Collaborative Hashing

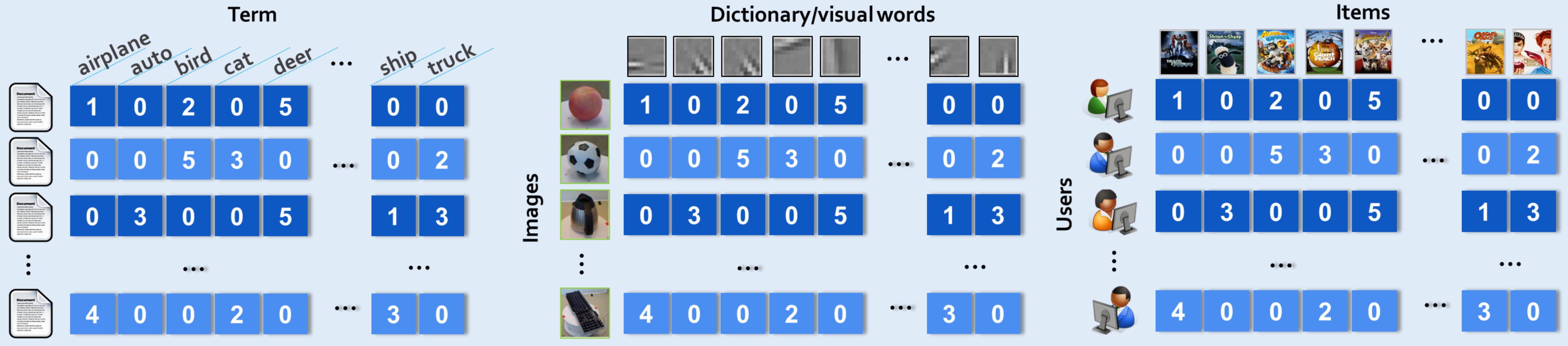
Xianglong Liu[1], Junfeng He[2], Cheng Deng[3] and Bo Lang[1]

[1]Beihang University, [2]Facebook, [3]Xidian University

IEEE 2014 Conference on Computer Vision and Pattern Recognition

1 Overview

- **Observation:** many scenarios involve nearest neighbor search on the data in matrix form, where two different types of, yet naturally associated entities respectively correspond to its two dimensions or views.



- **Problem:** most hashing research pursue the binary codes for the same type of entities only relying on their intra-similarities.
- **Motivation:** the semantic correlations between the coupled views together with the intra-view similarities will help enhance the discriminative power of the hash codes for each view by discovering their collective patterns and duality
- **Applications: a novel generic hashing scheme** for the fast and accurate (1) ***search inside a single view***: *eg.*, the visual search using local descriptors, and (2) ***search across different views***: *eg.*, recommendation using user-item rating matrix

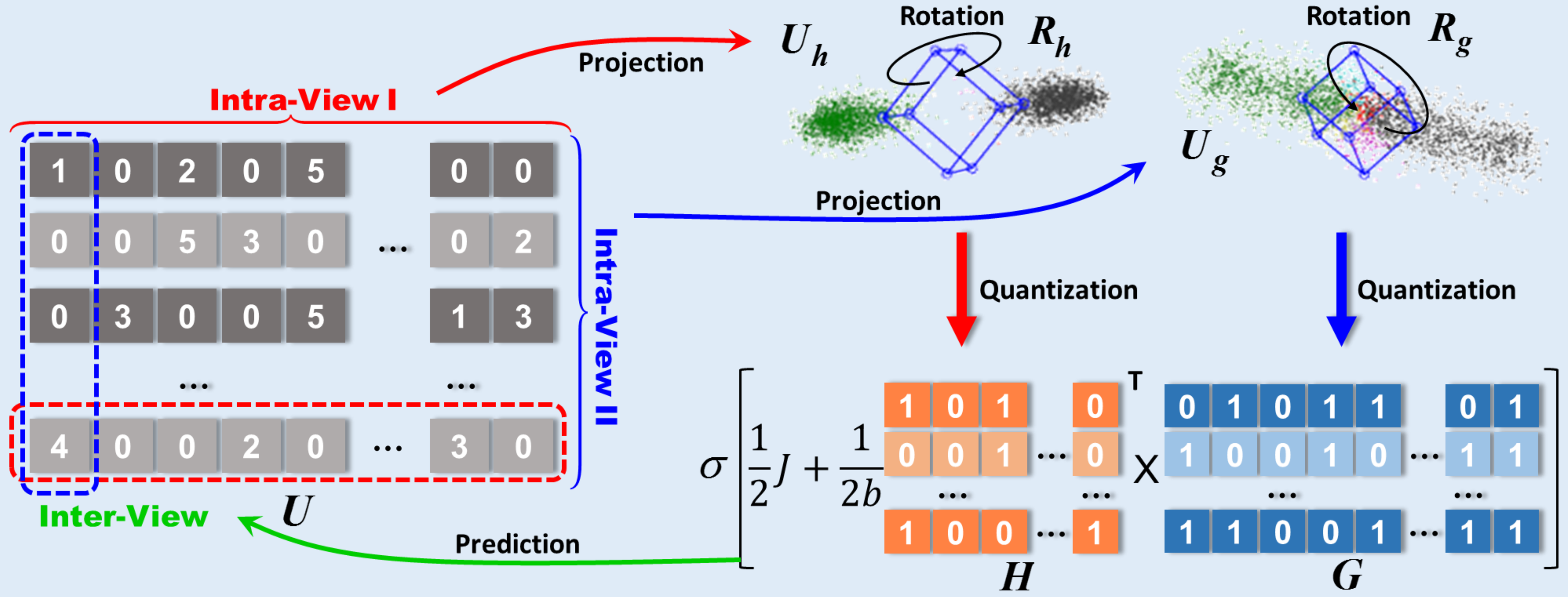
2 Collaborative Hashing

- **Formulation:** exploit the intrinsic relations in matrix data: both neighbor structures in each view and semantic correlations between views correspondingly by minimizing:

binary quantization loss: $E_{\text{quan}} = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{h}_i - R_h\tilde{\mathbf{u}}_i\|^2 + \frac{1}{m}\sum_{j=1}^{m}\|\mathbf{g}_j - R_g\hat{\mathbf{u}}_j\|^2$

correlation prediction loss: $E_{\text{rate}} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}(\sigma c(\mathbf{h}_i,\mathbf{g}_j) - u_{ij})^2$ $\quad c(\mathbf{h}_i,\mathbf{g}_j) = \frac{1}{2} + \frac{1}{2b}\mathbf{h}_i^T\mathbf{g}_j$ correlation prediction

s.t. constraints: $R_h^T R_h = I,\ R_g^T R_g = I$ orthogonal rotation $\quad HH^T = nI$ and $GG^T = mI$ independent codes



- **Optimization:** alternate among variables

binary codes H and G: singular value decomposition

rotation matrices R_h and R_g: singular value decomposition

Iterative Quantization [Gong et al. 2011] in each view, considering view correlations

prediction scalar: least square problem $\min_{\sigma} \left\|\sigma\left(\frac{1}{2}+\frac{1}{2b}H^T G\right) - U\right\|_F^2$

Preserve view correlations by matching the predictions and the true observations

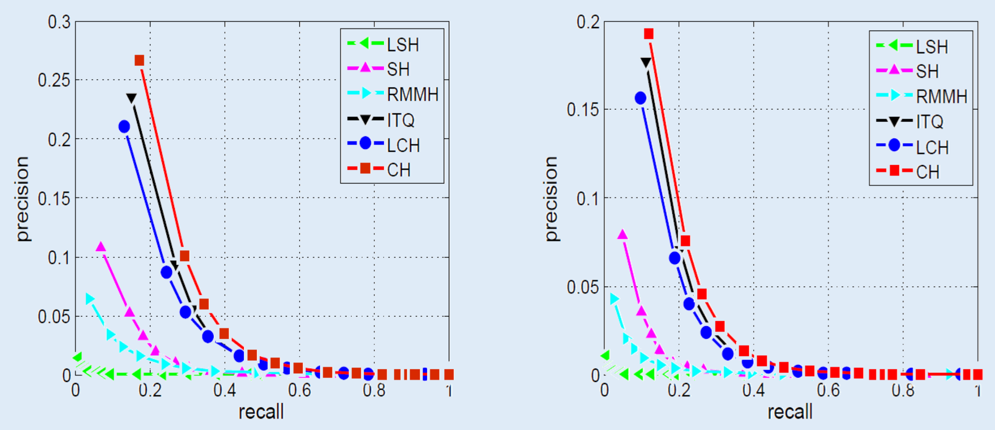
- **Applications:** a general hashing framework for different search paradigms

search inside a single view: finding similar entities of the same type, where U is dense with all elements observed

search across different views: seeking correlated entities of different types, where U is sparse $E_{\text{rate}} = \frac{1}{\|A\|_1}\left\|\sigma\left(\frac{1}{2}+\frac{1}{2b}H^T G\right)\circ A - U\right\|_F^2$

3 Experiments

- **Search with Bag of Words**



P-R @ 64 bits on Holidays+25K

P-R @ 64 bits on Holidays+100K

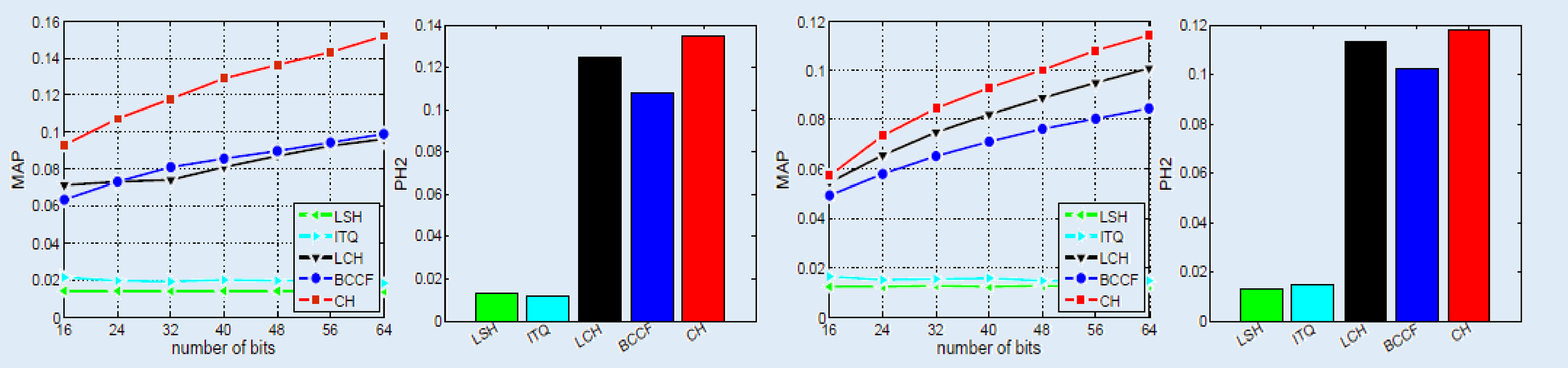
Table 1. MAP (%) of different hashing algorithms using 32 - 128 bits on Holidays dataset.

Holidays	Hash	32 bits	64 bits	128 bits
+15K	LSH	1.63±0.29	3.63±0.34	7.05±0.51
	SH	12.65±0.67	16.36±0.60	21.19±1.32
	RMMH	8.00±0.87	11.28±1.40	16.56±1.64
	ITQ	18.86±0.75	25.42±0.61	30.73±0.72
	LCH	17.96±0.47	25.95±0.72	31.87±0.50
	CH	**20.95**±0.82	**27.54**±0.49	**32.34**±1.30
+25K	LSH	1.01±0.19	1.68±0.24	4.38±0.56
	SH	8.84±1.13	11.76±0.75	15.59±1.05
	RMMH	5.39±1.03	7.59±1.14	11.60±0.40
	ITQ	16.49±0.81	22.92±0.96	28.52±0.71
	LCH	12.67±1.13	20.94±0.82	29.73±0.77
	CH	**18.42**±0.56	**25.61**±0.64	**31.23**±0.51
+100K	LSH	0.56±0.12	1.14±0.13	2.69±0.19
	SH	8.00±0.91	8.03±1.13	9.09±0.57
	RMMH	4.22±0.43	4.98±0.36	7.67±0.93
	ITQ	12.34±0.60	17.21±1.38	22.18±1.11
	LCH	8.67±0.83	15.64±1.27	22.41±0.70
	CH	**12.53**±0.62	**18.53**±0.80	**24.02**±0.23

- **Recommend using User-Item Ratings**

Table 2. NDCG (%) of different methods using 16 - 64 bits on MovieLens and Netflix.

Datasets	Hash	16 bits		32 bits		48 bits		64 bits	
		NDCG@5	NDCG@10	NDCG@5	NDCG@10	NDCG@5	NDCG@10	NDCG@5	NDCG@10
MovieLens	LSH	2.99±0.22	3.00±0.23	2.89±0.13	2.88±0.09	2.90±0.14	2.91±0.13	2.93±0.22	2.90±0.20
	ITQ	4.35±0.72	4.38±0.74	3.97±0.59	3.94±0.62	4.24±0.86	4.10±0.70	4.00±0.52	3.92±0.46
	LCH	22.72±1.43	20.77±1.26	23.52±0.73	21.77±0.69	27.64±0.95	25.43±0.76	30.89±0.69	28.24±0.57
	BCCF	20.87±0.42	19.02±0.33	26.11±0.42	23.24±0.30	28.43±0.29	25.12±0.16	30.47±0.27	26.71±0.18
	CH	**28.26**±0.70	**26.39**±0.55	**36.15**±0.40	**33.09**±0.42	**41.07**±0.49	**37.28**±0.37	**45.06**±0.37	**40.64**±0.28
Netflix	LSH	2.57±0.16	2.57±0.16	2.69±0.21	2.65±0.18	2.67±0.20	2.65±0.19	2.62±0.12	2.63±0.10
	ITQ	3.75±0.82	3.64±0.63	3.38±0.71	3.49±0.68	3.07±0.60	3.10±0.59	3.17±0.51	3.25±0.48
	LCH	20.92±0.70	19.50±0.65	28.04±0.38	25.93±0.34	33.00±0.45	30.21±0.26	36.97±0.24	33.50±0.23
	BCCF	20.12±0.66	18.79±0.56	28.04±0.45	25.70±0.36	31.90±0.55	29.27±0.34	34.59±0.60	31.62±0.46
	CH	**23.75**±0.55	**22.25**±0.43	**33.48**±0.38	**31.02**±0.34	**38.41**±0.43	**35.45**±0.32	**42.55**±0.51	**38.95**±0.39



(a) MAP on MovieLens (b) PH2 @ 16 bits on MovieLens (c) MAP on Netflix (d) PH2 @ 16 bits on Netflix

4 Conclusions

- a collaborative hashing scheme for data in matrix form that can learn hash codes for both types of entities in the matrix
- an efficient optimization method to learn hash functions simultaneously exploiting both intra- and inter-view similarities
- two classic search applications based on collaborative hashing