**Label Self-Adaption Hashing for Image Retrieval**

Jianglin Lu\textsuperscript{a,b,c}, Zhihui Lai\textsuperscript{a,b}, Jingxu Lin\textsuperscript{a}, Qinghong Lin\textsuperscript{a,c}, Jie Zhou\textsuperscript{a,b}

\textsuperscript{a}Shenzhen University, Shenzhen, China
\textsuperscript{b}Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China
\textsuperscript{c}Kazan Federal University, Kazan, Russia

**Motivation**

We attempt to **learn a self-adaptive cluster label matrix from the data** under the assumption that the nearest neighbor points should have a larger probability to be in the same cluster.

The hash codes are regressed to the learned cluster labels to make full use of potential discriminative information of data.

Based on anchor graph, the learning of hashing function is conducted on the $n \times m$ similarity matrix constructed by $n$ sample points and $m$ anchors. Therefore, it can perform efficient binary code learning for large-scale image retrieval.

**Contributions**

We propose a novel unsupervised hashing method called LSAH, which can **learn a self-adaptive cluster label matrix** to make full use of the potential discriminative information of data to guide the learning of binary codes.

An iterative algorithm based on Augmented Lagrange Multiplier (ALM) is elaborately designed to solve the resulting optimization problem.

Extensive experiments on three well-known data sets indicate the promising performance of the proposed LSAH.

**Overall Framework**

The proposed LSAH contains the following two parts, including the **effective hashing function learning** part and the **self-adaption label generation** part.

---

### Effective Hashing Function Learning

$$\min_{F,B} \sum_{i} \sum_{j} \|\tilde{F}(x_i) - b_j\|_2^2 S_{ij} + \lambda \mathcal{R}(\tilde{F})$$

s.t. $B \in \{-1,1\}^{m \times r}$

### Self-Adaption Label Generation

$$\min_{B,K,M} \alpha \Omega(K) + \beta \|K - BM^T\|_F^2$$

s.t. $B \in \{-1,1\}^{m \times r}$, $K^TK = I_c$, $K \geq 0$, $M^TM = I_c$

### Optimization

**Algorithm 1: Optimization algorithm of LSAH**

Input: Training set $X \subseteq \mathbb{R}^{n \times d}$, the number of classes c, bits r, iterations maxT, anchors m, nearest neighbor points k, the balance parameters $\lambda$, $\alpha$, $\beta$, $\mu > 0$, $\rho > 1$.


1. Generate m anchor points by using K-Means clustering method.
2. Construct anchor matrix $Z$ and Laplacian matrix $L$.
3. Initialize $B$, $P$ randomly.
4. Initialize $U$ and $K$ as two identity matrices, $\Theta$ with each element equal to $I$.

For $\text{iter} = 1$ to maxT do

6. Update $M$: $M = P \Theta F$
7. Update $P$: $P = (X^TX + \mu I)^{-1} X^T S B$
8. Update $K$: $K = \Theta F^T$
9. Update $U$: $U = (\Theta + \mu \mathcal{R}(S^TXF + \beta B))^T$

For $\text{i} = 1$ to d do

11. Update $\Theta$: $\Theta = \Theta - \mu (K - R)$
12. Update $\mu$: $\mu = \min(10^0, \mu)$
13. Update $\mathcal{R}(S^TXF + \beta B)$

end

end

---

### MAP RESULTS

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Code Length (bits)</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>96</th>
<th>128</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>96</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSIF</td>
<td>0.1210</td>
<td>0.1357</td>
<td>0.1476</td>
<td>0.1568</td>
<td>0.2102</td>
<td>0.2620</td>
<td>0.3155</td>
<td>0.3781</td>
<td>0.5768</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SH</td>
<td>0.1260</td>
<td>0.1252</td>
<td>0.1254</td>
<td>0.1249</td>
<td>0.2061</td>
<td>0.2686</td>
<td>0.3300</td>
<td>0.3632</td>
<td>0.3610</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AGIF</td>
<td>0.1025</td>
<td>0.1062</td>
<td>0.1046</td>
<td>0.1049</td>
<td>0.1511</td>
<td>0.4063</td>
<td>0.4926</td>
<td>0.3854</td>
<td>0.2993</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITQ</td>
<td>0.1352</td>
<td>0.1634</td>
<td>0.1664</td>
<td>0.1689</td>
<td>0.3750</td>
<td>0.3578</td>
<td>0.3595</td>
<td>0.4908</td>
<td>0.4809</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISMR</td>
<td>0.1376</td>
<td>0.1742</td>
<td>0.1775</td>
<td>0.1836</td>
<td>0.5016</td>
<td>0.4781</td>
<td>0.3982</td>
<td>0.3774</td>
<td>0.3455</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BP</td>
<td>0.1500</td>
<td>0.1665</td>
<td>0.1733</td>
<td>0.1779</td>
<td>0.3951</td>
<td>0.4616</td>
<td>0.4758</td>
<td>0.4531</td>
<td>0.4010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DREAMS</td>
<td>0.1060</td>
<td>0.1901</td>
<td>0.1926</td>
<td>0.2000</td>
<td>0.3673</td>
<td>0.4448</td>
<td>0.5417</td>
<td>0.5338</td>
<td>0.5419</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNSR</td>
<td>0.1716</td>
<td>0.1675</td>
<td>0.1664</td>
<td>0.1634</td>
<td>0.1617</td>
<td>0.1579</td>
<td>0.4527</td>
<td>0.3957</td>
<td>0.3726</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSAH</td>
<td>0.1751</td>
<td>0.1925</td>
<td>0.1987</td>
<td>0.1983</td>
<td>0.2065</td>
<td>0.5423</td>
<td>0.5423</td>
<td>0.5553</td>
<td>0.5649</td>
<td>0.5529</td>
<td></td>
</tr>
</tbody>
</table>

---

### Parameter Sensitivity

(a) Convergence

(b) Overall Framework