Large-Scale Remote Sensing Image Processing and Analysis

Xiao Bai
Beihang University
01 Background
02 Remote Sensing Image Processing
03 Remote Sensing Image Classification
04 Large-Scale Retrieval
Research Group Introduction

Bai Xiao
Professor of School of Computer Science and Engineering, Beihang University, China
Research Interests: Remote sensing/Hyperspectral data analysis, Vision computation, Image processing, Large-scale information retrieval.

School of Computer Science and Engineering of Beihang University is ranked by the Ministry of Science and Education as one of the top three computer science schools in China. Within the school there reside the State key Lab for Software Development Environment and the State key laboratory of Virtual Reality Technology and Systems. The school is proud of its academic staff including 3 members of the Chinese Academy of Sciences, 52 professors, 52 associate professors, 2 Chang Jiang Scholar, 3 professor of Recruitment Program of Global Experts and 4 part-time doctoral supervisors.

Laboratory Members:
The research team includes 5 teachers, 7 PhD students, 15 Master students. The team has published over 100 papers in the area of RS image processing and analysis, large-scale visual retrieval and visual computing and understanding in the past 5 years.
With the rapid development of remote sensing observation technologies, we have entered an era of remote sensing big data.

- **Large-Scale Data**: The amount of remote sensing images has increased dramatically, due to the recent advances in satellite technology.

- **Data Quality**: Noisy images, Low-resolution images, Mixed pixel images…

- **Data Tags**: Most of the remote sensing images are untagged. Manual generation of tags is often time consuming.

- **Efficient Applications**: Efficient algorithms for large-scale remote sensing images are highly demanded for practical applications.
Content

01  Background
02  Remote Sensing Image Processing
03  Remote Sensing Image Classification
04  Large-Scale Retrieval
Image Enhancement—Denoising

Nonnegative Tucker Decomposition for Hyperspectral Image Denoising

Conventional hyperspectral imaging process suffers from issues such as limited illumination and short sensing time, which introduce noises into the image acquisition step.
Nonnegative Tucker Decomposition for Hyperspectral Image Denoising

<table>
<thead>
<tr>
<th>Methods</th>
<th>σ = 20</th>
<th>SSIM</th>
<th>PSNR</th>
<th>σ = 30</th>
<th>SSIM</th>
<th>PSNR</th>
<th>σ = 40</th>
<th>SSIM</th>
<th>PSNR</th>
<th>σ = 50</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noisy</td>
<td>22.134</td>
<td>0.394</td>
<td>0.730</td>
<td>18.617</td>
<td>0.117</td>
<td>0.407</td>
<td>16.113</td>
<td>0.081</td>
<td>0.352</td>
<td>14.175</td>
<td>0.054</td>
</tr>
<tr>
<td>LRTA [26]</td>
<td>38.640</td>
<td>0.902</td>
<td>0.936</td>
<td>36.543</td>
<td>0.854</td>
<td>0.905</td>
<td>34.832</td>
<td>0.826</td>
<td>0.896</td>
<td>34.328</td>
<td>0.826</td>
</tr>
<tr>
<td>PARAFAC [25]</td>
<td>38.877</td>
<td>0.904</td>
<td>0.933</td>
<td>35.225</td>
<td>0.830</td>
<td>0.891</td>
<td>29.536</td>
<td>0.582</td>
<td>0.777</td>
<td>27.479</td>
<td>0.463</td>
</tr>
<tr>
<td>B=K-SVD [13]</td>
<td>30.384</td>
<td>0.678</td>
<td>0.863</td>
<td>28.947</td>
<td>0.582</td>
<td>0.775</td>
<td>28.129</td>
<td>0.529</td>
<td>0.760</td>
<td>27.349</td>
<td>0.486</td>
</tr>
<tr>
<td>K-SVD [16]</td>
<td>35.329</td>
<td>0.836</td>
<td>0.893</td>
<td>33.533</td>
<td>0.769</td>
<td>0.860</td>
<td>33.459</td>
<td>0.781</td>
<td>0.867</td>
<td>22.116</td>
<td>0.203</td>
</tr>
<tr>
<td>BwBM3D [14]</td>
<td>36.736</td>
<td>0.867</td>
<td>0.884</td>
<td>35.174</td>
<td>0.833</td>
<td>0.863</td>
<td>33.358</td>
<td>0.794</td>
<td>0.854</td>
<td>33.002</td>
<td>0.774</td>
</tr>
<tr>
<td>BwBD [17]</td>
<td>38.642</td>
<td>0.906</td>
<td>0.917</td>
<td>36.826</td>
<td>0.874</td>
<td>0.898</td>
<td>35.105</td>
<td>0.845</td>
<td>0.892</td>
<td>34.069</td>
<td>0.819</td>
</tr>
<tr>
<td>SPA+LR [18]</td>
<td>41.143</td>
<td>0.931</td>
<td>0.951</td>
<td>38.301</td>
<td>0.903</td>
<td>0.930</td>
<td>36.528</td>
<td>0.838</td>
<td>0.899</td>
<td>35.523</td>
<td>0.869</td>
</tr>
<tr>
<td>Ours</td>
<td>41.937</td>
<td>0.946</td>
<td>0.975</td>
<td>39.026</td>
<td>0.912</td>
<td>0.939</td>
<td>37.012</td>
<td>0.862</td>
<td>0.917</td>
<td>35.972</td>
<td>0.893</td>
</tr>
</tbody>
</table>

- Fan Xu, Xiao Bai, Jun Zhou: Non-local similarity based tensor decomposition for hyperspectral image denoising. ICIP 2017: 1890-1894
Image Super-Resolution

low-resolution image $\rightarrow$ high-resolution image

- Civil: GF-2 geometric resolution 1-4 m, google map's resolution 0.39 m
- A high-quality remote sensing image is significant to applications
Utilize deep neural network, find an end-to-end mapping from training data to desired output.

For this problem, the input is low-resolution image, the output is high-resolution image.

Large volume of low-resolution and corresponding high-resolution image pairs are demanded for deep neural network training. We can downsample the high-resolution image to low-resolution image to build training dataset.

This network architecture can be also used for hyperspectral images, multispectral images, IR images and so on.

Mixed pixels are frequent in remotely sensed hyperspectral images due to insufficient spatial resolution of the imaging spectrometer, or due to intimate mixing effects.

01 Background
02 Remote Sensing Image Processing
03 Remote Sensing Image Classification
04 Large-Scale Retrieval
Due to the large volume of untagged remote sensing images, the manual generation of tags is often time consuming and becomes especially prohibitive.

Remote sensing image classification technology is significant and important.
Remote Sensing Image Classification

Object Classification via Feature Fusion Based Marginalized Kernels

1. We use the SoftMax regression to model the probabilities of each sample object belonging to the object classes.
2. We introduce an approximate method for calculating the class-to-class similarities between different classes.
3. The obtained fusion and similarity information are integrated into a marginalized kernel to build a support vector machine classifier.

Example of object representation with three types of features concatenated. Local self-similarity (LSS) and gray-level co-occurrence matrices (GLCMs) stand for shape and texture features, respectively.

Remote Sensing Image Classification

Object Classification via Feature Fusion Based Marginalized Kernels

Hyperspectral Image Classification

(a) Training original classifier

(b) Maximizing scores of samples

(c) Computing spectral weight vector for each class

(d) Training new classifier for the weighted samples

(a) Training the initial classifier for each class by using traditional SVM.
(b) Balancing the spectral bands for each sample by maximizing the modified SVM classification scores.
(c) Computing the spectral weight vector for each class.
(d) Training new classifiers by using weighted samples.

The amount of remote sensing images has increased dramatically, due to the recent advances in satellite technology.

The efficiency of many traditional methods can't meet the requirements of practical application.
## Binary Coding

### Hashing-based efficient retrieval

<table>
<thead>
<tr>
<th>Data 1</th>
<th>-1</th>
<th>+1</th>
<th>-1</th>
<th>+1</th>
<th>……</th>
<th>-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data 2</td>
<td>+1</td>
<td>-1</td>
<td>+1</td>
<td>+1</td>
<td>……</td>
<td>+1</td>
</tr>
<tr>
<td>Data N</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>-1</td>
<td>……</td>
<td>+1</td>
</tr>
</tbody>
</table>

48 bits

- 10^{-9} s to calculate Hamming distance

Store **1.8 billion** data with **10G RAM**

- Transform remote sensing images into binary codes
- The Hamming distances between binary codes preserve the pairwise similarities of the data
- Significantly reduce the storage space for large-scale data
- Calculating the Hamming distance is very fast in computer
1. Learn a set of hash functions $h_i(\cdot)$ to convert input images to binary codes.
2. Organize all the codes in a hash table.
3. Return all images within a small radius of query in database using hash table.

Hamming distance: $1 < 4$
Locality Sensitive Hashing

A simple LSH hash function:

$$h_k(x) = \text{sgn}(w_k^T x)$$

Large-Scale Retrieval

Data-Dependent Hashing Based on p-Stable Distribution

➢ In the first stage, we use the refined projection vector based on deeper analysis of the p-stable property.

➢ In the second stage, our method incorporates different ways based on two different scenarios: unsupervised iterative quantization and supervised label propagation procedure are used to learn hash functions for retrieving Euclidean neighbors and semantically similar instances respectively.

Data-Dependent Hashing Based on $p$-Stable Distribution

- **Unsupervised Hashing For Preserving Euclidean Distance**
  
  quantization error:  
  \[
  \sum_{i}^{n} \sum_{k}^{r} (\text{sign}(U_k^T v_i) - U_k^T v_i)^2
  \]
  objective function:  
  \[
  \arg \min \limits_{R} \| \text{sign}((UR)^T V) - (UR)^T V \|_F^2
  \]

- **Supervised Hashing By Incorporating Semantic Similarity**
  
  supervised semantic similarity:  
  \[
  S_{ij} = \begin{cases} 
  1, & \text{L}(i) = L(j); \\
  0, & \text{otherwise}.
  \end{cases}
  \]
  objective function:  
  \[
  \arg \min \limits_{Y} \sum_{i,j} S_{ij} \| y_i - y_j \|_2^2
  \]
  subject to:  
  \[
  y_i \in \{0, 1\}^r, \sum_{i}^{n} y_i = \frac{n}{2}1_r
  \]

Since the similarity or distance to the nearest neighbors varies considerably for different data samples, simple thresholding on the similarity function returns different numbers of neighbors.

Large-Scale Retrieval

Adaptive Hash Retrieval with Kernel based Similarity

We present a novel adaptive similarity measure which is consistent with k-nearest neighbor search, and prove that it leads to a valid kernel if the original similarity function is a kernel function.

1. We use normalized Gaussian kernel to construct a new similarity function:

\[ \kappa(x_i, x_j) = \exp\left(-\frac{(d(x_i, x_j))^2}{2\sigma^2}\right) \]

2. We propose kernel reconstructive hashing that preserves the similarity defined by an arbitrary kernel using a compact binary code.

\[
\min \sum_{x_i, x_j \in X} \left( \langle \hat{x}_i, \hat{x}_j \rangle - \kappa(x_i, x_j) \right)^2
\]

Our objective formulation is learning a set of \( r \) hash functions which generate the binary code of \( x_i \) as a vector

\[ \hat{x}_i = [h_1(x_i), h_2(x_i), \ldots, h_r(x_i)] \]

Adaptive Hash Retrieval with Kernel based Similarity

Thanks for Your Attention!