电子科技大学
University of Electronic Science and Technology of China

A Further Talk about Metric Learning

Liu

Data Mining Lab, Big Data Research Center, UESTC
Email：lsl571@163.com

What exactly metric learning is

Euclidean ditance

$$dist_{ed}^2(x_i, x_j) = dist_{ij,1}^2 + dist_{ij,2}^2 + \ldots + dist_{ij,d}^2$$

Attribute weight

$$dist_{ed}^2(x_i, x_j) = w_1 * dist_{ij,1}^2 + w_2 * dist_{ij,2}^2 + \ldots + w_d * dist_{ij,d}^2$$

$$= (x_i - x_j)^T \boxed{W} (x_i - x_j)$$

$$\begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & w_d \end{pmatrix}$$

What if attributes are related? e.g. students' height and weight

Matrix W is replaced by a semi-definite matrix M

If rank(M) is less than d,

$$M = P P^T$$

$$P \in R^{d \times rank(M)}$$

➢ Primary challenges

- maintain M which is a semi-definite in an efficient way during theoptimization process.

- learn a low-rank matrix

Framework

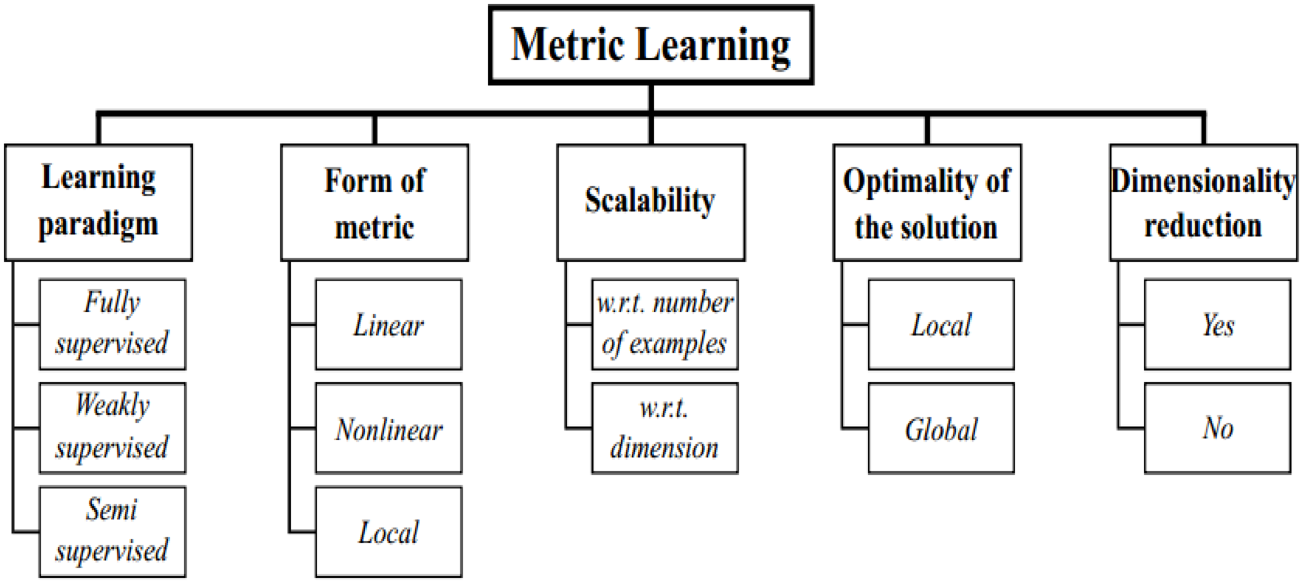


Figure 3: Five key properties of metric learning algorithms.

Xing et al. (2002)

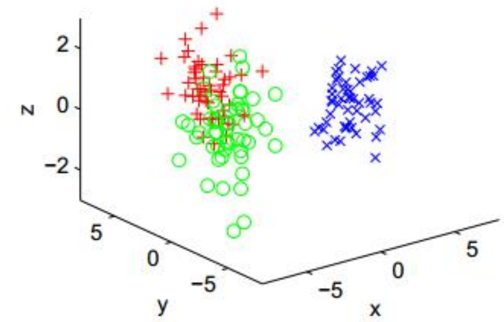
数据挖掘实验室
Data Mining Lab

The originator

$$S = \{(x_i, x_j) : x_i \text{ and } x_j \text{ should be similar}\}$$

$$D = \{(x_i, x_j) : x_i \text{ and } x_j \text{ should be dissimilar}\}$$

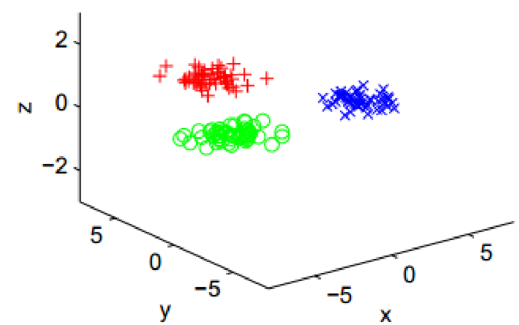
side information



(a)

$$\max_{M \in S_+^d} \sum_{(x_i, x_j) \in D} d_M(x_i, x_j)$$

$$s.t. \sum_{(x_i, x_j) \in S} d_M^2(x_i, x_j) \leq 1$$



(b)

LMNN

Weinberger et al. (2009)

➢Goal

- k-nearest neighbors always belong to the same class while examples from different classes are separated by a large margin.

Target neighbors

k other neighbors with the same labels.

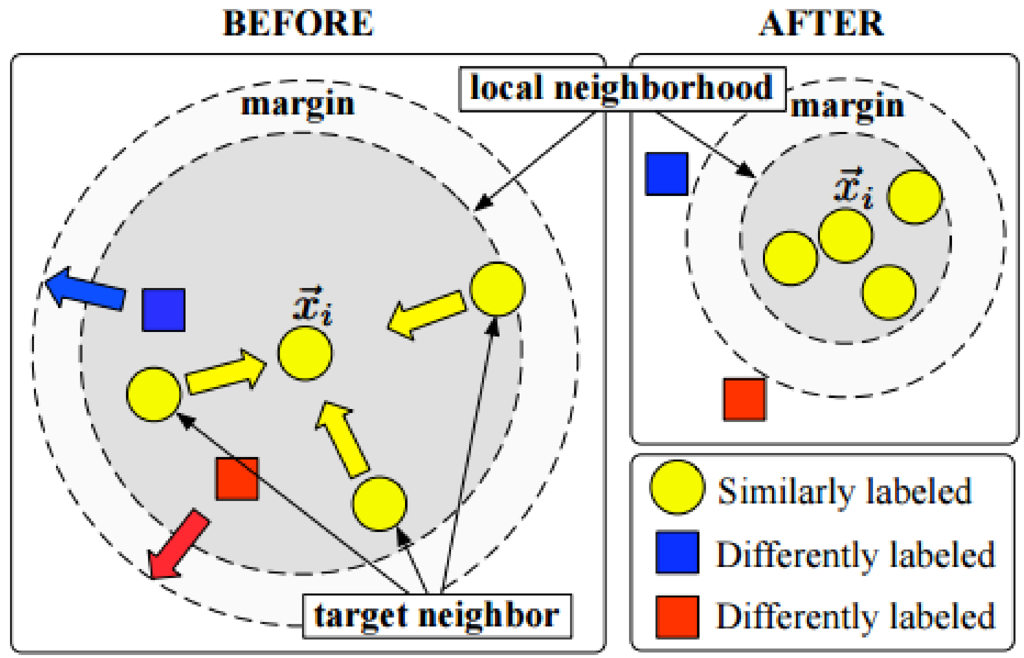
We use $y_{ij} \in \{0,\ 1\}$ to denote whether two labels are same or not.

We use $\eta_{ij} \in \{0,\ 1\}$ to denote whether two instances are target neighbors or not.

Cost function

The first part -- $\sum_{ij}\eta_{ij}\,\| L($

The second part --$\sum_{ijl}\eta_{ij}(1-$



$$\varepsilon(L)=\sum_{ij}\eta_{ij}\,\| L(\vec{x}_i-\vec{x}_j)\|^2+c\sum_{ijl}\eta_{ij}(1-y_{il})[1+\| L(\vec{x}_i-\vec{x}_j)\|^2-\| L(\vec{x}_i-\vec{x}_l)\|^2]_+$$

Shown in metric learning way and import slack variables

$$\min \sum_{ij} (x_i - \vec{x}_j)^T M (x_i - \vec{x}_j) + c \sum_{ij} \eta_{ij} (1 - y_{il}) \xi_{ijl}$$

$$s.t.$$

$$(x_i - \vec{x}_l)^T M (x_i - \vec{x}_l) - (x_i - \vec{x}_j)^T M (x_i - \vec{x}_j) \geq 1 - \xi_{ijl}$$

$$\xi_{ijl} \geq 0$$

$$M \geq 0$$

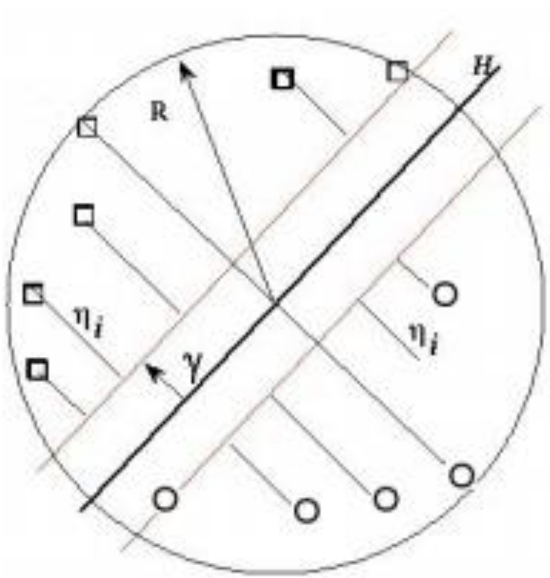
➢ ε -SVM

A framework of combining
both SVM and ML

$$R^2 / \gamma^2$$

smallest sphere

maximum margin



Notice that

$$\left(w^T\left(x_i - x_j\right)\right)^2 = \left(x_i - x_j\right)^T w w^T \left(x_i - x_j\right)$$

- ε-SVM

$$\min_{w,b}\ \mathrm{w}^{\mathrm{T}}w\ +\lambda\sum_i \max(0, y_i(\mathrm{w}^{\mathrm{T}}x_i+b)-1)$$

$$+C\sum_i \max(0,1-y_i(\mathrm{w}^{\mathrm{T}}x_i+b))$$

We can classify instances and get a good distance function meanwhile.

ITML

Davis et al. (2007)

➢Goal

- Minimizing the differential relative entropy between two multivariate Gaussians under constraints to get a proper distance function.

We use covariance matrix as target

Constraints in this paper:

$$d_A(x_i, x_j) \leq u$$

Similar points

$$d_A(x_i, x_j) \geq l$$

Dissimilar points

Now we need a tool to measure the difference of two distance functions(i.e. Mahalanobis matrix)

For each point $$p(x;A)=\frac{1}{Z}\exp(-\frac{1}{2}d_A(x,\mu))$$

The tool—
relative entropy between their corresponding multivariate Gaussians which have equal mean:

$$KL(p(x;A_0)\,||\,p(x;A)) = \int p(x;A_0)log\frac{p(x;A_0)}{p(x;A)}dx$$

Cost function

$$\min_{A} KL(p(x;A_0)\,\|\,p(x;A))$$

$$\text{s.t.}\quad \boxed{\begin{array}{ll} d_A(x_i,x_j)\le u & i,j\in S \\ d_A(x_i,x_j)\ge l & i,j\in D \end{array}}$$

can be replaced by arbitrary linear constraints

As a convex optimization problem (Davis & Dhillon, 2006)

The LogDet divergence

$$D_{\ell d}(A,A_0) \;=\; tr(AA_0^{-1}) - log\;det(AA_0^{-1}) - n$$

$$KL(p(x;A_0)\,||\,p(x;A))$$
$$=\ \frac{1}{2}D_{\ell d}(A_0^{-1},A^{-1})$$
$$=\ \frac{1}{2}D_{\ell d}(A,A_0)$$

Import slack variables

$$\min_{A\geq 0,\xi} D_{\ell d}(A,A_0)\ +\ \gamma D_{\ell d}(diag(\xi),diag(\xi_0))$$
$$s.t.\quad tr(A(x_i-x_j)(x_i-x_j)^T)\leq \xi_{c(i,j)}\quad (i,j)\in S$$
$$tr(A(x_i-x_j)(x_i-x_j)^T)\geq \xi_{c(i,j)}\quad (i,j)\in D$$

where c(i,j) denotes the index of constraints

Algorithm 1 Information-theoretic metric learning

Input: X: input $d \times n$ matrix, S: set of similar pairs
D: set of dissimilar pairs, u, ℓ: distance thresholds
A_0: input Mahalanobis matrix, γ: slack parameter, c: constraint index function

Output: A: output Mahalanobis matrix

1. $A \leftarrow A_0$, $\lambda_{ij} \leftarrow 0 \ \forall\, i, j$
2. $\xi_{c(i,j)} \leftarrow u$ for $(i, j) \in S$; otherwise $\xi_{c(i,j)} \leftarrow \ell$
3. **repeat**
   3.1. Pick a constraint $(i, j) \in S$ or $(i, j) \in D$
   3.2. $p \leftarrow (\boldsymbol{x}_i - \boldsymbol{x}_j)^T A(\boldsymbol{x}_i - \boldsymbol{x}_j)$
   3.3. $\delta \leftarrow 1$ if $(i, j) \in S$, -1 otherwise
   3.4. $\alpha \leftarrow \min\left(\lambda_{ij}, \frac{\delta}{2}\left(\frac{1}{p} - \frac{\gamma}{\xi_{c(i,j)}}\right)\right)$
   3.5. $\beta \leftarrow \delta\alpha/(1 - \delta\alpha p)$
   3.6. $\xi_{c(i,j)} \leftarrow \gamma\xi_{c(i,j)}/(\gamma + \delta\alpha\xi_{c(i,j)})$
   3.7. $\lambda_{ij} \leftarrow \lambda_{ij} - \alpha$
   3.8. $A \leftarrow A + \beta A(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T A$
4. **until** convergence

return A

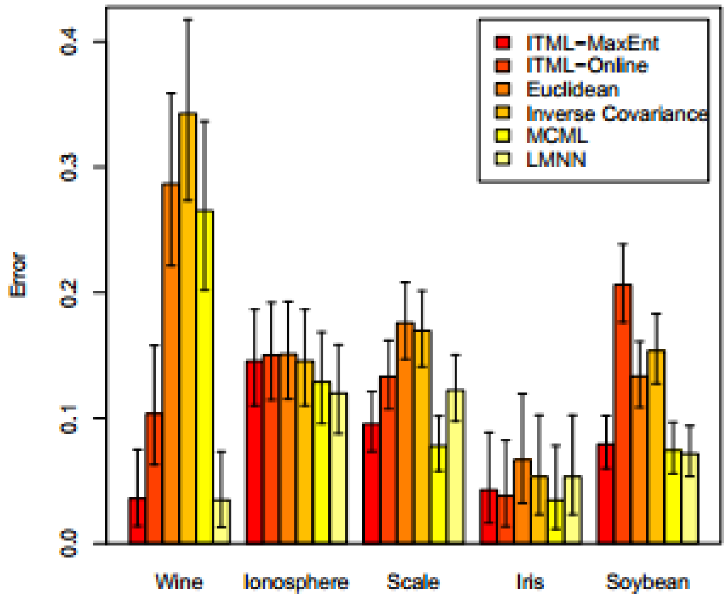
Advantages

- a wide variety of constraints and can optionally incorporate a prior on the distance function
- no eigenvalue computations or semi-definite programming are required

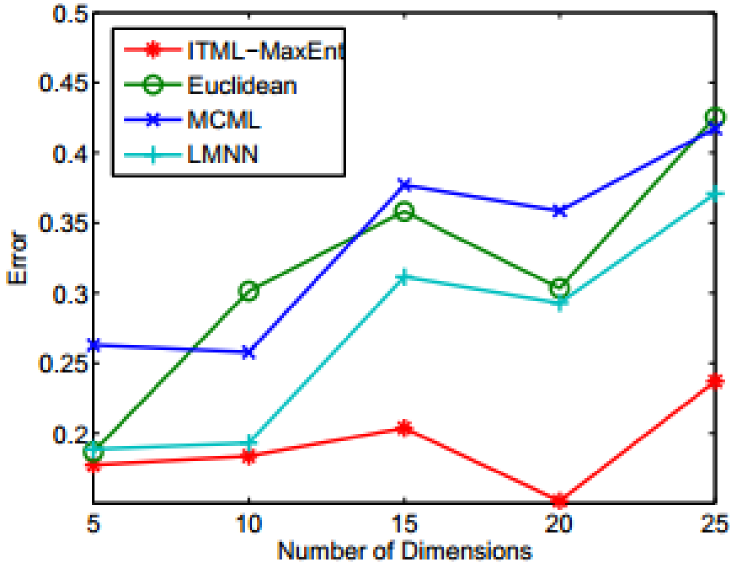
Table 1. Training time (in seconds) for the results presented in Figure 1(b).

Dataset	ITML-MaxEnt	MCML	LMNN
Latex	**0.0517**	19.8	0.538
Mpg321	**0.0808**	0.460	0.253
Foxpro	**0.0793**	0.152	0.189
Iptables	0.149	**0.0838**	4.19

Results



(a) UCI Datasets



(c) Latex

ISD

Zhan et al.(2009)

- Goal
  - propagating and adapting metrics of individual labeled examples to individual unlabeled instances.

Label propagation

A **graph** defined over both labeled and unlabeled instances is provided, and the labels are then propagated from labeled instances to unlabeled ones across the graph.

Assumption

Similar instances share similar properties, the distribution of the instance specific distance functions should be smooth within a local area.

Cost function

$$\min_{W} \lambda \sum_{i=1}^{n} \sum_{j \in S_i} l(\hat{y}_{ij}, D_i(x_j)) + \Omega(W, G)$$

$$s.t.\ \ w_i \geq 0, i = 1, \ldots, n+u$$

where $$D_i(x_j) = w_i^T (x_i - x_j)^T (x_i - x_j)$$

S_i *is similar set of* x_i

$l()$ *is a loss function*

$\hat{y}_{ij} \in \{-1, 1\}$

Ω *is where metric propagation works*

ISD with L1-loss

$$l(\hat{y}_{ij}, D_i(x_j)) = \max(0, \hat{y}_{ij}(D_i(x_j) - \eta))$$

ISD with L2-loss

$$l(\hat{y}_{ij}, D_i(x_j)) = \max(0, \hat{y}_{ij}(D_i(x_j) - \eta))^2$$

The metric propagation

$$\Omega(W,G)=\sum_{i,j=1}^{n+u}E_{ij}\,\|\,w_i-w_j\,\|^2$$

where

$$\mathbf{E}=\mathbf{U}^{\frac{-1}{2}}\mathbf{G}\mathbf{U}^{\frac{-1}{2}}$$

If weight(i.e. E_{ij}) is bigger,two points are more similar,their Distance function must be more similar.Otherwise is the same.

Thanks