Deep Cross-Modal Hashing
Qing-Yuan Jiang & Wu-Jun Li
LAMDA Group, Department of Computer Science and Technology, Nanjing University, Nanjing, China.
jiaqy@lamda.nju.edu.cn, liwujun@nju.edu.cn

Introduction

• Given a query point \( q \), return the points closest to \( q \) in the database (e.g., image retrieval).
• Challenges for NNS in big data applications: curse of dimensionality; storage cost; query speed

Hashing

• Similarity preserved hashing is to map the data points from the original space into a Hamming space of binary codes with similarity preserved.
• Hashing can solve the above challenges.

Cross-Modal Hashing (CMH)

• Cross-modal retrieval: the modality of the query point is different from the modality of the points in database.
• CMH hashing for cross-modal retrieval. Low storage cost and fast query speed.

Motivation

• Almost all existing CMH methods are based on hand-crafted features.
• Hand-crafted features might not be compatible for hash-code learning.

Contribution

• A novel CMH method, called deep cross-modal hashing (DCMH), for cross-modal retrieval applications.
• DCMH is an end-to-end learning framework with deep neural networks, one for each modality, to perform feature learning from scratch.
• DCMH achieves the state-of-the-art performance on three datasets.

Model

Hashing can solve the above challenges. Similarity preserved hashing is to map the data points from the original space into a Hamming space of binary codes with similarity preserved.

Configuration of the deep neural network for test modality.

\[ \text{Loss Function} \]

\[ \text{Learning} \]

\[ \text{Conclusion} \]

• Learn \( \theta_x \) with \( \theta_x \) and \( \theta_y \) fixed.

\[ \text{BP for updating } \theta_x \text{. For each sampled point } x_i, \text{ compute the gradient:} \]

\[ \frac{\partial}{\partial \theta_x} J^x = -\sum_i \left[ (s_{ij} - y_{ij}) (1 + e^{y_{ij}}) \right] \]

\[ + \left[ (\|B - F^x_i\|_1 + \|P - B\|_1) \right] \]

\[ + \left[ (\|F1\|_1 + \|G1\|_1) \right] \]

\[ \forall i: B \in \{-1, +1\}^{nc} \text{; binary codes, where } c \text{ is the code length.} \]

\[ F \in \mathbb{R}^{xc} \text{; with } F = f(x, \theta_x) \]

\[ G \in \mathbb{R}^{xc} \text{; with } G_i = g(y_{ij}, \theta_y) \]

\[ \theta_x = \theta_x + \eta \frac{\partial}{\partial \theta_x} J^x \]

• Learn \( \theta_y \) with \( \theta_x \) and \( \theta_y \) fixed.

\[ \text{BP for updating } \theta_y \text{. For each sampled point } y_j, \text{ compute the gradient:} \]

\[ \frac{\partial}{\partial \theta_y} J^y = -\sum_j \left[ (s_{ij} - y_{ij}) (1 + e^{y_{ij}}) \right] \]

\[ + \left[ (\|B - F^y_i\|_1 + \|G - B\|_1) \right] \]

\[ + \left[ (\|F1\|_1 + \|G1\|_1) \right] \]

\[ \forall j: B \in \{-1, +1\}^{nc} \text{; binary codes, where } c \text{ is the code length.} \]

\[ F \in \mathbb{R}^{yc} \text{; with } F = f(y, \theta_y) \]

\[ G \in \mathbb{R}^{yc} \text{; with } G_i = g(y_{ij}, \theta_y) \]

\[ \theta_y = \theta_y + \eta \frac{\partial}{\partial \theta_y} J^y \]

Notation

\[ X = \{x_i\}_{i \in I}, \text{; } n \text{ points of image modality} \]

\[ Y = \{y_j\}_{j \in J}, \text{; } p \text{ points of text modality} \]

\[ S = \{s_{ij}\}_{i \in I, j \in J}, \text{; cross-modal similarities} \]

\[ x_i, \theta_x \text{; the output of deep neural network for image modality} \]

\[ y_j, \theta_y \text{; the output of deep neural network for text modality} \]

\[ s_{ij} \text{; the similarity between } x_i \text{ and } y_j \]

Experiment

Datasets

• MIRFLICKR-25K: 25,000 image-text pairs which are annotated with one of the 24 unique labels.
• IAPR TC-12: 20,000 image-text pairs which are annotated using 255 labels.
• NUS-WIDE: 260,648 image-text pairs. Each point is annotated with one or multiple labels from 81 concept labels. We select 195,834 image-text pairs that belong to the 21 most frequent concepts.
• For MIRFLICKR-25K and IAPR TC-12: 2000/10000 test/training points. For NUS-WIDE: 2100/10500 test/training points.

Hamming Ranking Task (Mean Average Precision)

<table>
<thead>
<tr>
<th>Comparison to State-of-the-Art Baselines</th>
<th>Test Method</th>
<th>Task</th>
<th>Map</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCMH</td>
<td>IAPR TC-12</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>SCM</td>
<td>IAPR TC-12</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>STM H</td>
<td>IAPR TC-12</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>SCM</td>
<td>NUS-WIDE</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>STM H</td>
<td>NUS-WIDE</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Hash Lookup Task (Precision Recall Curve)

<table>
<thead>
<tr>
<th>Comparison to State-of-the-Art Baselines</th>
<th>Test Method</th>
<th>Task</th>
<th>Map</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCMH</td>
<td>IAPR TC-12</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>SCM</td>
<td>IAPR TC-12</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>STM H</td>
<td>IAPR TC-12</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>SCM</td>
<td>NUS-WIDE</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
<tr>
<td>STM H</td>
<td>NUS-WIDE</td>
<td>T ! I</td>
<td>0.65</td>
<td>0.7</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Sensitivity to Parameters

Further Analysis

• DCMH is an end-to-end deep learning framework which can perform simultaneous feature learning and hash-code learning.
• DCMH can significantly outperform other baselines to achieve the state-of-the-art performance.

Conclusion