Learning Dissimilarities by Ranking From SDP to QP

Hua Ouyang, Alexander Gray

FASTLab
College of Computing
Georgia Institute of Technology

07/07/2008 ICML

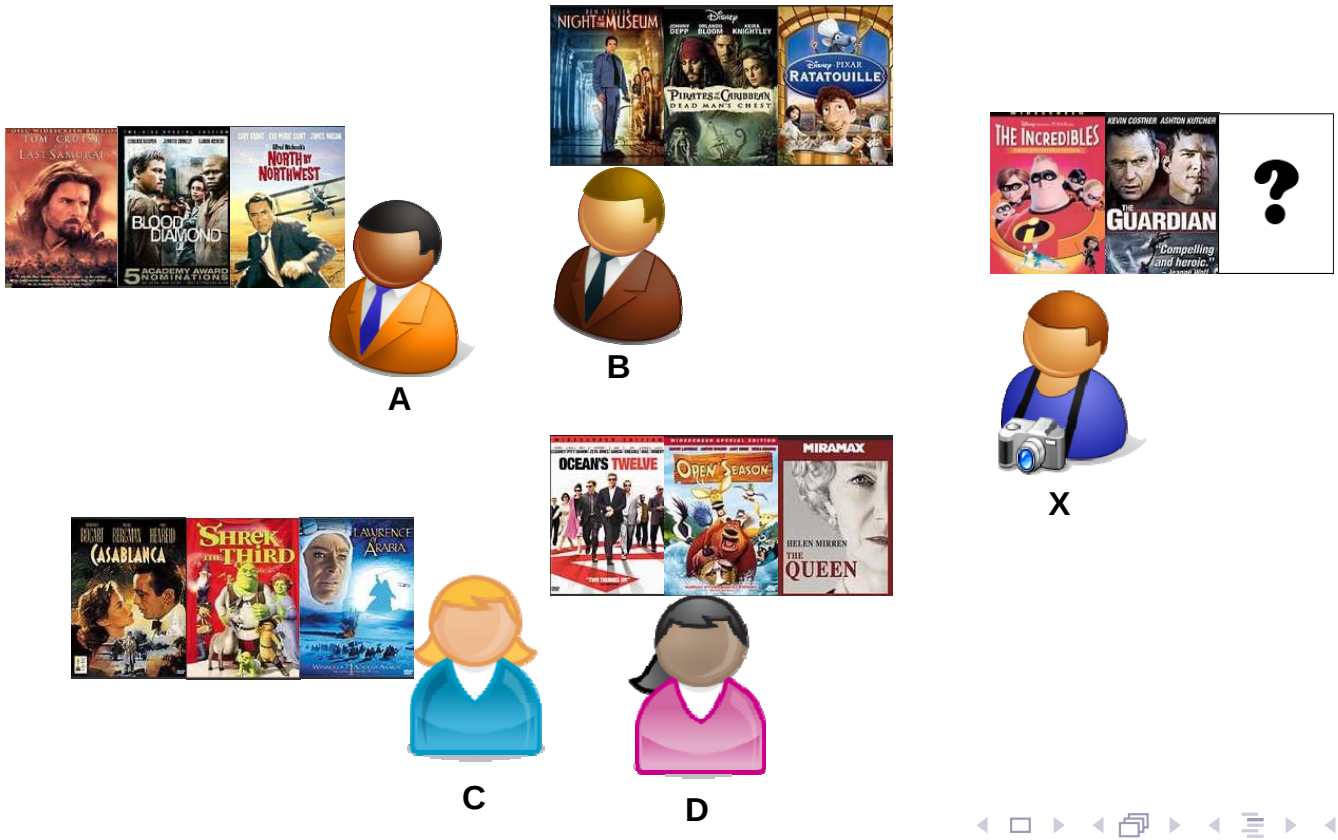
Outlines

Introduction and Motivations

- Ranking: learning of orderings: “A ranks lower than B” or “C ranks higher than D”
- SVM Ranking, BoostRank
- Our focus: learning rankings of dissimilarities between two samples (d-ranking)
- “A is more similar to B than C is to D” or “The distance between E and F is larger than that between G and H”
- Not well studied; nonmetric MDS (NMDS), Generalized NMDS (GNMDS)
- Kind of dissimilarity (metric) learning; optimal Mahanalobis distance, kernel learning, Multidimensional scaling
- Preserve metrics (distances) v.s. preserve dissimilarities

Applications of d-ranking

- Any application? Exact distances not exist / not accurate
- Movie recommendation (e.g. Netflix), protein folding, social science, information retrieval...



Problem Formulation of d-ranking

Different formulations:

- F1- Input: ranks $d_{ij} \leq d_{kl}$; Output: coefficients in $\mathbb{R}^L$, and assume Euclidean metric in $\mathbb{R}^L$
- F2- Input: coefficients in $\mathbb{R}^D$, ranks $d_{ij} \leq d_{kl}$; Output: explicit dissimilarity: $\hat{d} : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$
- F3- Input: coefficients in $\mathbb{R}^D$, ranks $d_{ij} \leq d_{kl}$; Output: $f : \mathbb{R}^D \rightarrow \mathbb{R}^L$, coefficients in $\mathbb{R}^L$

We will investigate F1 and F2 in this project. F3 will be future work.

F1- Basic Ideas

- F1- Input: ranks $d_{ij} \leq d_{kl}$; Output: coefficients in $\mathbb{R}^L$, and assume Euclidean metric
- Goal: find a proper Gram matrix K: $K_{mn} = \langle \mathbf{x}_m, \mathbf{x}_n \rangle$
- $\|x_i - x_j\|_2^2 = K_{ii} - 2K_{ij} + K_{jj}$
- Recover low dimensional embedding by eigen-decomposing K
- $K \succeq 0$, semidefinite programming

F1- Solution by SDP

Generalized nonmetric multidimensional scaling (GNMDS) [1]

$$\begin{aligned}
\min\ & \sum_{ijkl} \xi_{ijkl} + \lambda \mathrm{tr}(K) \\
\text{s.t.}\ & (K_{kk} - 2K_{kl} + K_{ll}) - (K_{ii} - 2K_{ij} + K_{jj}) + \xi_{ijkl} \geq 1 \\
& \sum_{ab} K_{ab} = 0 \\
& \xi_{ijkl} \geq 0,\ K \succeq 0,
\end{aligned}$$

Semidefinite programming, can be solved by SeDuMi, CSDP, SDPT3 etc.

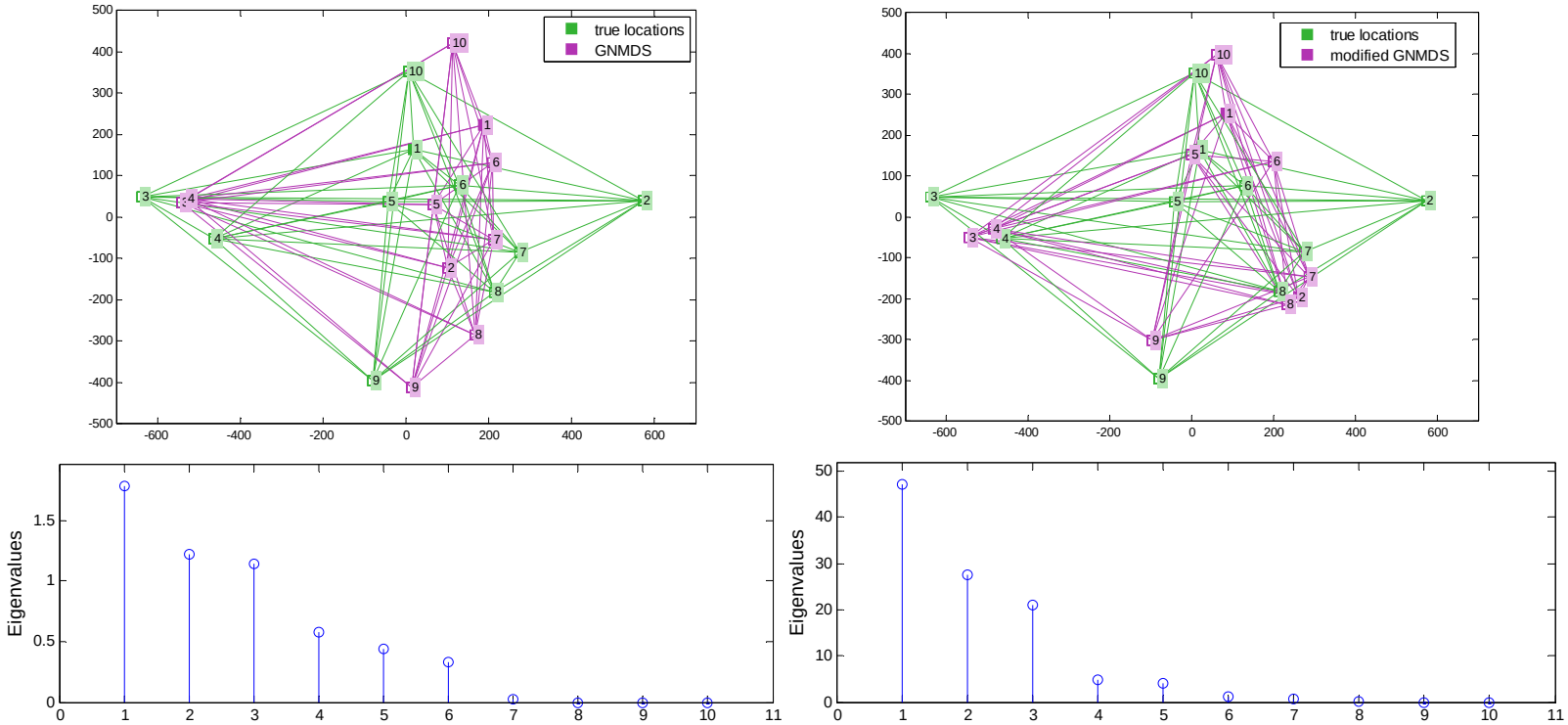
F1- Solution by SDP (cont.)

Modified GNMDS

$$\begin{aligned}
\min \ & \sum_{ijkl} \xi_{ijkl} + \lambda \mathrm{tr}(K) \\
\text{s.t.} \ & (K_{kk} - 2K_{kl} + K_{ll}) - (K_{ii} - 2K_{ij} + K_{jj}) - \xi_{ijkl} \geq 1 \\
& \sum_{ab} K_{ab} = 0 \\
& \xi_{ijkl} \geq 0, \, K \succeq 0,
\end{aligned}$$

Ensure that differences between distances ≥ 1. Pull embedding samples apart

Example: City Location Recovery



10 cities in Europe, 850 rank pairs

2D embedding space, error rate: 20.3% vs 14.14%

F2- Basic Ideas

- F2- Input: coefficients in $\mathbb{R}^D$, ranks $d_{ij} \leq d_{kl}$; Output: $\hat{d} : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$
- Goal: estimate a proper dissimilarity measure $d(\cdot, \cdot)$, based on limited number of ranking information
- Minimize empirical error, while control the complexity of $\hat{d}$, SRM, Regularized dissimilarity learning
- Reminiscent of large margin classifiers, e.g. SVM

F2- Solution by QP

d-ranking-VM, *primal* problem:

$$\begin{aligned} \min \ & \frac{1}{N}\sum_{ijkl}\xi_{ijkl} + \lambda\|d\|^2_{\underline{\mathbb{H}}} \\ \text{s.t. } & \forall i,j,k,l \subseteq \mathcal{S} \\ & d(\mathbf{x}_k,\mathbf{x}_l) - d(\mathbf{x}_i,\mathbf{x}_j) - \xi_{ijkl} \geq 1, \\ & \xi_{ijkl} \geq 0 \end{aligned}$$

where $\mathcal{S}$ is the ranking set, diss$(k,l) >$ diss(i,j).

- Solving this problem needs the following representer theorem for Hyper-RKHS:

Representer Theorem for Hyper-RKHS

Theorem

Denote by $\Omega : [0, \infty) \to \mathbb{R}$ *a strictly monotonic increasing function, by* $\mathbb{X}$ *a set, and by* $L : (\mathbb{X}^2 \times \mathbb{R}^2)^M \to \mathbb{R} \cup \{\infty\}$ *an arbitrary loss function. Then each minimizer* $d \in \underline{\mathbb{H}}$ *of the regularized risk*

$$L\Big((\underline{\boldsymbol{x}}_1, y_1, d(\underline{\boldsymbol{x}}_1)), \ldots, (\underline{\boldsymbol{x}}_M, y_M, d(\underline{\boldsymbol{x}}_M)) \Big) + \Omega\big(\|d\|_{\underline{\mathbb{H}}}\big)$$

admits a representation of the form $d(\underline{\boldsymbol{x}}) = \sum_{i=1}^{M} c_i \underline{k}\big(\underline{\boldsymbol{x}}_i, \underline{\boldsymbol{x}}\big)$, *where* $c_i \in \mathbb{R}$ *and* $\underline{\mathbb{H}}$ *is a hyper-RKHS induced by the hyperkernel* $\underline{k}$.

F2- Solution by QP

Utilizing KKT conditions, we arrive at the *dual* problem:

$$\max \sum_{ijkl} \alpha_p - \frac{A^T(Q-P)^T \underline{K}(Q-P)A}{4\lambda}$$

$$\text{s.t. } \alpha_p \geq 0$$

$$\forall i,j,k,l \subseteq \mathcal{S}$$

where $\underline{K} \in \mathbb{R}^{N\times N}$ is the hyper-kernel matrix; M: number of dissimilarities, $N = |\mathcal{S}|$, $A \in \mathbb{R}^N$ is a vector with the pth element being α_p; $P, Q \in \mathbb{R}^{M\times N}$ contain the rank information.

- Quadratic programming problem. Can be solve by general purpose opt tools or specialized sequential methods, e.g. SMO.

F2- Hyperkernels [2,3]

Proposition

Let $k_a(\cdot,\cdot)$ and $k_b(\cdot,\cdot)$ be positive definite kernels, then $\forall \boldsymbol{x}_1, \boldsymbol{x}_1', \boldsymbol{x}_2, \boldsymbol{x}_2' \in \mathbb{X}$, and $\forall \alpha, \beta > 0$, $\left(k_a(\boldsymbol{x}_1, \boldsymbol{x}_2)\right)^{\alpha}\left(k_b(\boldsymbol{x}_1', \boldsymbol{x}_2')\right)^{\beta}$ and $\alpha k_a(\boldsymbol{x}_1, \boldsymbol{x}_2) + \beta k_b(\boldsymbol{x}_1', \boldsymbol{x}_2')$ can give a hyperkernel $\underline{k}$.

- $\underline{k}\big((\mathbf{x}_1, \mathbf{x}_1'), (\mathbf{x}_2, \mathbf{x}_2')\big) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2 + \|\mathbf{x}_1' - \mathbf{x}_2'\|^2}{2\sigma^2}\right)$
- $\underline{k}\big((\mathbf{x}_1, \mathbf{x}_1'), (\mathbf{x}_2, \mathbf{x}_2')\big) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}\right) + \exp\left(-\frac{\|\mathbf{x}_1' - \mathbf{x}_2'\|^2}{2\sigma^2}\right)$

Experiments

- Obtaining ranks of pairs from data
- $M = C_n^2$, $N = C_M^2 = \frac{n^4}{8} - \frac{n^3}{4} - \frac{n^2}{8} + \frac{n}{4}$

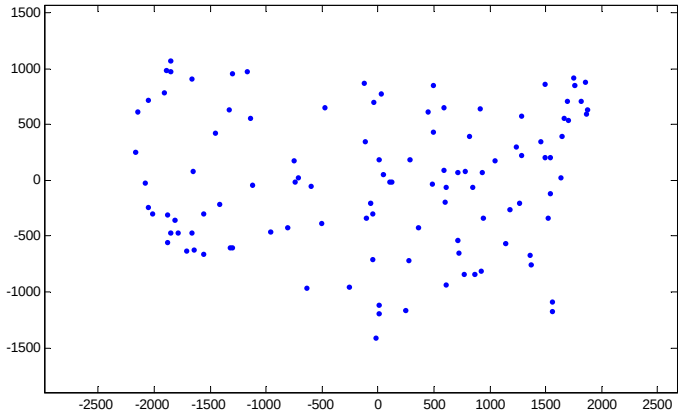
Table: Some examples of N v.s. n.

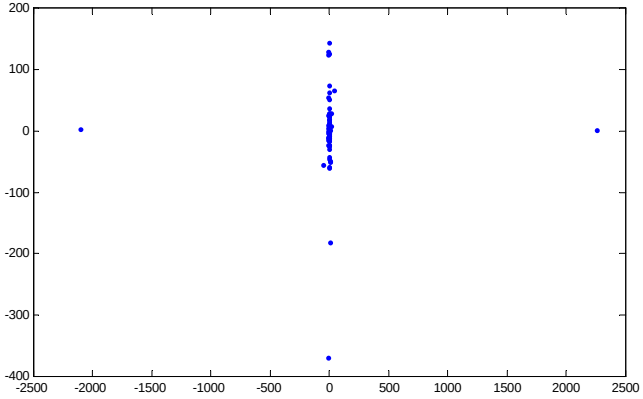
n	2	3	4	10	20	50	100	1000
N	0	3	15	990	17955	749700	12248775	1.2475e+11

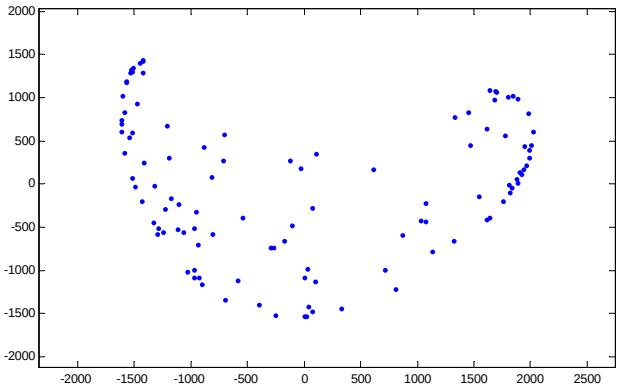
- If $A > B, B > C$, then $A > C$ can be ignored
- $N = \frac{n^2}{2} - \frac{n}{2} - 1$

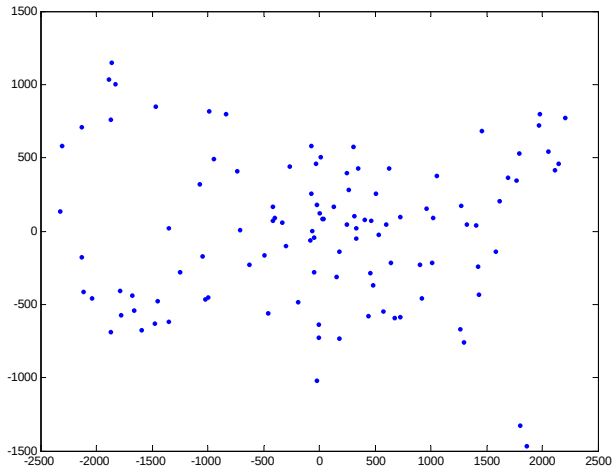
Experiments: 109 US Cities

- 2D locations of 109 US cities. Adjacent diss. considered: 5885 rankings of pairwised distances

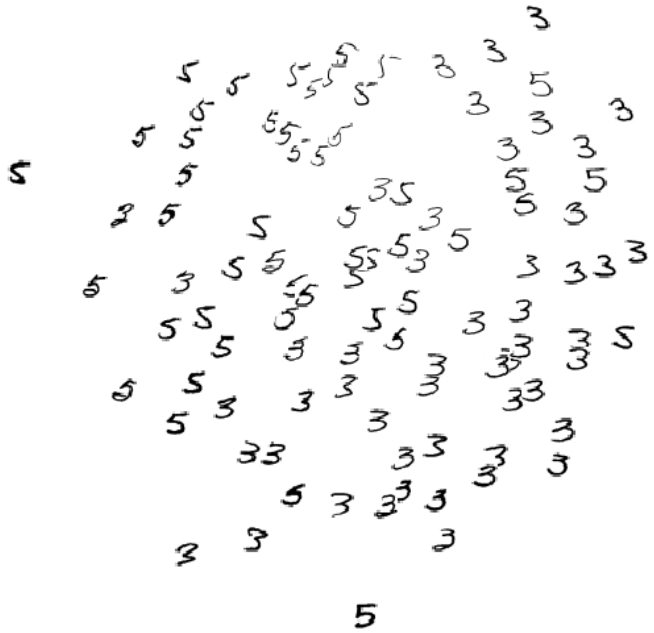


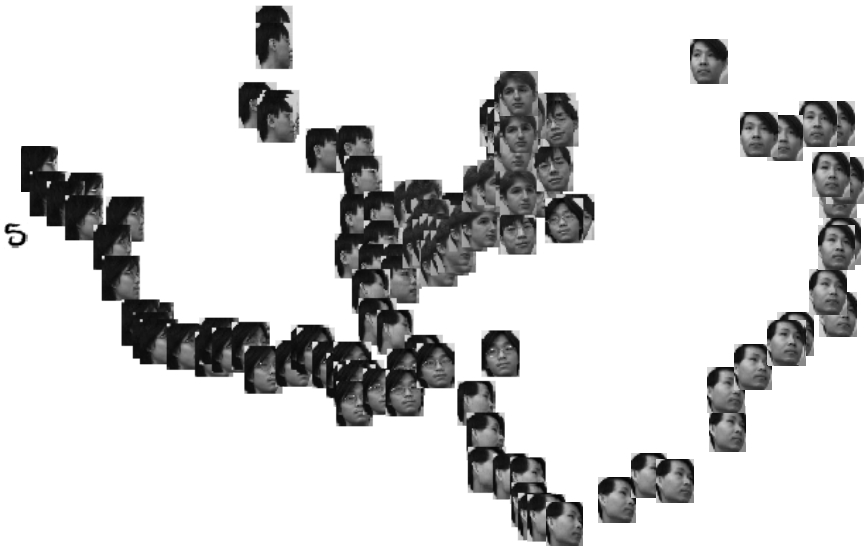






Experiments: USPS Digits and UMist Human Faces





results of d-ranking-VM

Discussions and Conclusions

- Learning ordering can be solved by SDP and QP
- Pros for GNMDS: Can recover low dim. embedding. Only rank information needed. No need values of original samples.
- Cons for GNMDS: Solving SDP is hard, esp. for large scale problems. Using existing SDP solver can only solve $N < 50$ problems. Cannot predict unseen samples.
- Pros for learning orderings by QP: Solving QP is much easier than SDP. Using SMO, can solve $N > 10^3$ problems. Can make prediction for unseen samples.
- Cons for learning orderings by QP: Cannot recover low-dimensional embedding explicitly. Learning dissimilarity measure needs values of original samples. How to choose a good hyperkernel is open problem.

Open Challenges

- The regularization properties of hyperkernels. Better hyperkernels.
- The formulation $F3$ - Input: coefficients in $\mathbb{R}^D$, orderings $d_{ij} \leq d_{kl}$; Output: $f : \mathbb{R}^D \rightarrow \mathbb{R}^L$, coefficients in $\mathbb{R}^L$
- Real-world nonmetric experiments on manifolds.
- Ranking to classification ? Learn optimal dissimilarities for kNN classifier

Reference

- [1] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. Kriegman, and S. Belongie, ”Generalized non-metric multidimensional scaling”, AISTATS 2007
- [2] Ong, C. S., Smola, A. J., Williamson, R. C., ”Learning the Kernel with Hyperkernels”, JMLR 2005
- [3] Kondor, R., Jebara, T., ”Gaussian and Wishart Hyperkernels”, NIPS 2006

Thank you. Q&A.