Distance based Kernel for First-Order Logic Data

Nirattaya Khamsemanan[1], Cholwich Nattee[1], and Masayuki Numao[2]

[1]Sirindhorn International Institute of Technology, Thammasat University
[2]The Institute of Scientific and Industrial Research, Osaka University

ILP 2015

First-Order Logic objects

- An object can be represented as a set of predicates.
- A predicate describes the object's properties, components, relationships among components, etc.
- Names or IDs are typically used for the identification.

```
nitro(d1,d1_19,d1_24,d1_25,d1_26).
bond(d1,d1_1,d1_2,7).
bond(d1,d1_2,d1_3,7).
bond(d1,d1_3,d1_4,7).
bond(d1,d1_4,d1_5,7).
atm(d1_1,c,22,-0.117).
atm(d1_2,c,22,-0.117).
atm(d1_3,c,22,-0.117).
atm(d1_4,c,195,-0.087).
atm(d1_5,c,195,0.013).
```

Setting for FOL Objects

In our setting, every predicate is written in form of

$$r(ID, x_1, x_2, \ldots, x_n)$$

where

- r is a predicate symbol,
- ID is an object described by the predicate,
- x_i is a property value.

Distance between FOL objects

d1 ⟵ *distance* ⟶ **e25**

```
nitro(d1,d1_19,d1_24,
      d1_25,d1_26).
bond(d1,d1_1,d1_2,7).
bond(d1,d1_2,d1_3,7).
bond(d1,d1_3,d1_4,7).
bond(d1,d1_4,d1_5,7).
atm(d1_1,c,22,-0.117).
atm(d1_2,c,22,-0.117).
atm(d1_3,c,22,-0.117).
atm(d1_4,c,195,-0.087).
atm(d1_5,c,195,0.013).
```

```
nitro(e25,e25_7,e25_21,
      e25_22,e25_23).
bond(e25,e25_1,e25_2,1).
bond(e25,e25_2,e25_3,1).
bond(e25,e25_3,e25_4,7).
bond(e25,e25_4,e25_5,1).
atm(e25_1,c,10,-0.083).
atm(e25_2,c,10,0.017).
atm(e25_3,c,22,-0.113).
atm(e25_4,c,22,0.017).
atm(e25_5,c,10,-0.083).
```

Distance Functions for FOL objects

A number of distance functions have been proposed to measure dissimilarity between two FOL objects:

- FOL Similarity (Bisson, 1992)
- RIBL (Emde and Wettschereck, 1996)
- RB distance (Ramon and Bruynooghe, 2001)
- Kernels and distances for structured data (Gärtner et al., 2004)
- DISTALL (Tobudic and Widmer, 2006)

Four-layer Distance Function

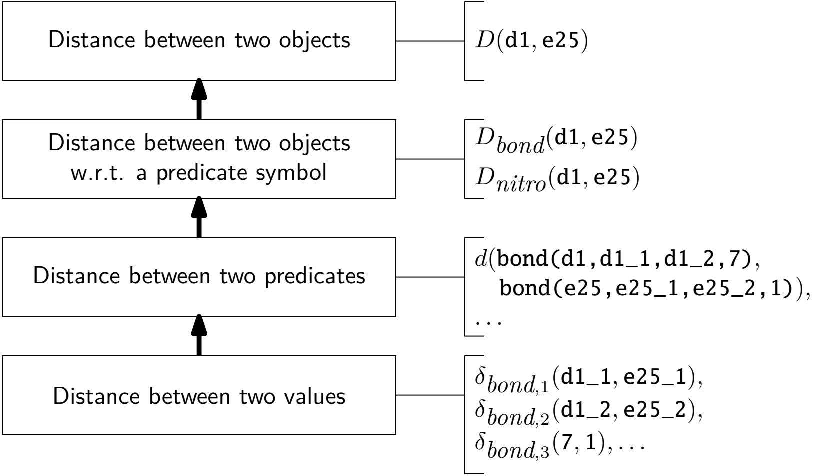
We have proposed a distance function for FOL objects, called four-layer distance function.

This function satisfies the metric properties:

- $d(x, y) = 0$ if and only if $x = y$ (coincidence axiom)
- $d(x, y) = d(y, x)$ (symmetry)
- $d(x, z) + d(z, y) \geq d(x, y), \forall z \in X$ (triangular inequality)

The properties that the function preserves the closeness of the objects.

Four-layer Distance Function



https://bitbucket.org/fol_dist/fol4l_distance

Four-layer Distance Function

Distance between two objects

$$D(X,Y) = \sqrt{\frac{\sum_{r\in\Omega} (D_r(X,Y))^2}{|\Omega|}}$$

Distance between two objects w.r.t. a predicate symbol

$$D_r(X,Y) = \begin{cases} \begin{array}{l} \max\{\max_{k=1}^{p} \min_{j=1}^{q} d_r(X^{r_k}, Y^{r_j}), \\ \quad \max_{j=1}^{q} \min_{k=1}^{p} d_r(X^{r_k}, Y^{r_j})\} \end{array} & \text{if } p,q \neq 0 \\ 1 & \begin{array}{l} \text{if } p \neq 0, q = 0, \\ \text{or } p = 0, q \neq 0 \end{array} \\ 0 & \text{if } p = q = 0 \end{cases}$$

Four-layer Distance Function

Distance between two predicates

$$d_r(X^r, Y^r) = \sqrt{\frac{\sum_{i=1}^{n} (\delta_{r,i}(x_i, y_i))^2}{n}}$$

where,

$$\delta_{r,i}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i, \\ \Delta_{r,i}(x_i, y_i) & \text{if at most one of } x_i, y_i \text{ is an ID,} \\ D(x_i, y_i) & \text{if both } x_i, y_i \text{ are ID's.} \end{cases}$$

Four-layer Distance Function

Distance between two values

$$\Delta_{r,i}(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i, \\ 1 & \text{if } x_i \neq y_i, \text{ and } x_i \notin \mathbb{R} \text{ or } y_i \notin \mathbb{R}, \\ \dfrac{|x_i - y_i|}{\max(r, i)} & \text{if } x_i \neq y_i \text{ and } x_i, y_i \in \mathbb{R}. \end{cases}$$

Four-layer Distance based Kernels

A kernel $k : \mathcal{C} \times \mathcal{C} \to \mathbb{R}$ is a real-valued function. If k is positive definite, then there is a map Φ that isometically embeds $\mathcal{C}$ into a Hilbert space $\mathcal{H}$

$$\langle \Phi(X), \Phi(Y) \rangle = k(X, Y)$$

A distance based kernel is a kernel created based on a distance metric d such that $d(X, Y) = \|\Phi(X) - \Phi(Y)\|_{\mathcal{H}}$, it can be defined as

$$k(X, Y) = k_O(X, Y) = \frac{1}{2} \left(d(X, Y)^2 - d(X, O)^2 - d(Y, O)^2 \right)$$

where O is a fixed object in a dataset C.

Four-layer Distance based Kernels

We can define a number of kernels (Haasdonk and Bahlmann, 2004):

1. Simple linear kernel

$$k_d^{lin} = \langle x, x' \rangle_d^O$$

2. Negative distance kernel

$$k_d^{nd} = -d(x, x')^\beta, \beta \in [0, 2]$$

3. Polynomial kernel

$$k_d^{pol} = \left(1 + \gamma \langle x, x' \rangle_d^O\right)^p, p \in \mathbb{N}, \gamma \in \mathbb{R}$$

4. Gaussian kernel

$$k_d^{rbf} = e^{\gamma d(x,x')^2}, \gamma \in \mathbb{R}$$

Positive Definiteness

A kernel function k is positive definite, if and only if its Gram matrix $\mathbf{K} = [k(X^i, X^j)]$ where $i, j = 1, \ldots, |\mathcal{C}|$ is positive-semidefinite.

The positive definite property of k is required to secure the maximal margin in the Hilbert space $\mathcal{H}$.

However, our four-layer distance based kernel are not positive definite on some datasets. We apply the shift spectrum transformation (Wu et al., 2005) on the indefinite Gram metrix to obtain the positive semidefinite one.

$$\widetilde{K} = U\widetilde{\Lambda}U^T = U(\Lambda + \eta I)U^T = K + \eta I$$

Experiment

- Datasets
  Mutagenesis, Alzheimers
- Techniques
  1. Aleph,
  2. k-NN using 4L distance,
  3. SVM using 4 distance based kernels
     with 4L and RB distance functions
  4. SVM using Structured data kernel
- Shift spectrum transformation is applied to indefinite matrices with $\eta = |\lambda_N|$, where λ_N is the smallest eigenvalue.

Experimental Results

Method	Muta	Alz amine	Alz toxic	Alz acetyl	Alz memory
Aleph	73.4 ± 11.8	$70.2 \pm 7.3^*$	$90.9 \pm 3.5^*$	$73.5 \pm 4.3^*$	$69.3 \pm 3.9^*$
k-NN	$\mathbf{92.0 \pm 8.2}$	$94.2 \pm 0.4^*$	$94.7 \pm 0.3^*$	$89.1 \pm 0.4^*$	$87.3 \pm 0.4^*$
k_{4L}^{lin}	70.2 ± 12.4	$92.1 \pm 4.5^*$	$94.4 \pm 2.1^*$	$93.0 \pm 1.6^*$	87.4 ± 5.6
k_{4L}^{nd}	72.8 ± 8.1	$93.3 \pm 5.3^*$	98.0 ± 1.6	$93.0 \pm 2.9^*$	88.5 ± 3.8
k_{4L}^{pol}	79.7 ± 8.1	$\mathbf{96.4 \pm 2.7}^\dagger$	$\mathbf{98.2 \pm 1.1}$	$\mathbf{95.6 \pm 2.0}$	$\mathbf{89.6 \pm 3.9}$
k_{4L}^{gs}	74.0 ± 9.3	$93.0 \pm 4.6^{\dagger *}$	$95.5 \pm 2.2^*$	$92.8 \pm 2.0^*$	88.3 ± 4.8
k_{RB}^{lin}	82.4 ± 6.3	$72.6 \pm 4.3^*$	$62.7 \pm 4.4^*$	$62.7 \pm 3.0^*$	$52.5 \pm 8.4^*$
k_{RB}^{nd}	81.9 ± 8.3	$70.8 \pm 5.7^*$	$59.9 \pm 5.3^*$	$62.0 \pm 3.8^*$	$52.2 \pm 8.3^*$
k_{RB}^{pol}	77.6 ± 5.2	$72.5 \pm 4.8^*$	$73.3 \pm 5.1^*$	$69.8 \pm 4.1^*$	$59.5 \pm 3.6^*$
k_{RB}^{gs}	83.6 ± 9.7	$85.1 \pm 3.8^*$	$88.5 \pm 2.6^*$	$81.3 \pm 3.0^*$	$74.3 \pm 3.0^*$
k_{SK}	82.0 ± 10.8	$93.2 \pm 3.5^*$	$96.4 \pm 1.8^*$	94.6 ± 2.3	88.8 ± 3.7

Conclusion

- We propose kernel functions based on distance function on FOL objects.
- Then, SVM can be used to learn classifiers from FOL datasets.
- The obtained classifiers outperform the classifiers from the existing kernel functions and techniques.

Thank you very much
for your time.