Deep Cross-Modal Hashing

Qing-Yuan Jiang, **Wu-Jun Li**

LAMDA Group
National Key Laboratory for Novel Software Technology
Collaborative Innovation Center of Novel Software Technology and Industrialization
Department of Computer Science and Technology, Nanjing University, China

jiangqy@lamda.nju.edu.cn, liwujun@nju.edu.cn

July, 2017

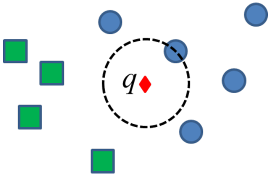
Outline

Outline

1. Introduction
2. Deep Cross-Modal Hashing (DCMH)
3. Experiment
4. Conclusion

Nearest Neighbor Search (NNS)

- Given a query point q, return the points closest to q in the database (e.g., image retrieval).



- Underlying many machine learning, data mining, information retrieval problems.

Challenge in Big Data Applications:

- Curse of dimensionality.
- Storage cost.
- Search (query) speed.

Similarity Preserving Hashing

h(Statue of Liberty) = 10001010

h (Napoléon) = 01100001

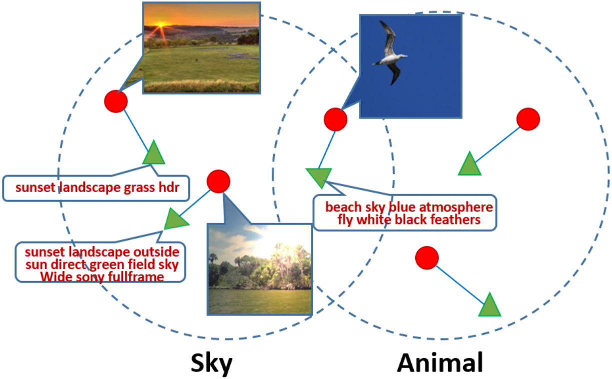
h (Napoléon) = 01100101

flipped bit

Should be very different

Should be similar

Cross-Modal Retrieval



- Given a query of either image or text, return images or texts similar to it in both feature space and semantics (label information).

Cross-Modal Hashing (CMH)

- CMH: the modality of a query point is different from the modality of the points in database.

Pros:

- Dimensionality reduction.
- Low storage cost.
- Fast query speed.

Motivation & Contribution

Motivation:

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

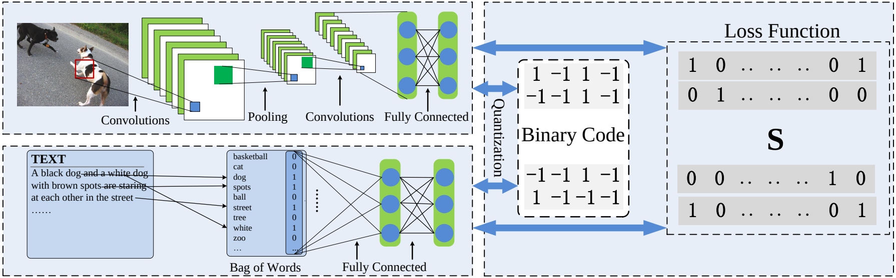
Contribution:

- An end-to-end framework, called deep cross-modal hashing (DCMH), is proposed for cross-modal retrieval application.
- DCMH achieves the state-of-the-art retrieval performance.

Outline

DCMH Model

The end-to-end deep learning framework of DCMH model.



DCMH model contains two major parts: *Feature Learning Part* and *Hash-Code Learning Part*.

Feature Learning Part

- This part contains two deep neural networks for feature learning.

Table: Configuration of the CNN for image modality.

Layer	Configuration
conv1	f. $64 \times 11 \times 11$; st. 4×4, pad 0, LRN, $\times 2$ pool
conv2	f. $265 \times 5 \times 5$; st. 1×1, pad 2, LRN, $\times 2$ pool
conv3	f. $265 \times 3 \times 3$; st. 1×1, pad 1
conv4	f. $265 \times 3 \times 3$; st. 1×1, pad 1
conv5	f. $265 \times 3 \times 3$; st. 1×1, pad 1, $\times 2$ pool
full6	4096
full7	4096
full8	Hash code length c

Table: Configuration of the deep neural network for text modality.

Layer	Configuration
full1	8192
full2	Hash code length c

Hash-Code Learning Part

$$\min_{\mathbf{B},\theta_x,\theta_y} \mathcal{J} = -\sum_{i,j=1}^{n}(S_{ij}\Theta_{ij} - \log(1+e^{\Theta_{ij}})) \\ + \gamma(\|\mathbf{B}-\mathbf{F}\|_F^2 + \|\mathbf{B}-\mathbf{G}\|_F^2) + \eta(\|\mathbf{F1}\|_F^2 + \|\mathbf{G1}\|_F^2) \\ s.t. \quad \mathbf{B} \in \{-1,+1\}^{c\times n}.$$

Notation:

- $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n / \mathbf{Y} = \{\mathbf{y}_j\}_{j=1}^n$: n points of image/text modality.
- $\mathbf{S} = \{S_{ij}\}_{n\times n}$: cross-modal similarities.
- $\mathbf{B} \in \{-1,+1\}^{c\times n}$: binary codes.
- $\mathbf{F} \in \mathbb{R}^{c\times n}$ with $\mathbf{F}_{*i} = f(\mathbf{x}_i;\theta_x)$, here $f(\mathbf{x}_i;\theta_x)$ is the output of deep neural network for image modality.
- $\mathbf{G} \in \mathbb{R}^{c\times n}$ with $\mathbf{G}_{*j} = g(\mathbf{y}_j;\theta_y)$, here $g(\mathbf{y}_j;\theta_y)$ is the output of deep neural network for text modality.
- $\Theta_{ij} = \frac{1}{2}\mathbf{F}_{*i}^T\mathbf{G}_{*j}$.

Alternating Learning Algorithm

Algorithm 1 The learning algorithm for DCMH.

Require: Image set $\mathbf{X}$, text set $\mathbf{Y}$, and cross-modal similarity matrix $\mathbf{S}$.
Ensure: Parameters θ_x and θ_y of the deep neural networks, and binary code matrix $\mathbf{B}$.
Initialization Initialize neural network parameters θ_x and θ_y, mini-batch size $N_x = N_y = 128$, and iteration number $t_x = \lceil n/N_x \rceil, t_y = \lceil n/N_y \rceil$.
repeat
  for $iter = 1, 2, \cdots, t_x$ **do**
    Randomly sample N_x points from $\mathbf{X}$ to construct a mini-batch.
    For each sampled point $\mathbf{x}_i$ in the mini-batch, calculate $\mathbf{F}_{*i} = f(\mathbf{x}_i; \theta_x)$ by forward propagation.
    Calculate the gradient by using $\frac{\partial \mathcal{J}}{\partial \mathbf{F}_{*i}} = \frac{1}{2}\sum_{j=1}^{n}(\sigma(\Theta_{ij})\mathbf{G}_{*j} - S_{ij}\mathbf{G}_{*j}) + 2\gamma(\mathbf{F}_{*i} - \mathbf{B}_{*i}) + 2\eta\mathbf{F}\mathbf{1}$.
    Update the parameter θ_x by using back propagation.
  end for
  for $iter = 1, 2, \cdots, t_y$ **do**
    Randomly sample N_y points from $\mathbf{Y}$ to construct a mini-batch.
    For each sampled point $\mathbf{y}_j$ in the mini-batch, calculate $\mathbf{G}_{*j} = g(\mathbf{y}_j; \theta_y)$ by forward propagation.
    Calculate the gradient by using $\frac{\partial \mathcal{J}}{\partial \mathbf{G}_{*j}} = \frac{1}{2}\sum_{i=1}^{n}(\sigma(\Theta_{ij})\mathbf{F}_{*i} - S_{ij}\mathbf{F}_{*i}) + 2\gamma(\mathbf{G}_{*j} - \mathbf{B}_{*j}) + 2\eta\mathbf{G}\mathbf{1}$.
    Update the parameter θ_y by using back propagation.
  end for
  Learn $\mathbf{B}$ according to $\mathbf{B} = \text{sign}(\gamma(\mathbf{F} + \mathbf{G}))$.
until a fixed number of iterations

Outline

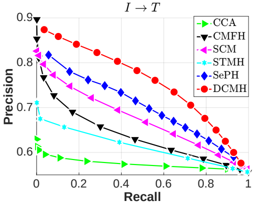
Hamming Ranking Task

Table: MAP on three datasets. The baselines are based on CNN-F features.

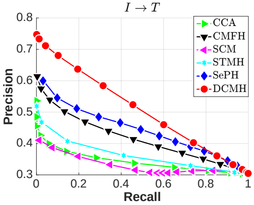
Task	Method	MIRFLICKR-25K			IAPR TC-12			NUS-WIDE		
		16	32	64	16	32	64	16	32	64
$I \rightarrow T$	DCMH	**.741**	**.747**	**.749**	**.453**	**.473**	**.484**	.590	.603	.609
	SePH	.712	.719	.723	.444	.456	.464	**.604**	**.614**	**.621**
	STMH	.613	.622	.627	.378	.400	.413	.471	.486	.494
	SCM	.685	.692	.700	.369	.367	.380	.541	.549	.555
	CMFH	.638	.642	.645	.419	.423	.425	.490	.505	.510
	CCA	.572	.569	.567	.342	.336	.330	.360	.349	.339
$T \rightarrow I$	DCMH	**.783**	**.790**	**.793**	**.519**	**.538**	**.547**	**.639**	**.651**	**.657**
	SePH	.722	.726	.732	.442	.456	.465	.598	.603	.611
	STMH	.607	.615	.622	.369	.390	.404	.447	.468	.478
	SCM	.694	.701	.706	.345	.341	.347	.534	.541	.548
	CMFH	.637	.640	.643	.417	.421	.428	.503	.519	.523
	CCA	.574	.571	.569	.349	.344	.338	.361	.349	.340

SePH [CVPR-15]; STMH [IJCAI-15]; SCM [AAAI-14]; CMFH [CVPR-14]; CCA [Biometrika-1936].

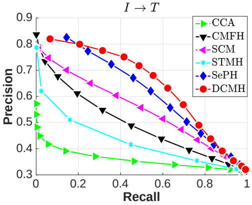
Hash Lookup Task



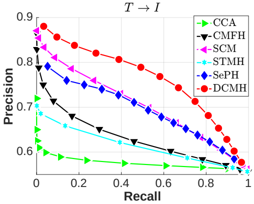
(a) MIRFLICKR-25K



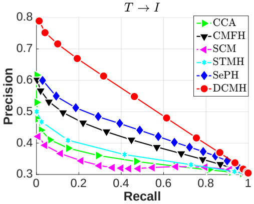
(b) IAPR TC-12



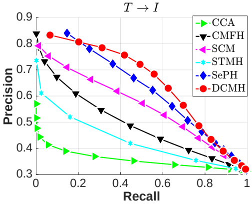
(c) NUS-WIDE



(d) MIRFLICKR-25K



(b) IAPR TC-12



(f) NUS-WIDE

Figure: Precision-recall curves. The baselines are based on CNN-F features.

Sensitivity to Parameters

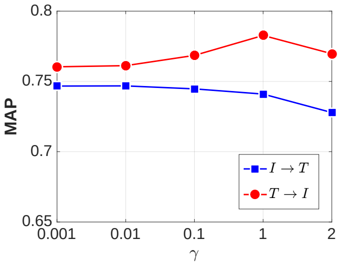


Figure: The influence of hyper-parameters.

The Effectiveness of Feature Learning

- DCMH-I denotes the variant without image feature learning.
- DCMH-T denotes the variant without text feature learning.
- DCMH-IT denotes the variant without both image and text feature learning.

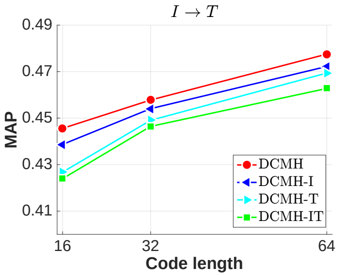


Figure: MAP on IAPR TC-12.

Outline

Conclusion

- 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.

Thanks!

Paper and code are available at `http://cs.nju.edu.cn/lwj`