SUPERVISED FEATURE EMBEDDING FOR CLASSIFICATION BY LEARNING RANK-BASED NEIGHBORHOODS

Ghazaal Sheikhi, Hakan Altıncay

Department of Computer Engineering

Eastern Mediterranean University

Famagusta, North Cyprus, Mersin 10, Turkey

Abstract

- Proposed feature embedding method, representative learning of rank-based neighborhoods (RLRN) is inspired by the well-known word embedding technique, word2vec.
- The notion of context words in word2vec is extended into neighboring instances.
- Each sample is represented by a unique one-hot vector whereas its neighbors are encoded by several two-hot vectors.
- A feed-forward neural network with a continuous projection layer, learns the mapping from one-hot vectors to multiple two-hot vectors.
- The hidden layer determines the reduced subspace for the train samples.
- Experimental results confirm that the proposed method is effective in finding a discriminative representation of the features.

An example of neighborhood encoding in RLRN

	f_1	f_2	f_3	Label
x_1	0.2	3	121	0
x_2	0.4	4	11	0
$\boldsymbol{x_3}$	**0.5**	**1**	**10**	**0**
x_4	0.3	0	40	1
x_5	0.1	5	26	1
x_4	0.7	8	32	1

Ranks		
2	3	6
4	4	2
5	**2**	**1**
3	1	5
1	5	3
6	6	4

$\boldsymbol{\alpha}$	(0, 0, 1, 0, 0, 0)

$\boldsymbol{t^1}$	(1, 0, 1, 0, 0, 0)
$\boldsymbol{t^2}$	(0, 1, 1, 0, 0, 0)
$\boldsymbol{t^3}$	(0, 1, 1, 0, 0, 0)
$\boldsymbol{t^4}$	(0, 1, 1, 0, 0, 0)

Neighborship window size:
$w = 2^*$

Rank of x_3 in f_1 dimension is 5:
Samples with ranks 3,4,5,6,7 are x_3's neighbors

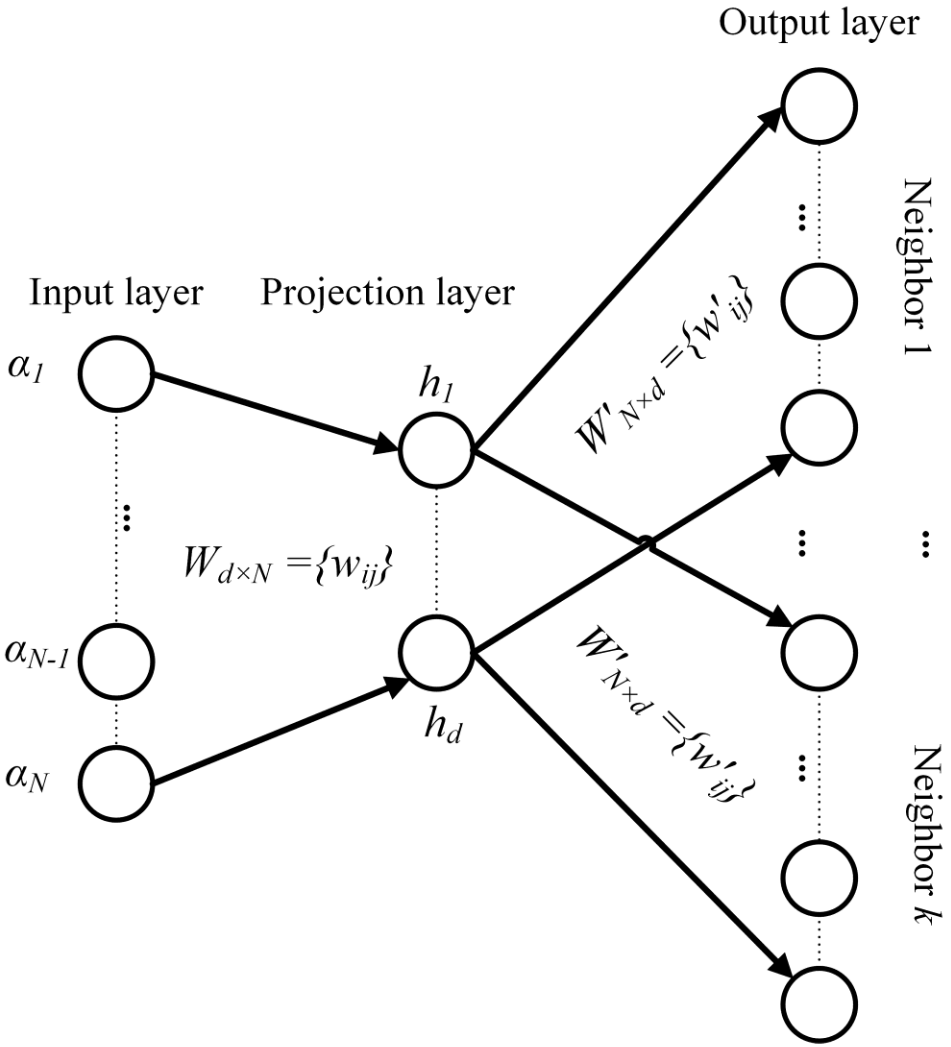
Rank of x_3 in f_2 dimension is 2:
Samples with ranks 1,2,3,4 are x_3's neighbors

Rank of x_3 in f_3 dimension is 1:
Samples with ranks 1,2,3 are x_3's neighbors

***w=2 means if the rank difference is less than or equal to 2, the samples are assumed neighbors given that they belong to the same class.**

The neural network architecture of RLRN

One-hot encoded samples



Two-hot encoded neighbors

The neural network architecture of RLRN

The weight matrix between the input and the hidden layer

$$\boldsymbol{h} = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_d \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{d1} & w_{d2} & \dots & w_{dN} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}$$

α is a one-hot vector with its α_{n^*} equal to 1 and its other elements are 0.

The matrix multiplication is thus unnecessary:

$$\boldsymbol{h} = \begin{bmatrix} w_{1n^*} \\ w_{2n^*} \\ \vdots \\ w_{dn^*} \end{bmatrix} = \boldsymbol{w_{n^*}}$$

The neural network architecture of RLRN

The weight matrix between the hidden layer and the output layer

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} w'_{11} & w'_{12} & \dots & w'_{1N} \\ w'_{21} & w'_{22} & \dots & w'_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w'_{d1} & w'_{d2} & \dots & w'_{dN} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} \Longrightarrow \boldsymbol{u} = W'.\boldsymbol{w_{n^*}}$$

The elements of the output vector

$$y_i = \frac{1}{1 + e^{-u_i}}$$

Updating weights

The error on the *i*th output node of the *k*th neighbor

$$e_i^{\kappa} = -t_i^{\kappa}.log(y_i)$$

The sum operator only takes values for the two-hot elements of $t_i{}^k$ i.e. $\boldsymbol{t_i{}^{k^*}}$

$$e_i = -\sum_{\kappa \in \kappa^*} t_i^{\kappa}.(u_i - log(1 + e^{u_i}))$$

The update formulas for W' and W

$$w'_{ij}(t+1) = w'_{ij}(t) + \eta. \sum_{\kappa \in \kappa^*} t_i^{\kappa}.(1 - y_i).h_j$$

$$w_{ij}(t+1) = w_{ij}(t) + \eta. \sum_{\kappa \in \kappa^*} t_i^{\kappa}.(1 - y_i).w'_{ij}.\alpha_i$$

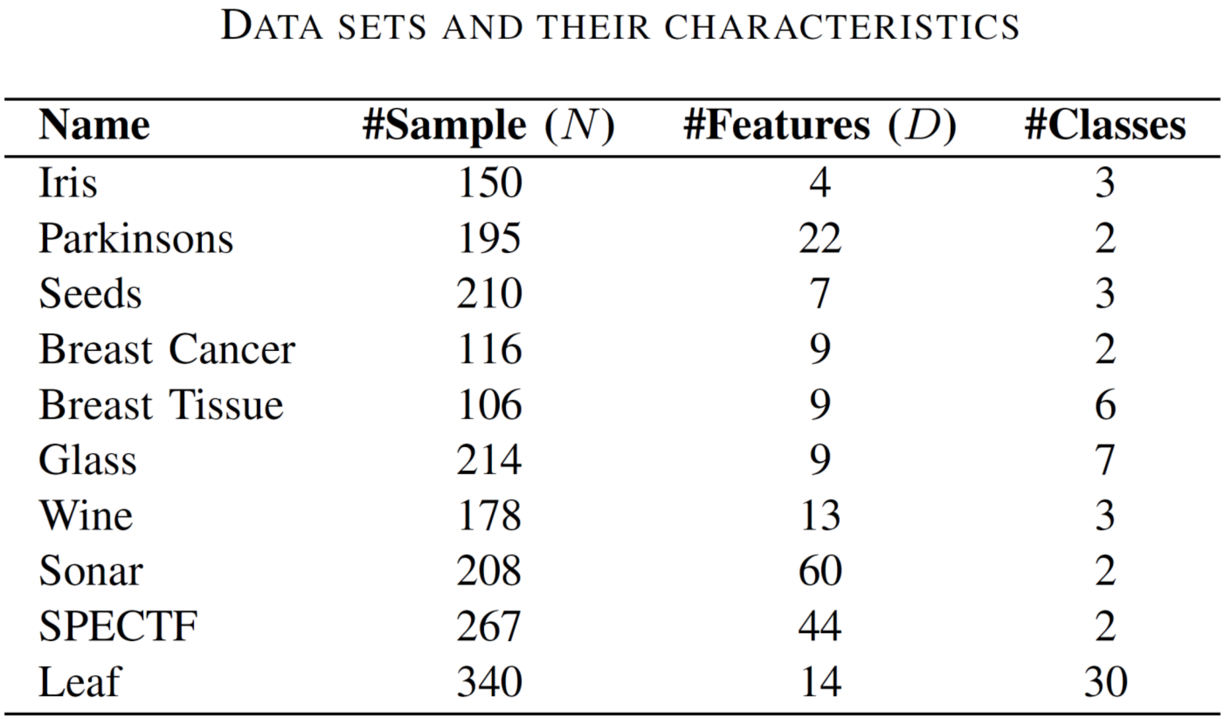
Experiments

10-fold cross validation

SVM and kNN

Accuracy and AUC

Target dimension $d = \{2,3,\dots,\min(10,D)\}$



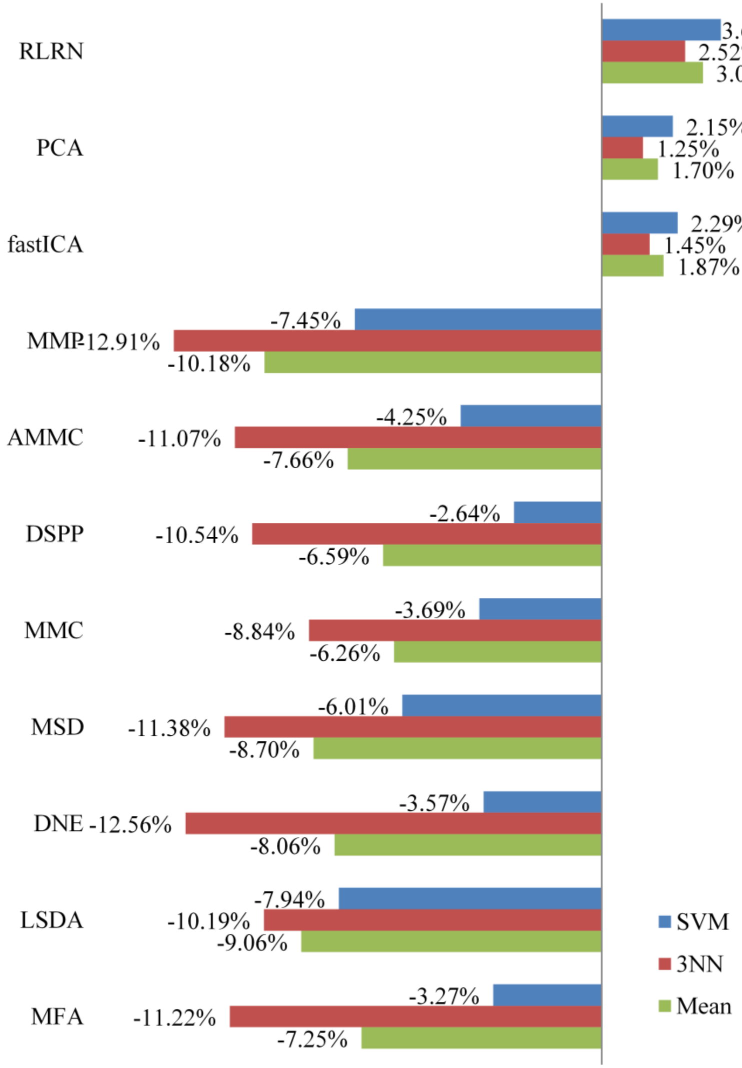
DATA SETS AND THEIR CHARACTERISTICS

Name	**#Sample** (N)	**#Features** (D)	**#Classes**
Iris	150	4	3
Parkinsons	195	22	2
Seeds	210	7	3
Breast Cancer	116	9	2
Breast Tissue	106	9	6
Glass	214	9	7
Wine	178	13	3
Sonar	208	60	2
SPECTF	267	44	2
Leaf	340	14	30

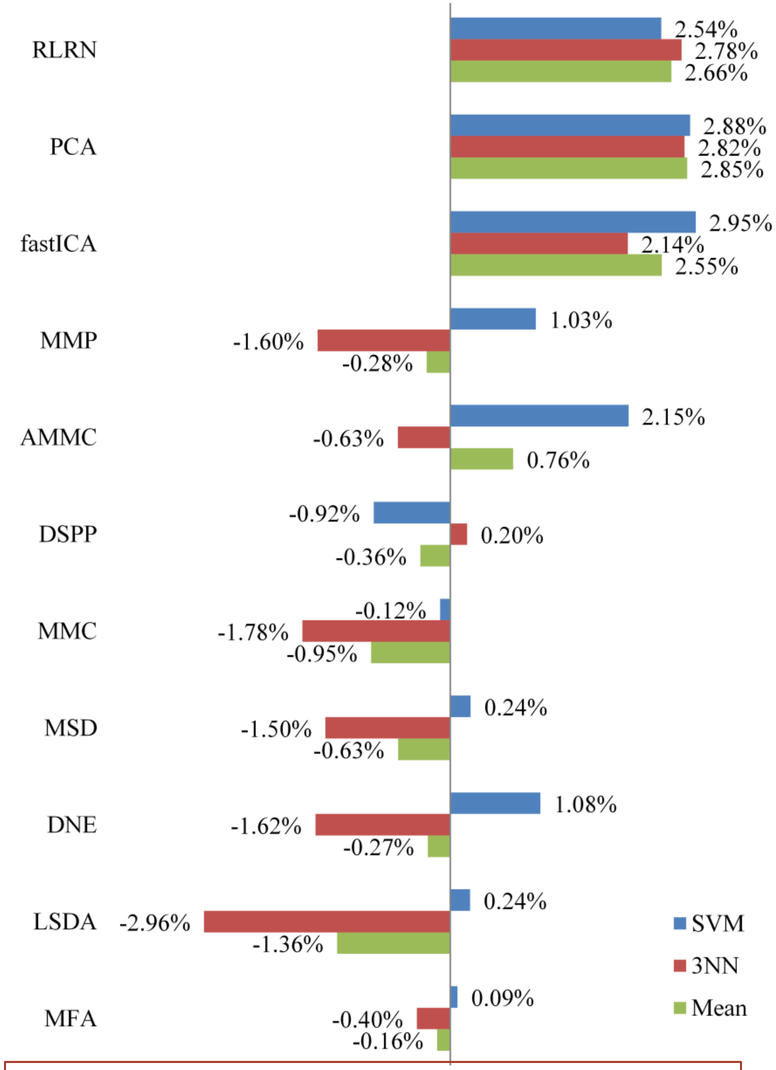
Methods for comparison

- Unsupervised
  1. PCA
  2. fastICA
- Unsupervised
  3. AMMC (Adaptive Maximum Margin Criterion)
  4. DSPP (Discriminative Sparsity Preserving Projection)
  5. MMC (Maximum Margin Criterion)
  6. MSD (Maximum Scatter Difference)
  7. DNE (Discriminant Neighborhood Embedding),)
  8. LSDA (Locality Sensitive Discriminant Analysis)
  9. MMP (Maximum Margin Projection)
  10. MFA (Marginal Fisher Analysis)

The highest improvement with respect to no embedding

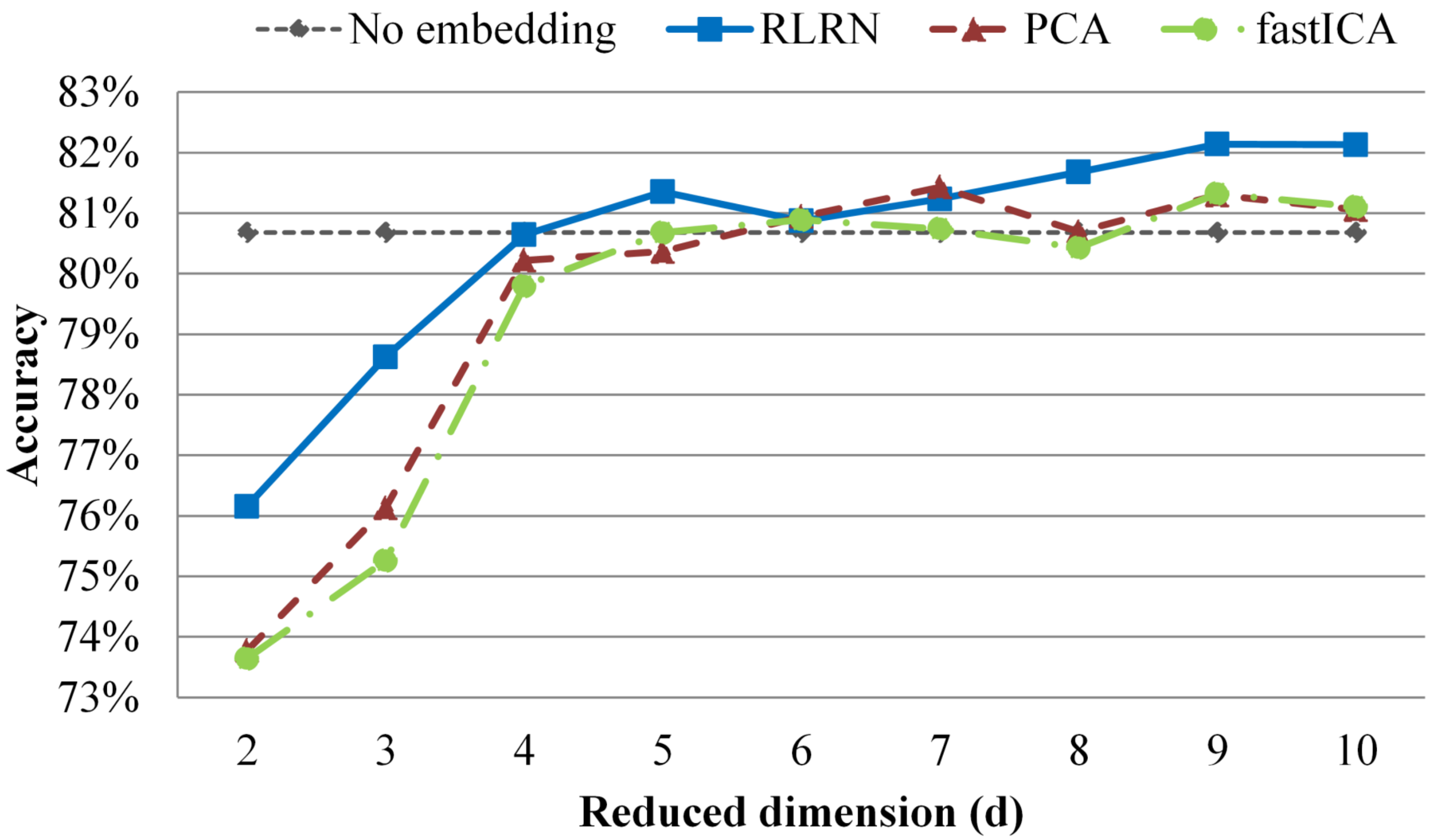


The highest improvement in AUC

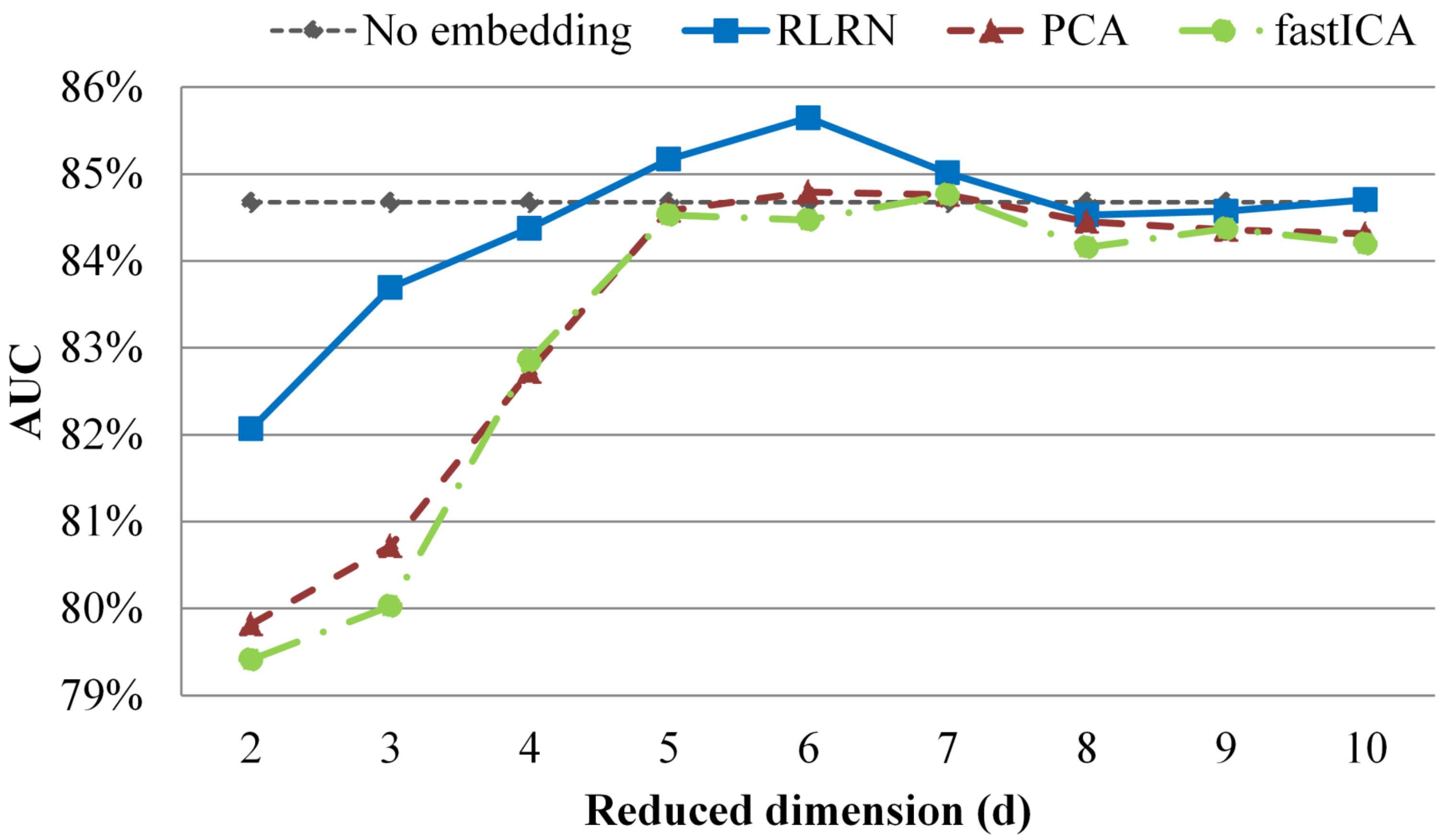


The highest improvement in Accuracy

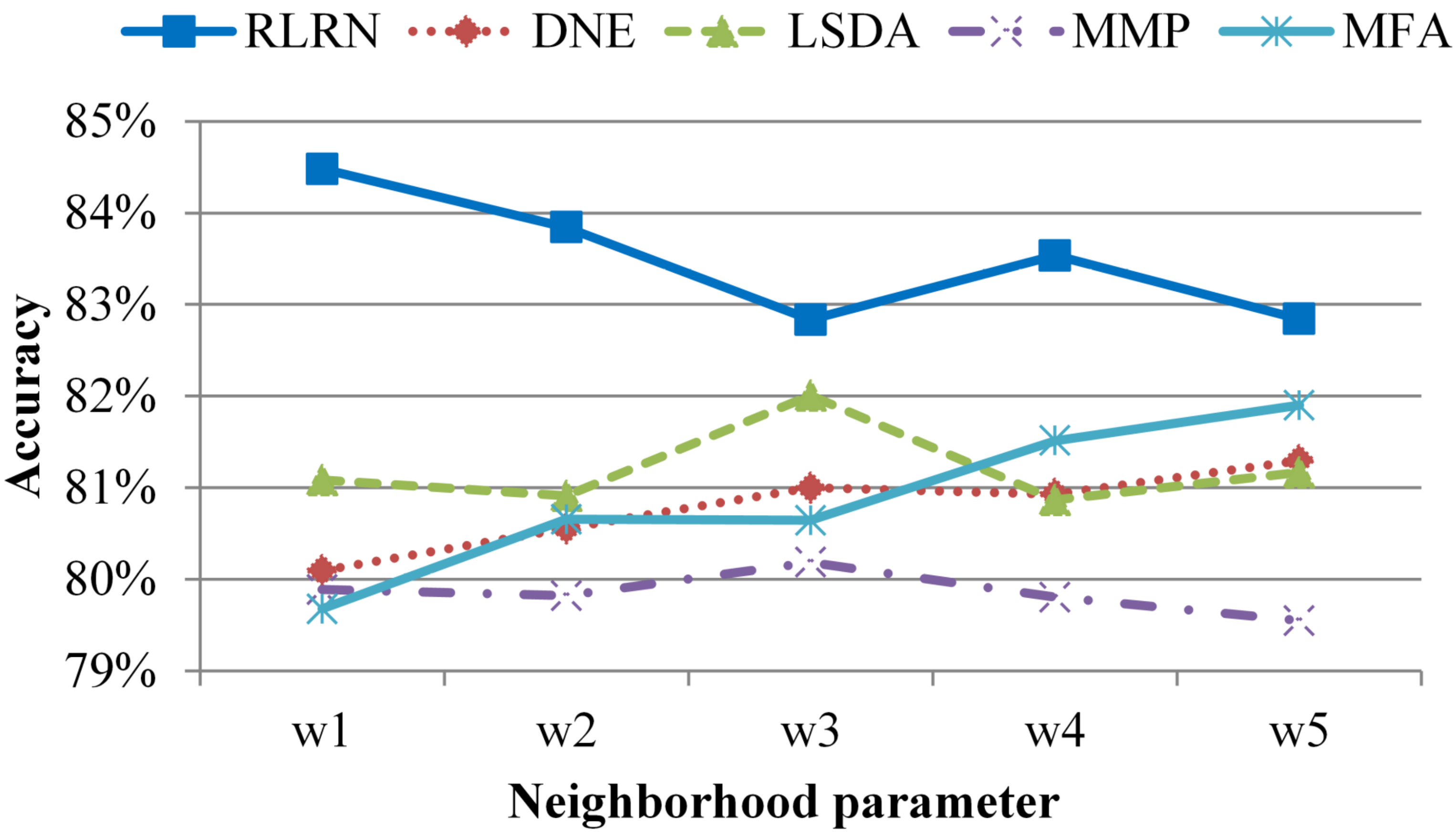
Mean accuracy of SVM and 3NN for different dimensions of embedded subspace



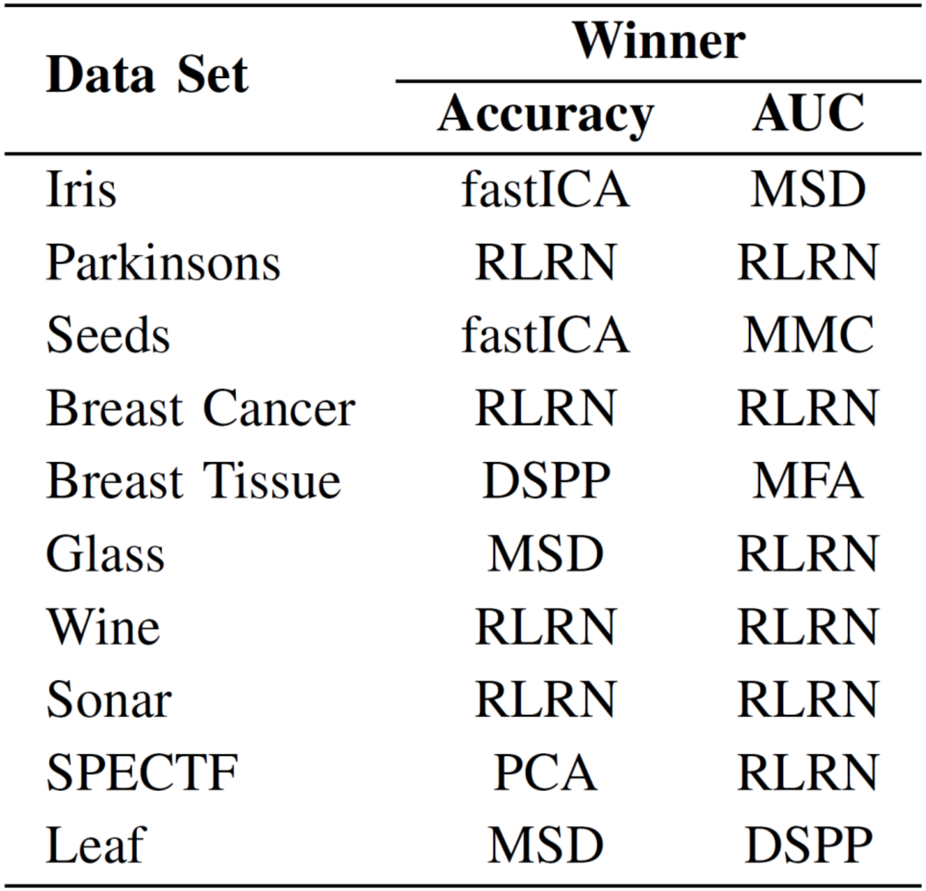
Mean AUC of SVM and 3NN for different dimensions of embedded subspace



The highest accuracy (mean of SVM and 3NN) for different settings of neighborhood parameter

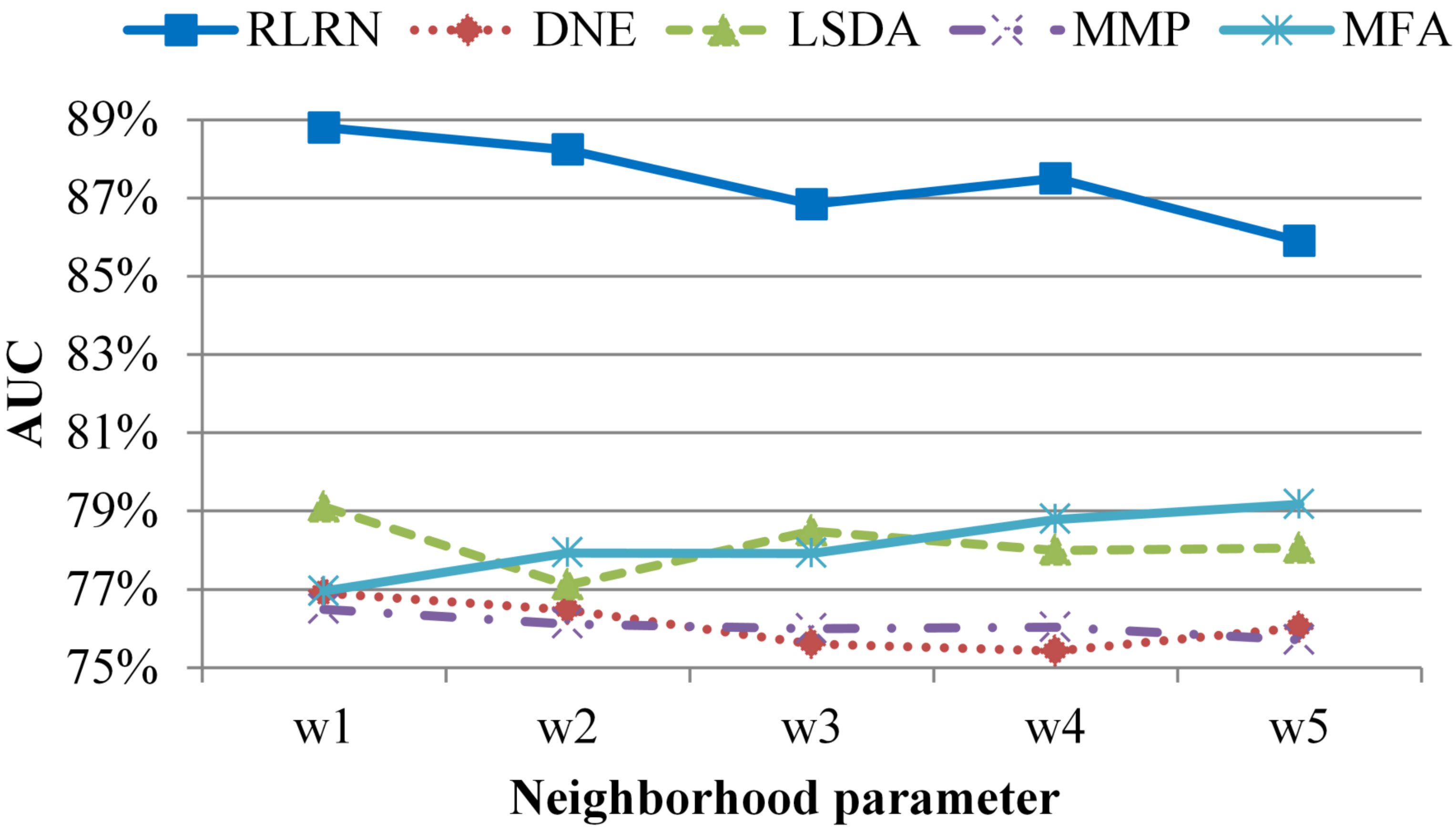


Methods reaching the highest accuracy and AUC for each data set



Data Set	Winner	
	Accuracy	AUC
Iris	fastICA	MSD
Parkinsons	RLRN	RLRN
Seeds	fastICA	MMC
Breast Cancer	RLRN	RLRN
Breast Tissue	DSPP	MFA
Glass	MSD	RLRN
Wine	RLRN	RLRN
Sonar	RLRN	RLRN
SPECTF	PCA	RLRN
Leaf	MSD	DSPP

The highest AUC (mean of SVM and 3NN) for different settings of neighborhood parameter



Conclusions

- Inspired by the well-known word2vec, RLRN employs a one/two-hot vector encoding and a FFNN.
- The FFNN learns the homogeneous neighborhoods of data points.
- A discriminative lower-dimensional subspace is generated in the hidden layer.
- RLRN is significantly superior to state-of-the art in terms of accuracy and AUC.

✓ As a possible future direction, adopting the proposed method to large data sets can be considered.

THANKS FOR YOUR ATTENTION