2020
ACM multimedia Seattle
Seattle, United States
12-16 October 2020

Label Embedding Online Hashing for Cross-Modal Retrieval

Yongxin Wang, Xin Luo, Xin-Shun Xu*

MIMA Machine Intelligence & Media Analysis
SHANDONG UNIVERSITY

Outline

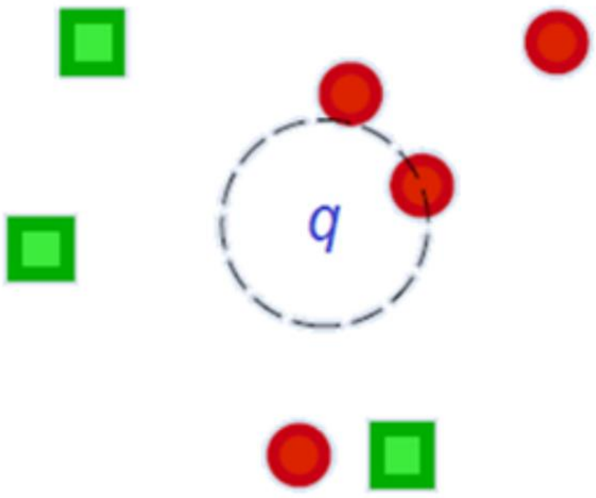
MIMA

- Introduction
- Method
  - hash codes learning
  - hash functions learning
- Experiments
- Conclusion and Future Work

Introduction

MIMA

- Nearest Neighbor Search (NNS)



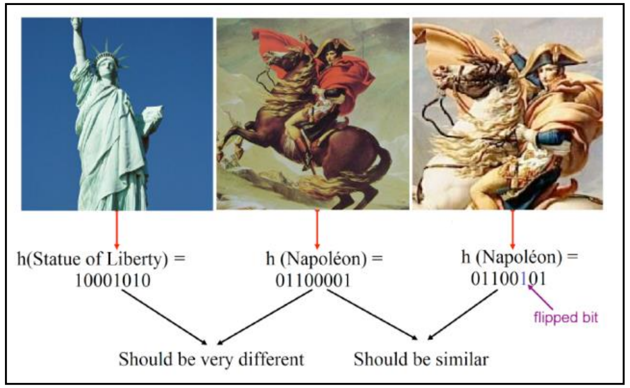
- Given a query point *q*, NNS returns the points closest (most similar) to *q* in the database.
- Underlying many machine learning, and information retrieval, problems.

- In the era of big data, traditional NNS methods faces the challenge of slow retrieval speed and expensive storage.

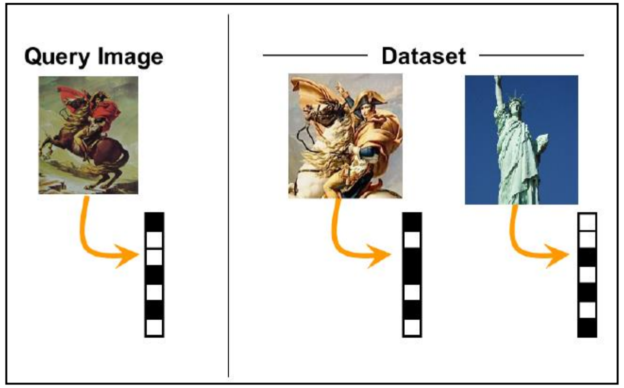
Introduction

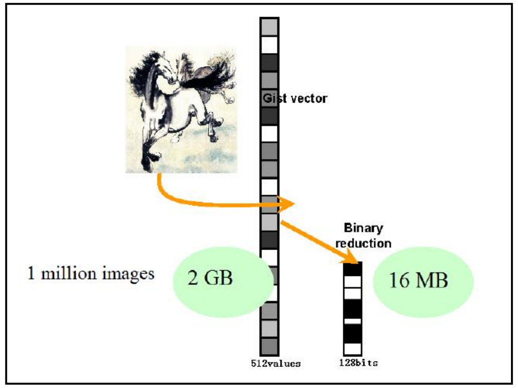
MIMA

- Hashing Methods
  - similarity relationship preserved binary codes representation



  - advantages
    - fast query speed
    - low storage cost





- Actually, data is usually collected in a stream fashion. Batch learning is inefficient.

Introduction

- Online Hashing
  - it learns hash functions in an online scenario
  - several issues to be address
    - performance is not satisfying
    - ignore the existing data
    - updating scheme is inefficient
    - discrete optimization is still an open issue

Introduction

MIMA

- **LEMON**: a novel Label EMbedding ONline hashing method
  - label embedding
  - online learning
  - efficient discrete optimization

Method

MIMA

- Notations (at *t*-th round):
  - $\mathbf{X}_m^{(t)} = [\mathbf{X}_m^{(t)}, \overrightarrow{\mathbf{X}}_m^{(t)}] \in \mathbb{R}^{d_m \times N_t}$ feature matrix of *m*-th modality
    - $\overrightarrow{\mathbf{X}}_m^{(t)} \in \mathbb{R}^{d_m \times n_t}$ new data with n_t instances
    - $\mathbf{X}_m^{(t)} \in \mathbb{R}^{d_m \times N_{t-1}}$ old data with $N_{t-1} = \sum_{k=1}^{t-1} n_k$ instances
  - $\mathbf{L}^{(t)} = [\mathbf{L}^{(t)}, \overrightarrow{\mathbf{L}}^{(t)}] \in \mathbb{R}^{c \times N_t}$ label matrix
    - $\overrightarrow{\mathbf{L}}^{(t)} \in \mathbb{R}^{c \times n_t}$ label matrix of new data
    - $\mathbf{L}^{(t)} \in \mathbb{R}^{c \times N_{t-1}}$ label matrix of old data
  - $\mathbf{B}^{(t)} = [\mathbf{B}^{(t)}, \overrightarrow{\mathbf{B}}^{(t)}] \in \mathbb{R}^{r \times N_t}$ unified binary codes
    - $\overrightarrow{\mathbf{B}}^{(t)}$ binary codes of new data
    - $\mathbf{B}^{(t)}$ binary codes of old data

Method –Hash codes learning

MIMA

- Label Embedding
  - binary codes should preserve the semantic similarity

$$\min_{\mathbf{B}^{(t)}} \|\mathbf{B}^{(t)\top}\mathbf{B}^{(t)} - r\mathbf{S}^{(t)}\|^2, \ s.t.\ \mathbf{B}^{(t)} \in \{-1,1\}^{r\times N_t},$$

  - labels could be reconstructed from binary codes

$$\min_{\{\mathbf{B},\mathbf{P}\}^{(t)}} \|\mathbf{L}^{(t)} - \mathbf{P}^{(t)}\mathbf{B}^{(t)}\|^2 + \gamma\|\mathbf{P}^{(t)}\|^2, \ s.t.\ \mathbf{B}^{(t)} \in \{-1,1\}^{r\times N_t},$$

  - jointly considering above functions

$$\min_{\{\mathbf{B},\mathbf{P}\}^{(t)}} \boxed{\alpha\|\mathbf{B}^{(t)\top}\mathbf{B}^{(t)} - r\mathbf{S}^{(t)}\|^2} + \boxed{\beta\|\mathbf{L}^{(t)} - \mathbf{P}^{(t)}\mathbf{B}^{(t)}\|^2 + \beta\gamma\|\mathbf{P}^{(t)}\|^2}, \ s.t.\ \mathbf{B}^{(t)} \in \{-1,1\}^{r\times N_t},$$

Method –Hash codes learning

MIMA

- Online Learning
  - define a block similarity matrix $\mathbf{S}^{(t)}=\begin{bmatrix}\mathbf{S}_{oo}^{(t)} & \mathbf{S}_{oc}^{(t)}\\ \mathbf{S}_{co}^{(t)} & \mathbf{S}_{cc}^{(t)}\end{bmatrix}$, $\mathbf{S}^{(t)}=2\mathbf{U}^{(t)\top}\mathbf{U}^{(t)}-\mathbf{1}\mathbf{1}^{\top}$,

    $\mathbf{U}^{(t)}$ is the 2-norm normalized label matrix

  - keep $\mathbf{B}^{(t)}$ unchanged and only update $\vec{\mathbf{B}}^{(t)}$

$$\min_{\{\vec{\mathbf{B}},\mathbf{P}\}^{(t)}} \alpha\|\vec{\mathbf{B}}^{(t)\top}\mathbf{B}^{(t)}-r\mathbf{S}_{co}^{(t)}\|^2+\alpha\|\mathbf{B}^{(t)\top}\vec{\mathbf{B}}^{(t)}-r\mathbf{S}_{oc}^{(t)}\|^2+\alpha\|\vec{\mathbf{B}}^{(t)\top}\vec{\mathbf{B}}^{(t)}-r\mathbf{S}_{cc}^{(t)}\|^2$$

$$+\beta\|\vec{\mathbf{L}}^{(t)}-\mathbf{P}^{(t)}\vec{\mathbf{B}}^{(t)}\|^2+\beta\|\mathbf{L}^{(t)}-\mathbf{P}^{(t)}\mathbf{B}^{(t)}\|^2+\beta\gamma\|\mathbf{P}^{(t)}\|^2,\ s.t.\ \vec{\mathbf{B}}^{(t)}\in\{-1,1\}^{r\times n_t}.$$

~~$\alpha\|\mathbf{B}^{(t)\top}\mathbf{B}^{(t)}-r\mathbf{S}_{oo}^{(t)}\|^2$~~

  - solve update-imbalance problem

$$\min_{\{\vec{\mathbf{B}},\vec{\mathbf{V}},\mathbf{P},\mathbf{R}\}^{(t)}} \alpha\|\vec{\mathbf{V}}^{(t)\top}\mathbf{B}^{(t)}-r\mathbf{S}_{co}^{(t)}\|^2+\alpha\|\mathbf{V}^{(t)\top}\vec{\mathbf{B}}^{(t)}-r\mathbf{S}_{oc}^{(t)}\|^2+\alpha\|\vec{\mathbf{V}}^{(t)\top}\vec{\mathbf{B}}^{(t)}-r\mathbf{S}_{cc}^{(t)}\|^2$$

$$+\beta\|\vec{\mathbf{L}}^{(t)}-\mathbf{P}^{(t)}\vec{\mathbf{V}}^{(t)}\|^2+\beta\|\mathbf{L}^{(t)}-\mathbf{P}^{(t)}\mathbf{V}^{(t)}\|^2+\beta\gamma\|\mathbf{P}^{(t)}\|^2$$

$$+\|\vec{\mathbf{B}}^{(t)}-\mathbf{R}^{(t)}\vec{\mathbf{V}}^{(t)}\|^2+\|\mathbf{B}^{(t)}-\mathbf{R}^{(t)}\mathbf{V}^{(t)}\|^2,\ s.t.\ \vec{\mathbf{B}}^{(t)}\in\{-1,1\}^{r\times n_t},\ \mathbf{R}^{(t)}\mathbf{R}^{(t)\top}=\mathbf{I},\ \vec{\mathbf{V}}^{(t)}\vec{\mathbf{V}}^{(t)\top}=n_t\mathbf{I},\ \vec{\mathbf{V}}^{(t)}\mathbf{1}=\mathbf{0}.$$

Method –Hash codes learning

MIMA

- Efficient Discrete Optimization
  - alternatively and iteratively update $\{\vec{\mathbf{B}}, \vec{\mathbf{V}}, \mathbf{P}, \mathbf{R}\}^{(t)}$
  - auxiliary variables $\mathbf{C}_*^{(t-1)}$

e.g., $\mathbf{C}_1^{(t)} = \vec{\mathbf{L}}^{(t)} \vec{\mathbf{V}}^{(t)\top} + \mathbf{L}^{(t)} \mathbf{V}^{(t)\top}$

$$\begin{aligned} &\mathbf{L}^{(t)} \mathbf{V}^{(t)\top} \\ &= [\mathbf{L}^{(t-1)}, \vec{\mathbf{L}}^{(t-1)}][\mathbf{V}^{(t-1)}, \vec{\mathbf{V}}^{(t-1)}]^\top \\ &= \mathbf{L}^{(t-1)} \mathbf{V}^{(t-1)\top} + \vec{\mathbf{L}}^{(t-1)} \vec{\mathbf{V}}^{(t-1)\top} \\ &= \mathbf{C}_1^{(t-1)}. \end{aligned}$$

$\mathbf{C}_1^{(t-1)} \in \mathbb{R}^{c \times r}$ is calculated at the previous round. $\mathbf{C}_1^{(t)}$ is only relevant to new data, i.e., n_t.

$$\mathbf{C}_1^{(t)} = \vec{\mathbf{L}}^{(t)} \vec{\mathbf{V}}^{(t)\top} + \mathbf{C}_1^{(t-1)}$$

Method –Hash functions learning

MIMA

- Hash Mapping: linear regression

$$\min_{\mathbf{W}_m^{(t)}} \|\mathbf{B}^{(t)} - \mathbf{W}_m^{(t)}\mathbf{X}_m^{(t)}\|^2 + \xi \|\mathbf{W}_m^{(t)}\|^2,$$

- Online Learning: $\mathbf{B}^{(t)} = [\mathbf{B}^{(t)}, \vec{\mathbf{B}}^{(t)}]$

$$\min_{\mathbf{W}_m^{(t)}} \|\mathbf{B}^{(t)} - \mathbf{W}_m^{(t)}\mathbf{X}_m^{(t)}\|^2 + \|\vec{\mathbf{B}}^{(t)} - \mathbf{W}_m^{(t)}\vec{\mathbf{X}}_m^{(t)}\|^2 + \xi \|\mathbf{W}_m^{(t)}\|^2 .$$

- Efficient Optimization
  - auxiliary variables $\mathbf{H}_*^{(t-1)}$ and $\mathbf{F}_*^{(t-1)}$

- Out-of-Sample: $H_m^{(t)}(\mathbf{x}_m) = \mathrm{sign}(\mathbf{W}_m^{(t)}\mathbf{x}_m).$

Method-A summary

MIMA

- Step-1: hash codes learning

$$\min_{\{\vec{\mathbf{B}},\vec{\mathbf{V}},\mathbf{P},\mathbf{R}\}^{(t)}} \alpha \|\vec{\mathbf{V}}^{(t)\top}\mathbf{B}^{(t)} - r\mathbf{S}_{co}^{(t)}\|^2 + \alpha \|\mathbf{V}^{(t)\top}\vec{\mathbf{B}}^{(t)} - r\mathbf{S}_{oc}^{(t)}\|^2 + \alpha \|\vec{\mathbf{V}}^{(t)\top}\vec{\mathbf{B}}^{(t)} - r\mathbf{S}_{cc}^{(t)}\|^2$$
$$+\beta\|\vec{\mathbf{L}}^{(t)} - \mathbf{P}^{(t)}\vec{\mathbf{V}}^{(t)}\|^2 + \beta\|\mathbf{L}^{(t)} - \mathbf{P}^{(t)}\mathbf{V}^{(t)}\|^2 + \beta\gamma\|\mathbf{P}^{(t)}\|^2 \quad (1)$$
$$+\|\vec{\mathbf{B}}^{(t)} - \mathbf{R}^{(t)}\vec{\mathbf{V}}^{(t)}\|^2 + \|\mathbf{B}^{(t)} - \mathbf{R}^{(t)}\mathbf{V}^{(t)}\|^2,\ s.t.\ \vec{\mathbf{B}}^{(t)} \in \{-1,1\}^{r\times n_t},\ \mathbf{R}^{(t)}\mathbf{R}^{(t)\top} = \mathbf{I}, \vec{\mathbf{V}}^{(t)}\vec{\mathbf{V}}^{(t)\top} = n_t\mathbf{I}, \vec{\mathbf{V}}^{(t)}\mathbf{1} = \mathbf{0}.$$

- Step-2: hash functions learning

$$\min_{\mathbf{W}_m^{(t)}} \|\mathbf{B}^{(t)} - \mathbf{W}_m^{(t)}\mathbf{X}_m^{(t)}\|^2 + \|\vec{\mathbf{B}}^{(t)} - \mathbf{W}_m^{(t)}\vec{\mathbf{X}}_m^{(t)}\|^2 + \xi\|\mathbf{W}_m^{(t)}\|^2. \quad (2)$$

- Out-of-Sample: $\mathbf{b}_{query} = H_m^{(t)}(\mathbf{x}_m) = \mathrm{sign}(\mathbf{W}_m^{(t)}\mathbf{x}_m).$ (3)

- Retrieval: $\mathbf{b}_{query}$ ←hamming distance→ $\mathbf{B}^{(t)}$

Experiments

- Datasets
  - MIRFlickr-25K
  - IAPR TC-12
  - NUS-WIDE

MIRFlickr-25K

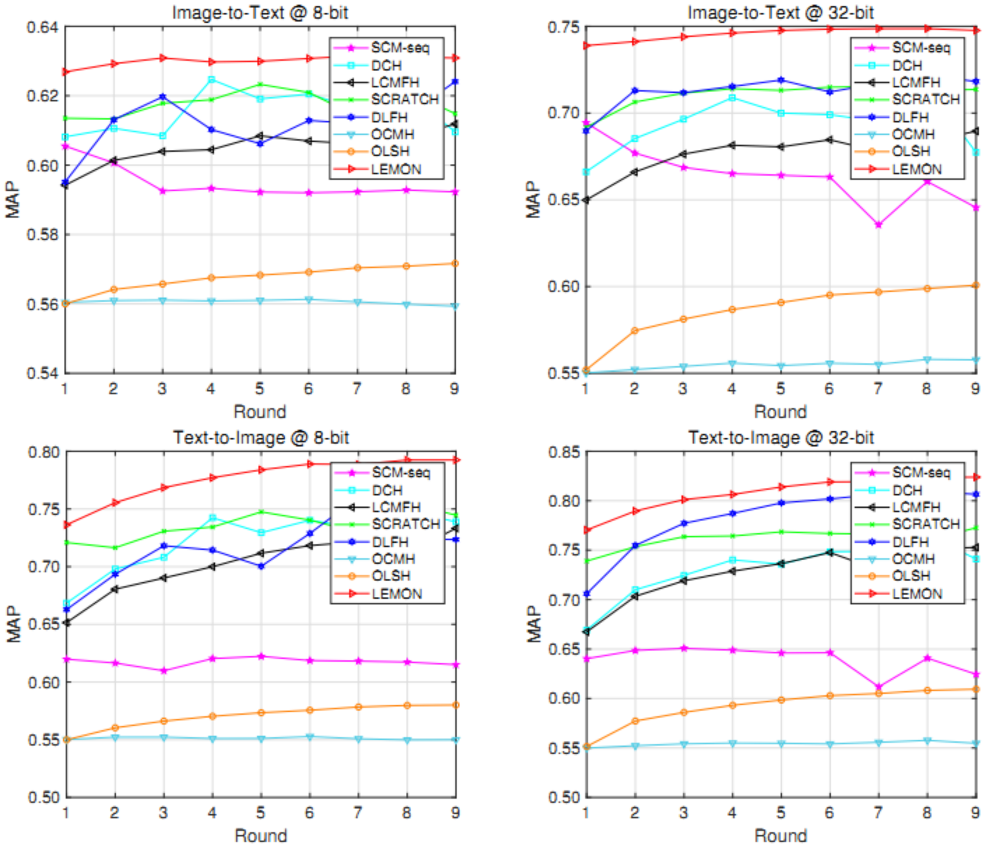
- Compared Methods
  - offline methods: SCM-seq, DCH, LCMFH, SCRATCH, DLFH
  - online methods: OCMH, OLSH
- Evaluation Metrics
  - Mean Average Precision (MAP), and Training Time

Experiments

MIMA

- The MAP results of all methods on MIRFlickr-25K

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \rightarrow T$	SCM-seq	0.6307	0.6457	0.6455	0.6629	0.6148
	DCH	0.6739	0.7093	0.6774	0.7219	0.7424
	LCMFH	0.6796	0.6750	0.6896	0.6898	0.7002
	SCRATCH	0.6870	0.7084	0.7136	0.7234	0.7256
	DLFH	0.7102	0.7076	0.7182	0.7188	0.7254
	OCMH	0.5484	0.5515	0.5578	0.5565	0.5536
	OLSH	0.5791	0.5778	0.6008	0.5971	0.5935
	LEMON	**0.7272**	**0.7258**	**0.7476**	**0.7474**	**0.7485**
$T \rightarrow I$	SCM-seq	0.6151	0.6257	0.6245	0.6512	0.6010
	DCH	0.7388	0.7649	0.7410	0.7786	0.8010
	LCMFH	0.7332	0.7293	0.7528	0.7627	0.7740
	SCRATCH	0.7446	0.7692	0.7727	0.7810	0.7877
	DLFH	0.7235	0.7836	0.8066	0.8225	0.8285
	OCMH	0.5500	0.5530	0.5547	0.5565	0.5533
	OLSH	0.5801	0.5829	0.6094	0.6038	0.6024
	LEMON	**0.7924**	**0.8166**	**0.8238**	**0.8298**	**0.8327**

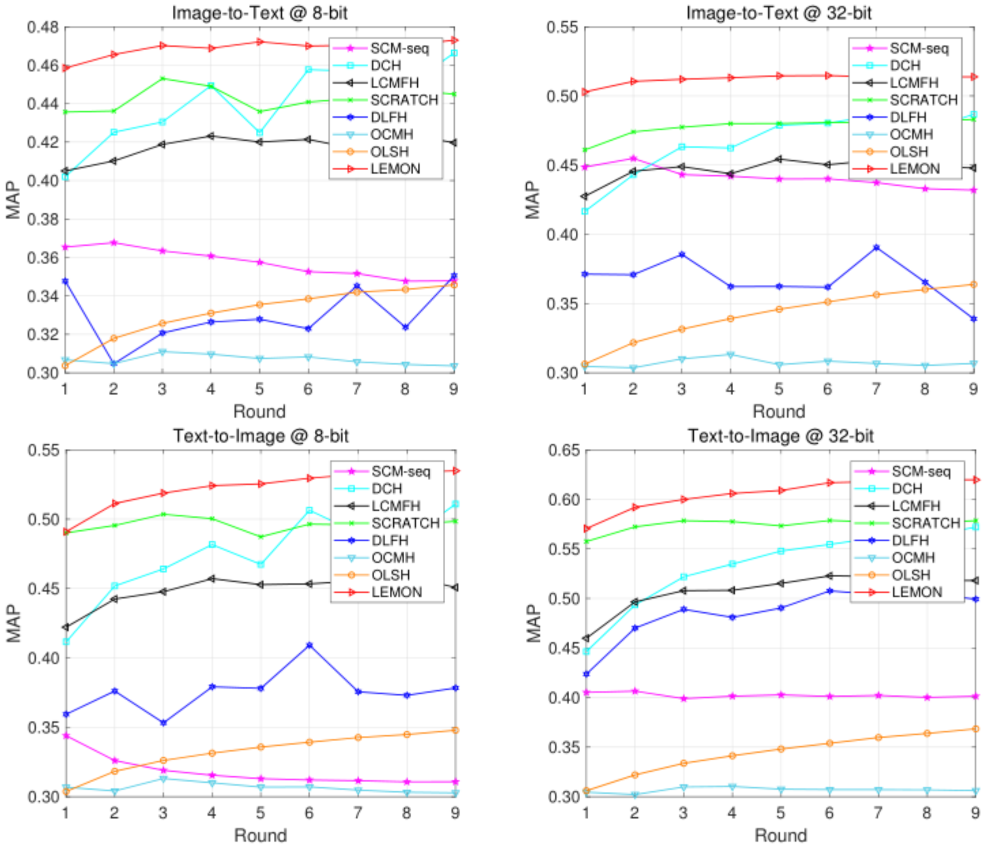


Experiments

MIMA

- The MAP results of all methods on IAPR TC-12

Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \to T$	SCM-seq	0.3479	0.4047	0.4320	0.4444	0.3917
	DCH	0.4664	0.4825	0.4867	0.5061	0.5203
	LCMFH	0.4197	0.4359	0.4480	0.4626	0.4688
	SCRATCH	0.4449	0.4588	0.4831	0.4949	0.4955
	DLFH	0.3505	0.3415	0.3390	0.3644	0.3913
	OCMH	0.3037	0.3105	0.3069	0.3046	0.3064
	OLSH	0.3457	0.3335	0.3639	0.3486	0.3617
	LEMON	**0.4730**	**0.4998**	**0.5138**	**0.5318**	**0.5431**
$T \to I$	SCM-seq	0.3108	0.3532	0.4014	0.4077	0.3624
	DCH	0.5109	0.5444	0.5720	0.5948	0.6243
	LCMFH	0.4508	0.4977	0.5181	0.5424	0.5598
	SCRATCH	0.4984	0.5385	0.5784	0.6052	0.6201
	DLFH	0.3784	0.4170	0.4992	0.5742	0.6174
	OCMH	0.3028	0.3099	0.3062	0.3048	0.3064
	OLSH	0.3479	0.3354	0.3685	0.3565	0.3683
	LEMON	**0.5348**	**0.5822**	**0.6197**	**0.6519**	**0.6708**

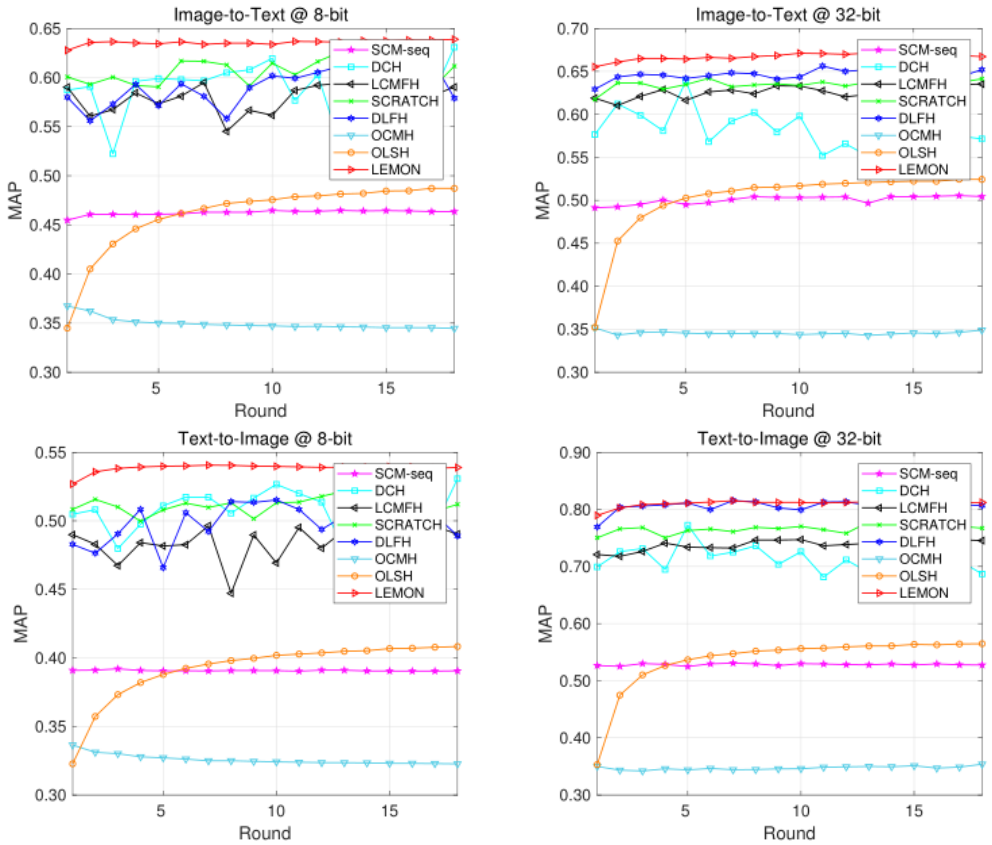


Experiments

MIMA

- The MAP results of all methods on NUS-WIDE

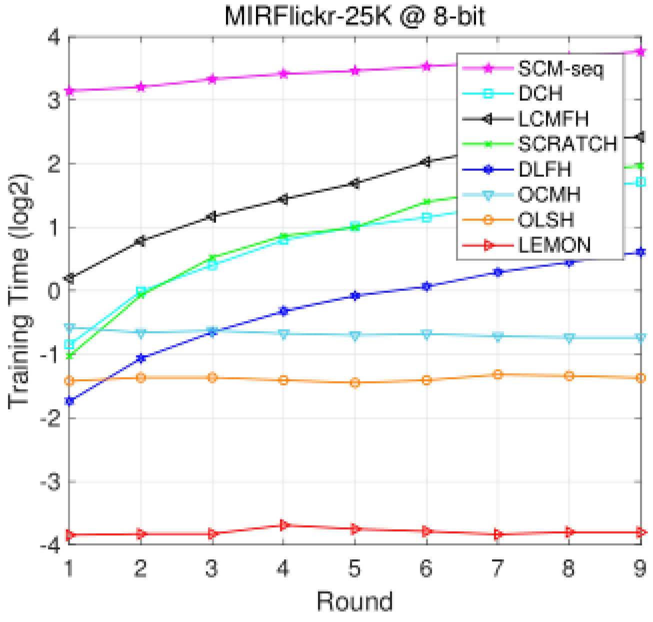
Task	Method	8-bit	16-bit	32-bit	64-bit	128-bit
$I \rightarrow T$	SCM-seq	0.4636	0.4324	0.5045	0.4941	0.4941
	DCH	0.6309	0.6076	0.5714	0.5835	0.5937
	LCMFH	0.5903	0.6047	0.6353	0.6354	0.6371
	SCRATCH	0.6116	0.6184	0.6413	0.6487	0.6468
	DLFH	0.5789	0.6211	0.6519	0.6617	0.6684
	OCMH	0.3447	0.3411	0.3491	0.3429	0.3439
	OLSH	0.4872	0.5162	0.5243	0.5368	0.5270
	LEMON	**0.6389**	**0.6579**	**0.6672**	**0.6711**	**0.6700**
$T \rightarrow I$	SCM-seq	0.4812	0.4536	0.5276	0.5253	0.5286
	DCH	0.7615	0.7419	0.6868	0.7110	0.7349
	LCMFH	0.6807	0.7275	0.7455	0.7549	0.7552
	SCRATCH	0.7241	0.7567	0.7673	0.7853	0.7893
	DLFH	0.6779	0.7736	0.8066	0.8071	0.8140
	OCMH	0.3454	0.3420	0.3539	0.3417	0.3436
	OLSH	0.5167	0.5594	0.5648	0.5920	0.5804
	LEMON	**0.7778**	**0.7946**	**0.8122**	**0.8288**	**0.8333**

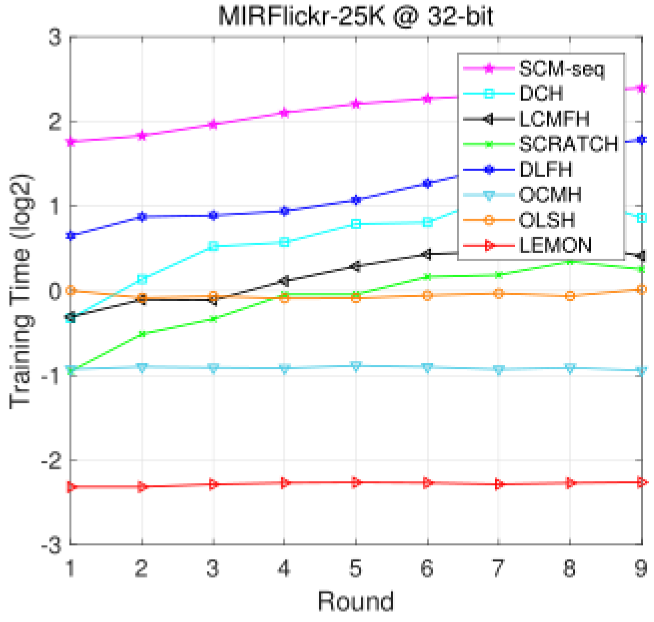


Experiments

MIMA

- Training time (log2 seconds) and efficiency on MIRFlickr-25K





- For more results, please refer to our paper.

Conclusion and Future work

MIMA

- A novel label embedding online hashing method, i.e., LEMON.
  - capturing the semantic structure by label embedding
  - performing online learning via a block similarity matrix
  - efficient and discrete optimization
- Future Work
  - deep-to-deep
  - semi-supervised

Thank You!

Any Question?

Yongxin Wang: yxinwang@hotmail.com