[14]] \]

Table 1. Semantics of the histogram bins

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Magenta</td>
<td>Blue</td>
<td>Cyan</td>
<td>Green</td>
<td>Yellow</td>
<td>Red</td>
<td>Black</td>
</tr>
<tr>
<td>Dark Gray</td>
<td>Gray</td>
<td>Light Gray</td>
<td>White</td>
<td>Vertical Edge</td>
<td>Horizontal Edge</td>
<td>Complex Edge</td>
</tr>
</tbody>
</table>

There are three groups for the features in H, namely colors (h[1]-h[6]), luminance features (i.e., brightness) with 5 bins (h[7]-h[11]), and 3 edge types (h[12]-h[14]). Then, we can use the YCbCr color space, which can be extracted from a compressed bit stream such as MPEG or JPEG, to calculate the pseudo-hue (\(ph\)) and pseudo-saturation (\(ps\)) components [1]. Finally, we can get the 14 histogram bins [1]. As long as we can obtain hue, saturation, and luminance components, we can always calculate values for the bins of six colors and five luminances in Table 1. For examples, if we use the YCbCr color space, which can be extracted from a compressed bit stream such as MPEG or JPEG, we can calculate the pseudo-hue (\(ph\)) and pseudo-saturation (\(ps\)) components from YCbCr components as in (2) and (3)[24].

\[ ph = \tan^{-1} \frac{Cb}{Cr} \]

\[ ps = \sqrt{Cr^2 + Cb^2} \]
To generate the composite histogram, we first divide the image space into non-overlapping image blocks with, for example, 16x16 pixels. Let us denote the average (pseudo) hue and (pseudo) saturation at \((p,q)\)th image block in the image as \(ph(p,q)\) and \(ps(p,q)\), respectively. If there is no edge at the image block and \(ps(p,q)>Ts\), then we classify the corresponding image block as a color block. If there is no edge but \(ps(p,q)<Ts\), then the block is classified as a luminance block. Finally, if an edge is detected in the block, we classify it as an edge block. For the edge block, we update the corresponding histogram bin by adding 1. So, it is a ‘hard’ updating. For those color and luminance blocks, we adopt a ‘soft’ updating method, which updates all relevant histogram bins at the same time. Suppose \(ph(p,q)\) is located between \(i\)th pure color and \((i+1)\)th pure color as in Fig. 1. Then, we update corresponding two histograms \(h[i]\) and \(h[i+1]\) as shown in (4) and (5).

\[
\begin{align*}
  h[i] &= h[i] + (1.0 - \alpha) \\
  h[i + 1] &= h[i + 1] + \alpha
\end{align*}
\]

(4) \hspace{2cm} (5)

where

\[
\alpha = \frac{ph(p,q) - ph_i}{ph_{i+1} - ph_i} = \frac{\theta}{60}
\]

\(ph_i\) denotes the angle of the (pseudo) hue of the \(i\)th pure color, \(i=1, \ldots, 6\) and \(\theta\) is the angle from the \(i\)th pure color to \(ph(p,q)\) as shown in Fig. 1. If an image block is classified as a luminance block as in Fig. 1, then we update the histograms for both colors and luminances (i.e., \(h[1], \ldots, h[11]\)) as (6), (7) and (8). For color histogram bins (i.e., \(i\in\{1, 2, \ldots, 6\}\)) :

\[
\begin{align*}
  h[i] &= h[i] + (1.0 - \alpha) \times \beta \\
  h[i + 1] &= h[i + 1] + \alpha \times \beta
\end{align*}
\]

(6) \hspace{2cm} (7)

where \(\beta=ps(p,q)/Ts\).

For luminance histogram bins, we have

- IF \((0<=L(p,q)<1x256/5)\) THEN \(h[7]=h[7]+(1-\beta)\)
- ELSEIF \((1x256/5<=L(p,q)<2x256/5)\) THEN \(h[8]=h[8]+(1-\beta)\)
- ELSEIF \((2x256/5<=L(p,q)<3x256/5)\) THEN \(h[9]=h[9]+(1-\beta)\)
- ELSEIF \((3x256/5<=L(p,q)<4x256/5)\) THEN \(h[10]=h[10]+(1-\beta)\)

Fig. 1. A luminance color

Fig. 2. A simple linear SVM

3. PSO-SVM Algorithm Design

3.1 Support Vector Machine

With strong theoretical foundations available, Support Vector Machine (SVM)[3] has been used for object recognition, text classification, etc. and is considered a good candidate for learning in image retrieval system. SVM is originally designed for binary classification. Assume that we
have a set of training data \{x_i, y_i\} as vectors in space \(X \subseteq \mathbb{R}_d\) belonging to two separate classes with their labels \(y_i \in \{-1,1\}\). We want to find a hyper-plane \(wx+b=0\) to separate the data, the foundation is \(g(x) = wx+b\). Among many possible hyper-planes, the optimal separating plane (OSP) is the one which maximizes the margin (the distance between the hyper-plane and the nearest data point of each class). After \(g(x)\) is normalized, OSP is equal to the problem that to minimize \(\|w\|\). The subject foundation is:

\[
\min \phi(w) = \frac{1}{2} \|w\|^2 \\
\text{s.t.} \quad y_i(w \cdot x_i + b) - 1 \geq 0, \quad i = 1,2,\cdots,N
\]

Defining \(N\) Lagrangians \(\alpha_i, i = 1,\ldots, N\). After solving this quadratic optimization problem, we can get the best hyper-plane, which \(w = \sum_{i=1}^{N} \alpha_i \cdot y_i \cdot x_i\) is the support vector, which refer to the training samples that lie closest to the hyper-plane. As in Fig. 2 the vectors lying on one side are labeled as \(-1\), and those lying on the other side are labeled as \(+1\). The classification function is:

\[
f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i \cdot y_i \cdot x_i + b\right)
\]

For image set, the data cannot be linearly separated. So we use a kernel foundation to transform the data into a higher dimensional space. The classification foundation (4) is transformed to

\[
f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i \cdot y_i \cdot K(x_i, x) + b\right)
\]

To learn multiple concepts for image retrieval, a SVM has to be trained for each concept. In this paper, we choose the RBF as the kernel foundation:

\[
K(x_i, x) = \exp\left(\frac{(x_i-x)^2}{\sigma^2}\right)
\]

To train the data samples, we put the 14 histogram bins into SVM as the vectors, while useparticle swarm optimization to confirm \{\alpha_1, \ldots, \alpha_N\}, as discussed below.

3.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithm is a global optimization method originally developed by Kennedy and Eberhart [5][6]. It is swarm intelligence [7] algorithm that emulates swarm behaviors such as birds flocking and fish schooling. It is a population-based iterative learning algorithm that shares some common characteristics with other evolutionary computation (EC) algorithms [8]. However, particle flying in the search space and adjusting its flying trajectory according to its personal best experience and its neighborhood’s best experience rather than through particles undergoing genetic operations like selection, crossover, and mutation [9]. Owing to its simple concept and high efficiency, PSO has become a widely adopted optimization technique and has been successfully applied to many real-world problems [10][11][12].

PSO searches for an optimum through each Particle swarm optimization (PSO) relies on its learning strategy to guide its search direction. Traditionally, each particle utilizes its historical best experience and its neighborhood’s best experience through linear summation. Such a learning strategy is easy to use.

When searching in a D-dimensional hyperspace, each particlei has a velocity vector \(V_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]\) and a position vector \(X_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]\) to indicate its current state, where \(i\) is a positive integer indexing the particle in the swarm and \(D\) is the dimensions of the problem under study. Moreover, particle \(i\) will keep its personal historical best position vector \(P_i = [p_{i1}, p_{i2}, \ldots, p_{iD}]\). The best position of all the particles in the \(i\)th particle’s neighborhood (the neighborhood of a particle is defined by a topology structure, e.g., the neighborhood of particle \(i\)
includes the particles $i-1$, $i$, and $i+1$ in a ring topology structure) is denoted as $P_n = \{p_{n1}, p_{n2}, \ldots, p_{nD}\}$. The vectors $V_i$ and $X_i$ are initialized randomly and are updated by (13) and (14) generation-by-generation through the guidance of $P_i$ and $P_n$.

$$v_{id} = v_{id} + c_1 r_1 d (p_{id} - x_{id}) + c_2 r_2 d (p_{nd} - x_{id}) \quad (13)$$

$$x_{id} = x_{id} + v_{id} \quad (14)$$

Coefficients $c_1$ and $c_2$ are acceleration parameters which are commonly set to 2.0 or are adaptively controlled according to the evolutionary states. The $r_1$ and $r_2$ are two randomly generated values within range $[0, 1]$ for the $d$th dimension. In order to control the flying velocity within a reasonable range, a positive value $V_{\text{MAX}d}$ is used to clamp the updated velocity. If $|v_{id}|$ exceeds $V_{\text{MAX}d}$, then it is set to $\text{sign}(v_{id}) V_{\text{MAX}d}$. However, the updated position $x_{id}$ needs not to be clamped if only the particles within the search space will be evaluated. In this way, all the particles will be drawn back to the range by $P_i$ and $P_n$ which are both within the search space.

To control or adjust the flying velocity, however, an inertia weight or a constriction factor is introduced by Shi and Eberhart, and Clerc and Kennedy, respectively. Using inertia weight $\omega$, (13) is modified to be (15), whilst using the constriction factor $\chi$, (13) is modified to be (16)

$$v_{id} = \omega v_{id} + c_1 r_1 d (p_{id} - x_{id}) + c_2 r_2 d (p_{nd} - x_{id}) \quad (15)$$

$$v_{id} = \chi [v_{id} + c_1 r_1 d (p_{id} - x_{id}) + c_2 r_2 d (p_{nd} - x_{id})] \quad (16a)$$

where

$$\chi = \frac{2}{\sqrt{|2 - 4 \varphi|^2}} \quad (16b)$$

$$\varphi = c_1 + c_2 \quad (16c)$$

In (15), $\omega$ usually decreases linearly from 0.9 to 0.4 during the run time whereas $\chi$ in (10) is preferably set to be 0.729 together with $c_1 = c_2 = 2.05$. Moreover, the parameter $V_{\text{MAX}d}$ can be omitted in a PSO with constriction factor. In this paper, $x_{id}$ represents for $\alpha_i$ in (11), finally we can get fourteen $\alpha_i$ (largrangians) for a SVM.

4. Numerical Experiments
To experiment the accuracy and speed of PSO-SVM approach for image clustering, we create 4 classes of JPG images of Arabic numerals (2, 4, 6, 8), each of which contains 500 samples with different fonts. Then each sample’s low-level features are represented as a composite histogram [2]. Every time we train two kinds of classes with traditional SVM approach [13] and PSO-SVM approach separately. Table 2 shows the results.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Accuracy %</th>
<th>Time sec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSO-SVM</td>
<td>SVM</td>
</tr>
<tr>
<td>“2”</td>
<td>89.05</td>
<td>87.70</td>
</tr>
<tr>
<td>“4”</td>
<td>88.21</td>
<td>83.75</td>
</tr>
<tr>
<td>“6”</td>
<td>90.34</td>
<td>85.86</td>
</tr>
<tr>
<td>“8”</td>
<td>90.10</td>
<td>84.50</td>
</tr>
</tbody>
</table>

The results of this experiment demonstrate that improved SVM approach with PSO can achieve higher recognition accuracy and higher recognition speed than traditional SVM approach in image retrieval and clustering.
5. Conclusions

In this paper, we propose an improved SVM approach in image clustering, which employ a composite histogram as inputs and use PSO approach to get the lagrangians. The experimental results show that after several training times PSO-SVM works better both in accuracy and searching speed than traditional SVM method. However, this method can be improved further. In the future, we hope to do more experiments to use more accurate low-level features to train SVM, and use better PSO method to work for it.

Acknowledgments

This work is partly supported by the Research 897 Start-up Capital by Dalian University of Technology (No. 1600-852012) 898 and also partly supported by the Dongseo Frontier Project Research 899 Fund of Dongseo University in 2010.

References