Metric Learning to Rank

Brian McFee and Gert Lanckriet
University of California, San Diego

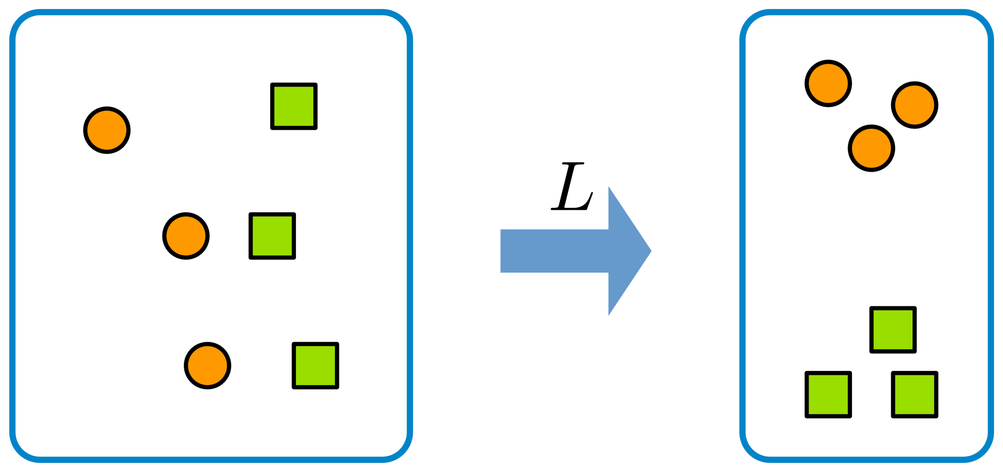
computer audition laboratory

UCSD Jacobs School of Engineering

Overview

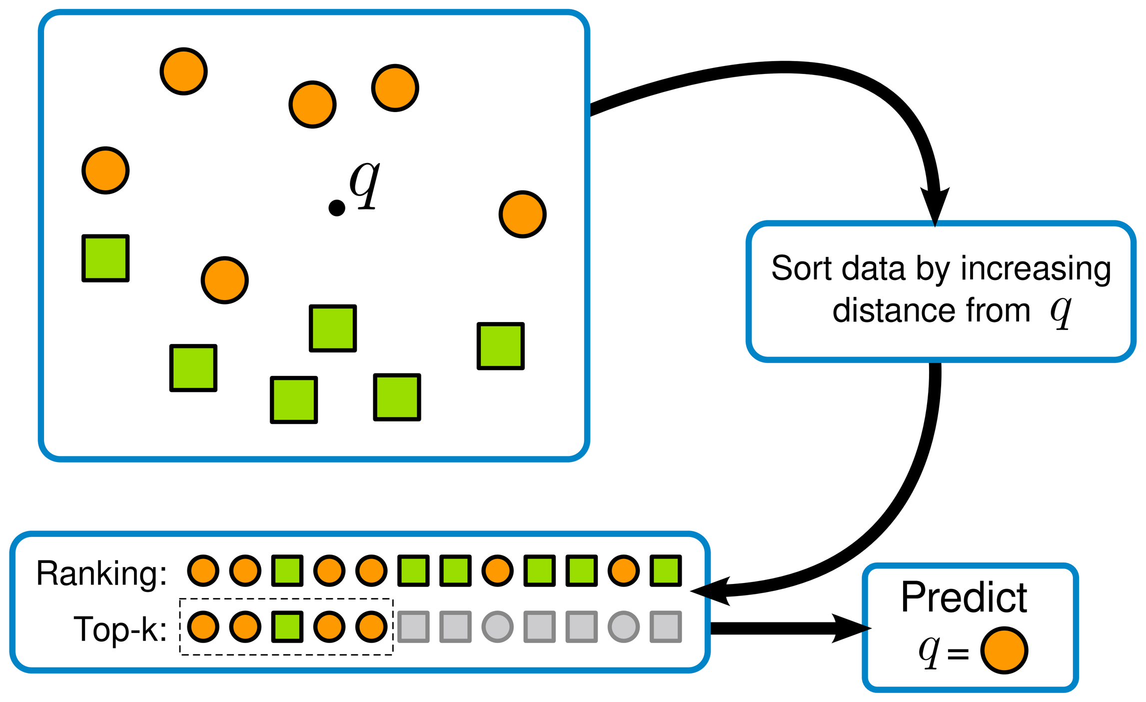
Introduction

In metric learning, the goal is to learn a transformation of the data so that distances conform to similarity, e.g. class labels:



*k*NN error

Learned metrics are typically evaluated by *k*-Nearest Neighbor error. For a query point q, predict its label by kNN:



☆ *k*NN error is a loss function over rankings induced by distance in the learned space.

Metric IR

We formulate metric learning as a learning to rank problem. This leads to an algorithm which can be used for **query-by-example information retrieval** problems.

The algorithm supports general ranking loss measures in addition to binary *k*NN.

Notation

$\mathcal{X} \subset \mathbb{R}^d$ Input: the training set of n points in $\mathbb{R}^d$

$\mathcal{Y}$ Output: the set of permutations over $\mathcal{X}$

y_q^* The true ranking for point q

$\Delta(y_q^*, y)$ The loss incurred by predicting y instead of y_q^*

$W \succeq 0$ The learned (positive semidefinite) metric

$$W = L^\mathsf{T} L$$

$\|a-b\|_W$ The learned distance between a and b

$$\|a-b\|_W^2 = (a-b)^\mathsf{T} W (a-b) = \langle W, (a-b)(a-b)^\mathsf{T}\rangle_\mathrm{F}$$

Algorithm

Structural SVM

The algorithm is based on ranking with Structural SVM[1]:

OPT1

$$\min_w \frac{1}{2}\|w\|^2 + C \cdot \frac{1}{n}\sum_{q\in\mathcal{X}} \xi_q$$

$$\forall q \in \mathcal{X} \quad \forall y \in \mathcal{Y} \setminus \{y_q^*\}$$

$$\langle w, \psi(q, y_q^*)\rangle \geq \langle w, \psi(q, y)\rangle + \Delta(y_q^*, y) - \xi_q$$

Score(good ranking) > Score(bad ranking) + Loss(bad ranking)

$$\psi(q, y) = \sum_{i\in\mathcal{X}_q^+} \sum_{j\in\mathcal{X}_q^-} y_{ij} \frac{\phi(q,i) - \phi(q,j)}{|\mathcal{X}_q^+| \cdot |\mathcal{X}_q^-|}$$

$$y_{ij} = \begin{cases} +1 & i \text{ before } j \\ -1 & i \text{ after } j \end{cases}$$

Rankings are encoded by the *partial order* feature

$\phi(q, i)$ is a feature function for the query/data pair (q, i)

At test time, predict the ranking that maximizes the score. Sort $\mathcal{X}$ in descending order:

$$\max_{y\in\mathcal{Y}} \langle w, \psi(q, y)\rangle \Rightarrow \langle w, \phi(q, i)\rangle \searrow_{i\in\mathcal{X}}$$

Distance ranking

In metric learning, the query is also a data point. We want to sort by increasing distance from the query.

We choose a matrix-valued feature function

$$\phi_M(q, i) = -(q-i)(q-i)^\mathsf{T}$$

and generalize the inner products in OPT1 to Frobenius inner products. For a PSD W, the inner product is the negative distance between q and i:

$$\langle W, -(q-i)(q-i)^\mathsf{T}\rangle_\mathrm{F} = -\|q-i\|_W^2$$

Max-score prediction: sort by increasing distance from the query.

This leads to the Metric Learning to Rank (MLR) optimization:

OPT2

$$\min_{W\succeq 0} \operatorname{tr}(W) + C \cdot \frac{1}{n}\sum_{q\in\mathcal{X}} \xi_q$$

$$\forall q \in \mathcal{X} \quad \forall y \in \mathcal{Y} \setminus \{y_q^*\}$$

$$\langle W, \psi(q, y_q^*)\rangle_\mathrm{F} \geq \langle W, \psi(q, y)\rangle_\mathrm{F} + \Delta(y_q^*, y) - \xi_q$$

Optimization

OPT2 has exponentially many constraints. We find an approximate solution to OPT2 by adapting the 1-Slack margin-rescaling cutting plane algorithm [2].

Experiments

Ranking loss

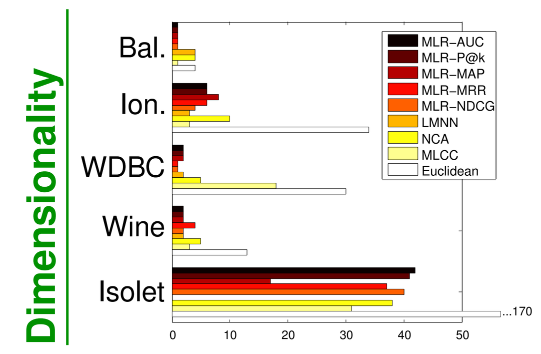
Our experiments test five different choices for $\Delta(y_q^*, y)$:

AUC	Area under the ROC curve
P@k	Precision-at-*k*
MAP	Mean Average Precision
MRR	Mean Reciprocal Rank
NDCG	Normalized Discounted Cumulative Gain

Classification: UCI

The first experiment tests nearest-neighbor classification on standard data sets. Relevance is derived from label agreement between points.

We compare to kNN using the native metric (Euclidean), LMNN, NCA and MLCC.

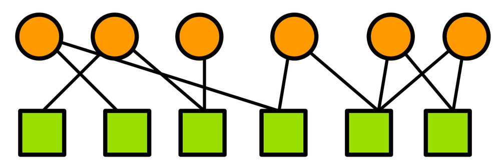


kNN error rate

Algorithm	Bal.	Ion.	WDBC	Wine	Isolet
MLR-AUC	7.9	12.3	2.7	1.4	4.5
MLR-P@k	8.2	12.3	2.9	1.5	4.5
MLR-MAP	6.9	12.3	2.6	1.0	5.5
MLR-MRR	8.2	12.1	2.6	1.5	4.5
MLR-NDCG	8.2	11.9	2.9	1.6	4.4
LMNN	8.8	11.7	2.4	1.7	4.7
NCA	4.6	11.7	2.6	2.7	10.8
MLCC	5.5	12.6	2.1	1.1	4.4
Euclidean	10.3	15.3	3.1	3.1	8.1

IR: eHarmony

The second experiment tests information retrieval on a large-scale set of matching data provided by eHarmony, Inc.



Users of the system can be both queries and results. A pair of users are mutually relevant if the match was succesful.

Each user is represented as a vector in $\mathbb{R}^{56}$.

Data is given for two equal length intervals, corresponding to training and test sets.

Data

	Matchings	Users	Queries
Train	506,688	294,832	22,391
Test	439,161	247,430	36,037

Results

Algorithm	AUC	MAP	MRR	Rounds
MLR-AUC	0.612	0.445	0.466	7
MLR-MAP	**0.624**	**0.453**	**0.474**	23
MLR-MRR	0.616	0.448	0.469	17
SVM-MAP	0.614	0.447	0.467	36
Euclidean	0.522	0.394	0.414	

References

[1] Tsochantaridis, Ioannis, Joachims, Thorsten, Hofmann, Thomas, and Altun, Yasemin. Large margin methods for structured and interdependent output variables. *JMLR*, 6: 1453-1484, 2005.

[2] Joachims, Thorsten, Finley, Thomas, and Yu, Chun-nam John. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27-59, 2009.