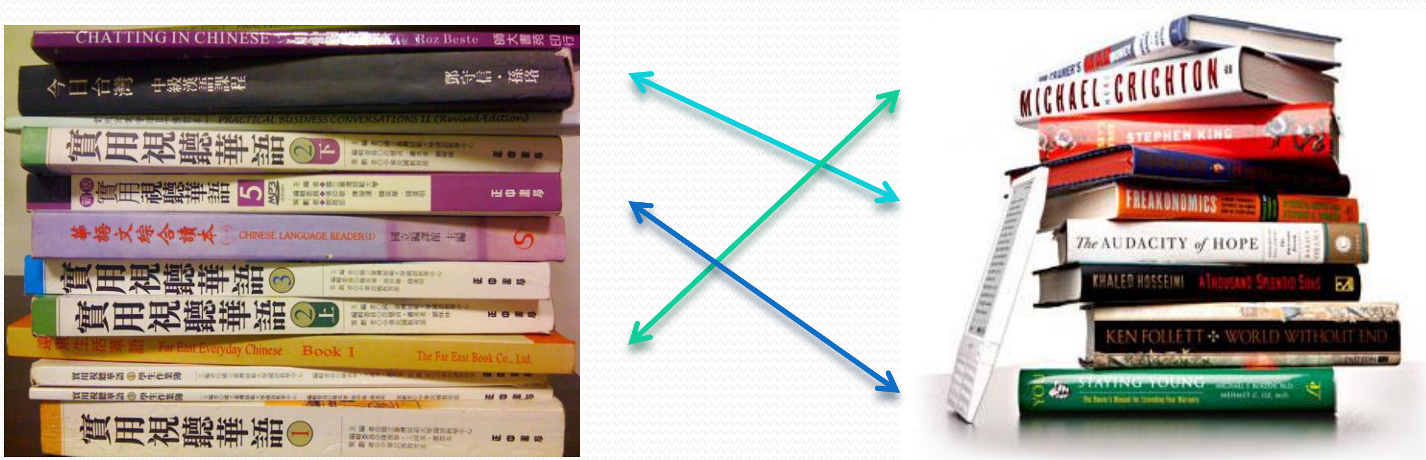
Convex Kernelized Sorting

Nemanja Djuric, Mihajlo Grbovic, Slobodan Vucetic
Temple University, Philadelphia

AAAI-2012
Toronto, July 24^{th}, 2012

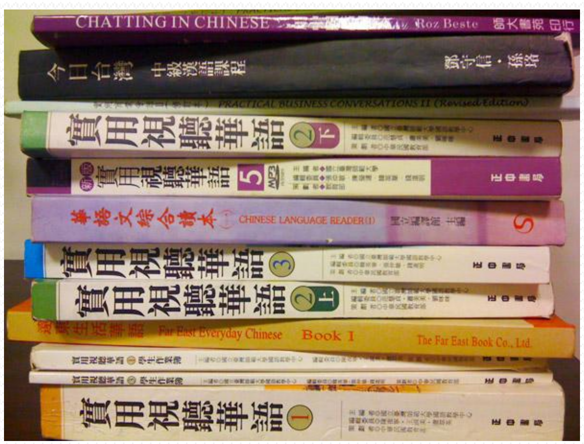
Introduction

- Object matching
  - Match objects from one set with objects from another set, so that matched pairs are ‘similar’
  - E.g., graph matching (computer vision), multiple sequence alignment (bioinformatics),…



Introduction

- Matching Chinese and English texts is a relatively simple task if we can directly compare them
  - E.g., if we have a bilingual Chinese-English dictionary, or if we use Google Translate



Previous work

- So, how to match objects from different domains without cross-domain similarity measure?
- Several proposed ideas:
  - Information about local relationships is translated into cross-domain similarity measure (*Wang et al*. *2009* - Manifold Alignment, *Quadrianto et al*. *2010* - Kernelized Sorting)
  - Project to common space and then directly compare objects (*Haghighi et al*. *2008*, *Tripathi et al*. *2011* - Canonical Correlation Analysis)

Formalities

- Let $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_m\}$ be two sets of objects from two different domains $\mathcal{X}$ and $\mathcal{Y}$
- Let k: $\mathcal{X} \times \mathcal{X} \to R$ and l: $\mathcal{Y} \times \mathcal{Y} \to R$ be two kernels associated with domains $\mathcal{X}$ and $\mathcal{Y}$
- Given $m \times m$ matrices K and L, the task is to find 1-to-1 matching between X and Y
  - This correspondence is encoded in $m \times m$ permutation matrix π, where $\pi_{ij} = 1$ if x_j and y_i objects are matched, and $\pi_{ij} = 0$ otherwise

Kernelized Sorting

- Use Hilbert-Schmidt Independence Criterion (HSIC) to measure the dependency between two aligned sets

$$\Delta^2 = m^{-2} \cdot trace(K \cdot L)$$

- Find an alignment π by solving

$$\pi^* = \arg\max_{\pi \in \Pi_m} trace(K \cdot \pi^{\mathrm{T}} \cdot L \cdot \pi)$$

- However, there is a slight problem
  - The optimization involves ***maximization*** over convex function of matrix π (local optima issue!)

Kernelized Sorting extensions

- Several methods to mitigate the instability issue of KS have been proposed
  - *Jagarlamudi et al. 2010* proposed *p*-smooth which smoothes out the kernel matrices, resulting in more robust method (**KS-*p***)
  - *Yamada et al. 2011* proposed using Least-Squares Mutual Information instead of HSIC (**LSOM**)
  - Both ideas result in more accurate algorithms, but the instability issue still remains

Convex Kernelized Sorting

- We present a convex formulation of the Kernelized Sorting problem
- First, rewrite KS problem into equivalent

$$\pi^* = \arg\max_{\pi \in \Pi_m} trace\,(K \cdot \pi^{\mathrm{T}} \cdot L \cdot \pi) \Longrightarrow \pi^* = \arg\min_{\pi \in \Pi_m} \| K \cdot \pi^{\mathrm{T}} - (L \cdot \pi)^{\mathrm{T}} \|_F^2$$

- Next, we allow π to be a doubly-stochastic matrix
  - Constraint $\pi_{ij} \in \{0, 1\}$ becomes $\pi_{ij} \in [0, 1]$
  - Rows and columns sum up to 1

Convex Kernelized Sorting

- Finally, CKS optimization problem becomes:

$$\begin{aligned} &\underset{\pi}{\text{minimize}} \| K \cdot \pi^{\mathrm{T}} - (L \cdot \pi)^{\mathrm{T}} \|_F^2 \\ &\text{s.t.} \qquad \pi_{ij} \geq 0, \ \text{for}\, i, j \in \{1, 2, \ldots, m\} \\ &\qquad\quad \pi \cdot \mathbf{1}_m = \mathbf{1}_m \\ &\qquad\quad \pi^{\mathrm{T}} \cdot \mathbf{1}_m = \mathbf{1}_m \end{aligned}$$

- Unlike KS, problem is now convex in π, lending itself to effective computational methods and globally-optimal solution
- Also, real-valued π_{ij}'s give us informative soft matches, instead of hard ones

Convex Kernelized Sorting

- But how to calculate hard, one-to-one matches?
- We need to solve Linear Assignment Problem (LAP) defined by a learned π
  - LAP – if we have *m* workers each requiring certain pay to work on one of *m* jobs, find lowest-cost 1-to-1 assignment of workers to jobs
  - We use the Hungarian algorithm, effective, polynomial-time algorithm for solving LAP

Experiment 1: Data visualization

- We used data set of 320 images, used in most previous object matching papers
- Gaussian RBF kernel was used, $k(x, x') = l(x, x') = exp(-\gamma \|x - x'\|^2)$, setting γ to inverse median of $\|x - x'\|^2$
- Results:

Random placement

After matching

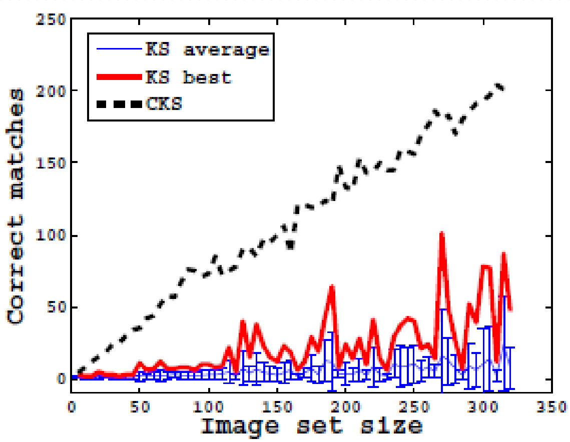
Experiment 2: Image matching

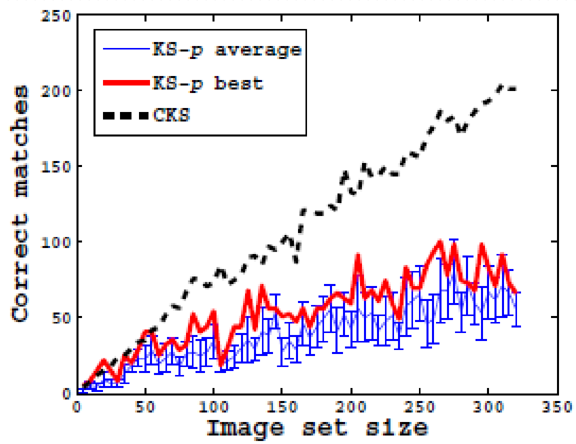
- We split 320 images to left and right halves, and retrieve the original images by matching two sets

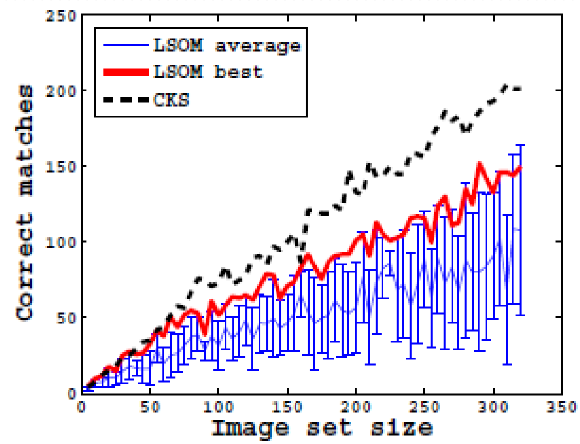
206 of 320 images correctly matched (64%)

Experiment 2: Image matching

- Comparison to existing methods
- Increase image set size from 5 to 320
- Average and best results after 10 runs reported







Experiment 3: NLP

- *Europarl Parallel Corpus* data set (with linear kernel)
- Task: Match English documents to documents in 9 European languages
- Average and highest number of correct matches after 10 runs reported

Baseline: match objects using document length

Language	**Corpus size**	Baseline	KS	KS-p	LSOM	CKS
Danish	387	39	261 (318)	258 (273)	159 (173)	**379**
Dutch	387	50	266 (371)	237 (317)	146 (375)	**383**
Finnish	308	54	19 (32)	22 (38)	10 (10)	**114**
French	356	64	319 (356)	320 (334)	354 (354)	**356**
German	356	50	282 (344)	258 (283)	338 (350)	**356**
Italian	387	49	341 (382)	349 (353)	378 (381)	**385**
Portuguese	356	46	308 (354)	326 (343)	342 (356)	**356**
Spanish	387	48	342 (365)	351 (364)	386 (387)	**387**
Swedish	337	76	20 (39)	20 (33)	5 (5)	**97**

Conclusion

- We presented a novel, convex formulation of the object matching problem
- Globally-optimal solution leads to much improved results
- Ongoing/future work
  - Explore a semi-supervised extension of the method
  - What to do when two sets have different number of objects?

Thank you!

- Questions?

Experiments

- Visualization of a learned matrix π