Unsupervised Feature Representation Learning via Random Features for Structured Data: Theory, Algorithm, and Applications

Lingfei Wu[1] and Ian En-Hsu Yen[2]
[1]IBM Research AI, IBM T. J. Watson Research Center
[2]Machine Learning Department, Carnegie Mellon University

Joint work with: Fangli Xu,
Pradeep Ravikumar, and Michael J. Witbrock

December 13, 2018

Knowledge Representation in Big Data

Knowledge representation is the key but very challenging:

- Learning representation from structured inputs from "unlabelled data" is non-trivial: transfer learning and domain adaptation
- Developing an effective and scalable approach for coping with large volumes of data is difficult

Contributions and Highlighted Research

Fundamental contributions in this research:

- D2KE: turn any distance functions into alignment-aware positive definite kernels and its embeddings
- D2KE: a generic learning framework for generating vector representations of structured inputs of any size such as time-series, text, strings, and graphs
- D2KE: generalized Random Features methods for structured inputs

Highlighted Research in this talk:

- Random Warping Series (RWS): an efficient and scalable method for multivariate time-series embedding
- Word Mover's Embedding (WME): an universal text embedding technique built on pre-trained word embeddings

Random Warping Series: A Random Features Method for Time Series Embedding

The problem and challenges

In general, a sequence is an ordered list of events.

"If data sequences exhibit temporal dependency implicitly (an ordering on values) or explicitly (with time stamps), they are often referred as the set of time series data $\{x_i\}_{i=1}^{N}$, where N is the number of time series and $L = |x_i|$ is the maximum length of each time series. "

Challenges for handling time series:

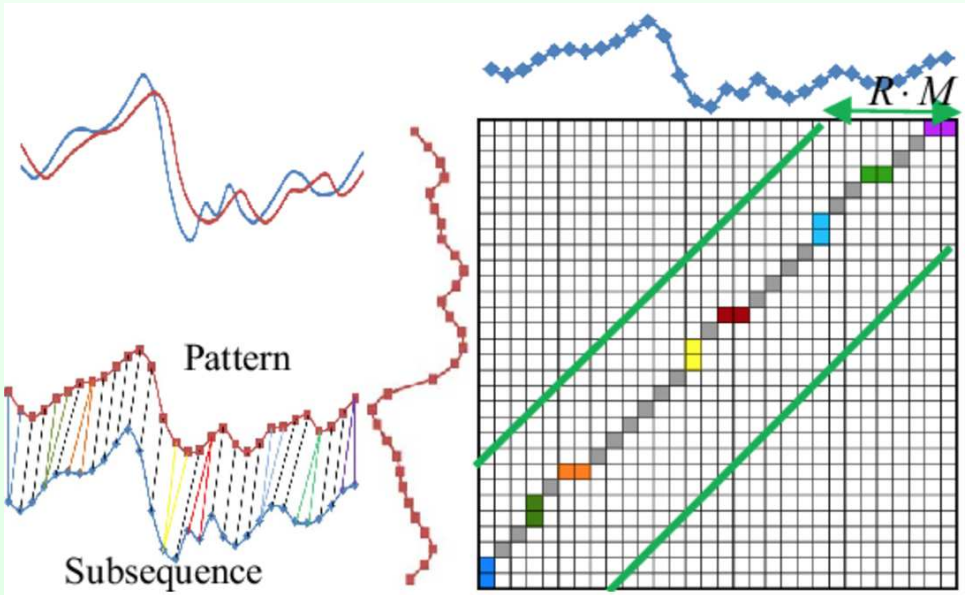
- No explicit features in sequences
- Two distinct characteristics of time series
  - variable length
  - dynamic time scaling and shifts

A deep look at Dynamic Time Warping (DTW)

The DTW distance between x and y is defined as

$$S(x, y) = \min_{a \in \mathcal{A}(x,y)} \tau(x, y; a),$$

where τ could be defined as any commonly used distance such as the squared Euclidean distance.



Existing approaches and their drawbacks

Three main threads of research from ML/DM:

- Feature representation methods deriving from only local patterns rather than global properties
  - effectiveness is application-dependent
  - high (quadratic) computation and memory costs
- Define a distance function to measure the similarity (Dynamic Time Warping)
  - 1NN classifier with DTW - standard benchmark
  - high (quadratic) computation and memory requirements
- Global alignment kernels with SVM (inspired by DTW)
  - consider all possible alignments resulting in diagonal dominance of kernel matrix
  - quadratic complexity in computation and memory

Random Warping Series for time-series embedding

We propose a novel time-series kernel via alignment to random sequences based on DTW:

$$k(x,y) = \int_{\omega} p(\omega)\phi_{\omega}(x)\phi_{\omega}(y)d\omega,$$

where $\phi_{\omega}(x) := \sum_{a \in \mathcal{A}(\omega,x)} p(a|\omega)\tau(\omega, x; a)$

To avoid the diagonal dominance problem of the kernel matrix, one can choose: $p(a|\omega) = \begin{cases} 1, & a = \arg\min_{a'} \ \tau(\omega, x; a') \\ 0, & o.w. \end{cases}$

Efficient computation of RWS

Although the kernel does not yield a simple analytic form, it naturally yields an random approximation of the form,

$$k(x,y) \approx \langle T(x), T(y) \rangle = \frac{1}{R} \sum_{i=1}^{R} \langle \phi_{\omega_i}(x), \phi_{\omega_i}(y) \rangle.$$

where $\mathsf{T}(x) := \frac{1}{\sqrt{R}} \tau(\{\omega_i\}_{i=1}^{R}, x)$ gives a vector representation of x and $\{\omega_i\}_{i=1}^{R}$ is a set of random series of length D with each value drawn from a distribution $p(\omega)$.

We reduce computation complexity from $O(N^2L^2)$ to $O(NRLD)$ and memory consumption from $O(NL + N^2)$ to $O(NR)$.
Note: we achieve linear (quadratic) speedup!

The RWS techqniue is fully parallelizable, and highly extensible, where the building block DTW can be replaced by recently proposed elastic distance measures such as CID and DTDC.

Convergence of RWS

Definition 1. *The* Minimum Shape-Preserving Length (MSPL) d_ϵ *of tolerance* ϵ *is the smallest* $\tilde{L}$ *such that*

$$\min_{\tilde{x}\in\mathbb{R}^{\tilde{L}},(A,I)\in\mathcal{A}(\tilde{x},x)} \|A\tilde{x} - Ix\| \leq \epsilon, \forall x \in \mathcal{X} \tag{1}$$

where $\mathcal{A}(\tilde{x}, x)$ *denotes the set of possible alignments between* $\tilde{x}$ *and* x *considered by DTW and* I *is the* L*-dimensional identity matrix.*

Theorem 1. *Let* $\tau(A\omega, Bx)$ *be bounded with* $|\tau(.,.)| \leq \gamma$ *and Lipschitz-continuous w.r.t.* x *with parameter* $\beta(\omega)$*, where* $Var[\beta(\omega)] \leq \sigma_\tau^2$*. The RWS approximation with* R *features satisfies*

$$P\left[\max_{x,y\in\mathcal{X}} |s_R(x,y) - k(x,y)| \geq 3\epsilon\right] \leq 8r^2\left(\frac{4\gamma\sigma_\tau}{\epsilon}\right)^2 e^{-\frac{R\epsilon^2}{32\gamma^4(1+d_\epsilon)}}.$$

where r *is the radius of time series domain* $\mathcal{X}$ *in the* ℓ_∞ *norm and* d_ϵ *is the MSPL with precision* ϵ*.*

Experiments and Setup

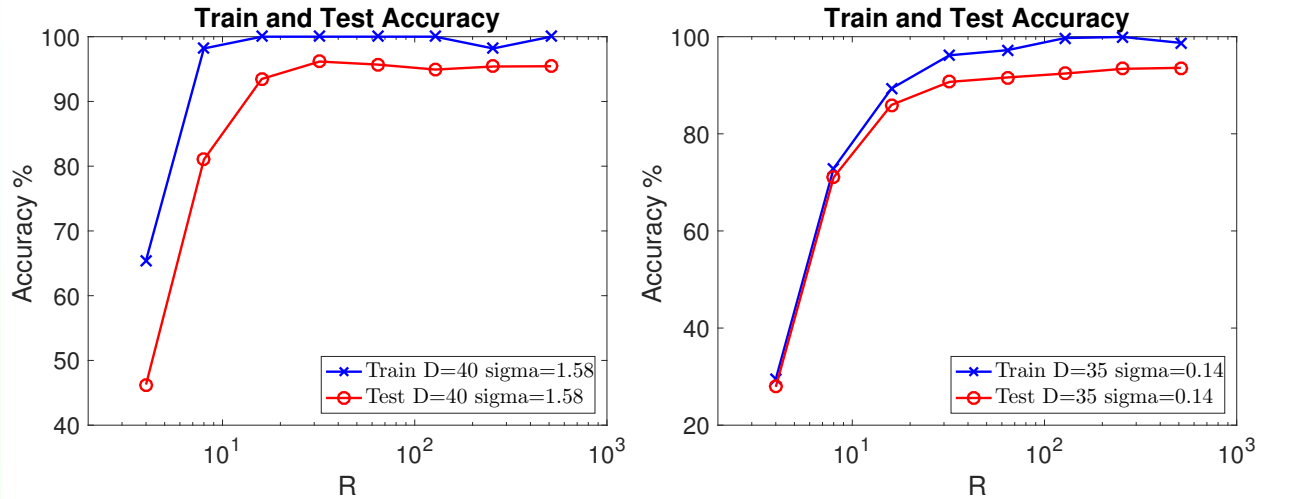
We compare RWS against 9 baselines on 16 real-world datasets from the widely-used UCR time-series archive.

Datasets are from various applications, including ECG, sensor, image, spectro, simulated and device.

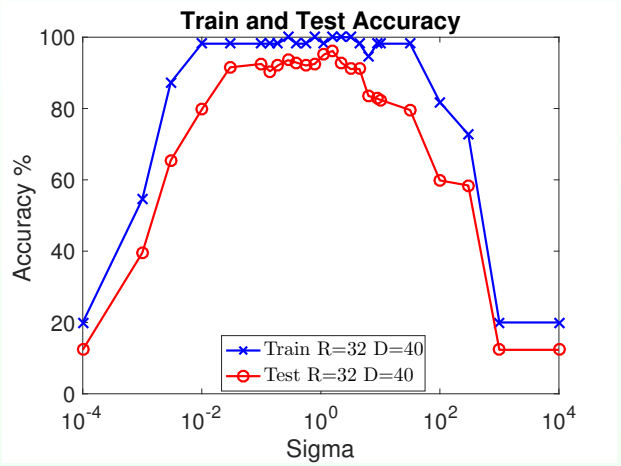
Table 1: Properties of the datasets. The number and the length of time series are sorted increasingly.

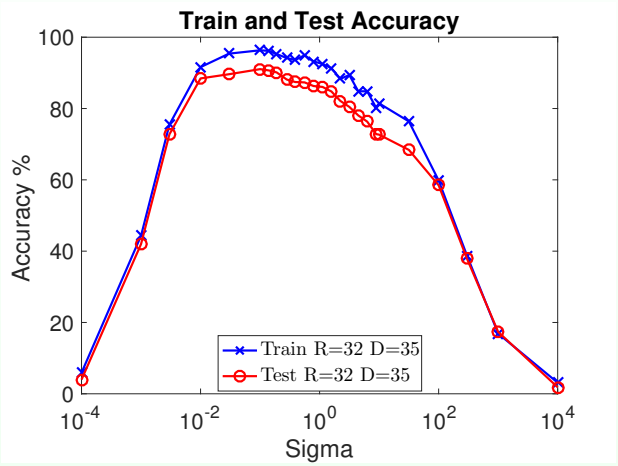
Name	C:Classes	N:Train	M:Test	L:length
LKA	3	375	375	720
IWBS	11	220	1,980	256
TWOP	4	1,000	4,000	128
ECG5T	5	500	4,500	140
MALLAT	8	55	2,345	1,024
FordB	2	3636	810	500
NIFECG	42	1,800	1,965	750
HO	2	370	1,000	2,709

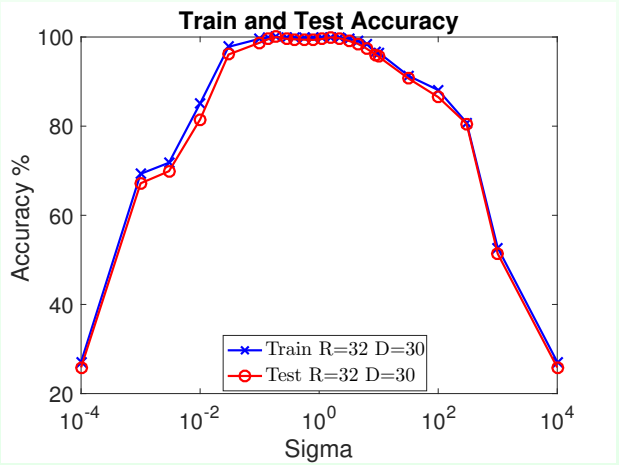
How fast is the convergence of RWS?

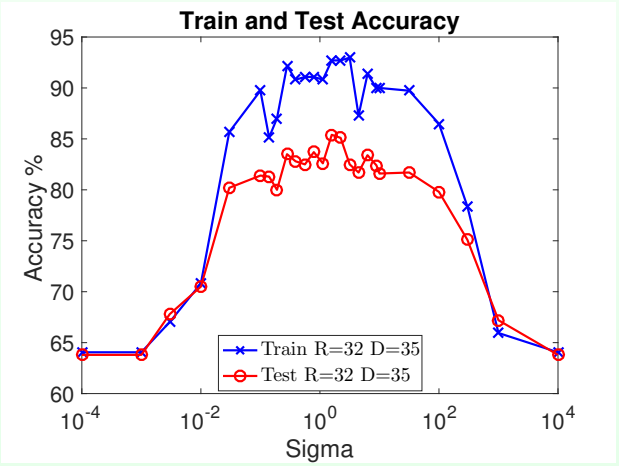


Is normal distribution enough for RWS?









Comparisons for time-series classification

Table 2: Classification performance comparison among multiple methods using DTW or DTW-like kernels.

Dataset	LKA		IWBS		TWOP		ECG5T	
Classifier	Accu	Time	Accu	Time	Accu	Time	Accu	Time
RWS(LR)	**0.843**	54.9	**0.641**	132.4	**1**	16.1	**0.94**	9.2
RWS(SR)	0.816	**13.6**	0.619	**8.8**	0.999	**4.4**	0.934	**4.9**
1NN-DTW	0.712	97.7	0.504	70.9	1	222.2	0.928	137.8
1NN-DTW^{opt}	0.837	573.6	0.589	36.1	1	157.5	0.928	70.1
TGAK	0.645	13484	0.126	2413	0.269	5690	0.927	2822
DTWF	0.80	1220	0.609	260.3	1	481.7	0.933	278.3

Dataset	MALLAT		FordB		NIFECG		HO	
Classifier	Accu	Time	Accu	Time	Accu	Time	Accu	Time
RWS(LR)	**0.952**	72.8	0.793	543.8	**0.936**	140.2	0.871	336.9
RWS(SR)	0.937	**33.8**	0.62	**5.6**	0.903	**20.0**	0.834	**41.9**
1NN-DTW	0.937	150.3	0.589	1476	0.845	2699	0.816	4883
1NN-DTW^{opt}	0.925	65.5	0.581	577.6	0.857	1432	0.807	5837
TGAK	0.257	11882	N/A	N/A	N/A	N/A	N/A	N/A
DTWF	0.915	988.4	**0.83**	8402	0.906	32493	**0.898**	40407

Comparisons for time-series clustering

Table 3: Clustering performance comparison among different methods.

Dataset	Beef		DPTW		PPOAG		IWBS	
Clustering	NMI	Time	NMI	Time	NMI	Time	NMI	Time
RWS(LR)	**0.29**	*1.1*	0.52	*0.6*	**0.56**	*0.5*	**0.43**	*43.9*
RWS(SR)	*0.27*	**1.0**	**0.56**	**0.5**	*0.54*	**0.2**	0.36	**6.3**
KMeans-DTW	0.25	377	*0.55*	182	0.44	105.4	0.37	5676
CLDS	0.24	61.3	*0.55*	176.8	0.55	191.1	0.38	1109
K-Shape	0.22	1.8	0.45	14.9	0.27	40.2	**0.43**	377.6

Dataset	TWOP		ECG5T		MALLAT		NIFECG	
Clustering	NMI	Time	NMI	Time	NMI	Time	NMI	Time
RWS(LR)	0.23	*11.2*	*0.46*	*25.7*	**0.92**	*48.2*	*0.71*	*346.1*
RWS(SR)	*0.3*	**4.7**	0.4	**7.0**	*0.91*	**25.4**	0.68	**43.7**
KMeans-DTW	0.12	1960	**0.48**	2539	0.72	95218	0.63	101473
CLDS	0.02	1312	0.37	1308	**0.92**	2448	0.67	3442
K-Shape	**0.4**	292.1	0.35	360.7	0.75	900.4	**0.73**	5387

Word Mover's Embedding: From Word2Vec to Document Embedding

The problem and challenges

Text representation plays an important role in many NLP tasks. In general, a document is an ordered list of words.

"A definition of a document is that it is made of a joint membership of terms which have various patterns of occurrence. Document representation is concerned about how textual documents should be represented in various tasks, e.g. text processing, retrieval..."

Two challenges for handling document/text:

- Learning semantic preserving document representation
- Capturing long range dependence of words (word order)

Interesting question: shall sentence/document representation be built on word representation or from scratch?

Existing approaches and their drawbacks

Main threads of research for learning text representation:

- Traditional BOW and TF-IDF
  - near-orthogonality with high-dimension sparse vectors
  - fail to consider word order and semantics of words
- Latent semantic indexing or Latent Dirichlet Allocation
  - Latent low-dimensional document representation
  - often not improve empirical performance of BOW
- Word2Vec for representation of sentences and documents
  - surprisingly high-quality pre-trained word embeddings
  - simple weighted average loses word order information
- Doc2Vec for representation of sentences and documents
  - jointly learn embeddings of words and paragraphs
  - word order not fully captured by small context window
  - low quality of word embeddings limited by size of corpus

Word Mover's Distance for documents

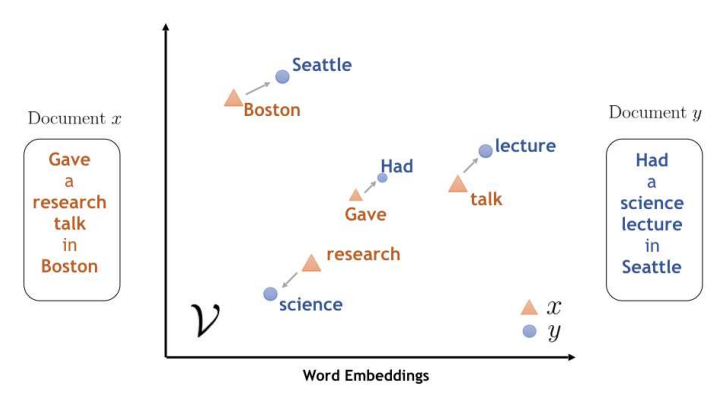
Word Mover's Distance measures dissimilarity between two multi-dimensional distributions over words in documents x and y:

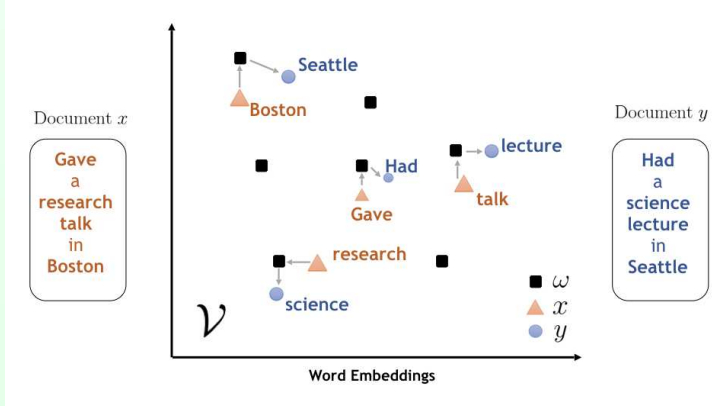
$$\text{WMD}(x, y) := \min_{F \in \mathbb{R}_+^{|x| \times |y|}} \langle C, F \rangle, \ s.t., \ F\mathbf{1} = \boldsymbol{f}_x, F^\top \mathbf{1} = \boldsymbol{f}_y.$$

However, two disadvantages:

1) A distance that can only be combined with KNN or KMean;

2) High computational complexity $O(N^2 L^3 \log(L))$





Document Embedding via Word Mover's Kernel

The Word Mover's Kernel is defined:

$$k(x, y) := \int p(\omega)\phi_\omega(x)\phi_\omega(y)d\omega$$

where $\phi_\omega(x) := \exp(-\gamma \text{WMD}(x, \omega))$ is an infinite-dimensional feature map derived from WMD.

An insightful interpretation of the kernel is expressing as:

$$k(x, y) := \exp\left(-\gamma softmin_{p(\omega)}\{\text{WMD}(x, \omega) + \text{WMD}(\omega, y)\}\right)$$

where $softmin_{p(\omega)}\ f(\omega) := -\frac{1}{\gamma}\log\int p(\omega)e^{-\gamma f(\omega)}d\omega$.

According to the triangular inequality of WMD, we have

$$\text{WMD}(x, y) \ \leq\ \min_{\omega\in\Omega}\ \{\text{WMD}(x, \omega) + \text{WMD}(\omega, y)\}$$

Thus, $k(x, y)$ serves as a good approximation to the WMD.

Computation of Word Mover's Embedding

Although the kernel does not yield a simple analytic form, it naturally yields an random approximation of the form,

$$k(x,y) \approx \langle Z(x), Z(y) \rangle = \frac{1}{R} \sum_{i=1}^{R} \phi_{\omega_i}(x) \phi_{\omega_i}(y)$$

where $\{\omega_i\}_{i=1}^{R}$ are i.i.d. random documents drawn from $p(\omega)$ and $\boldsymbol{z}(x) := (\frac{1}{\sqrt{R}} \phi_{\omega_i}(x))_{i=1}^{R}$ gives a vector representation of x.

We reduce computation complexity from $O(N^2 L^3 log(L))$ to $O(NRLlog(L))$. Note: we achieve linear speedup in number of documents (N) and quadratic speedup in length of documents (L)!

WME is fully parallelizable, and is highly extensible where its two building blocks, Word2Vec and WMD, can be replaced by other techniques such as GloVe or S-WMD.

Convergence of WME

Lemma 1. *There is an ϵ-covering $\mathcal{E}$ of $\mathcal{X}$ under the metric defined by WMD with Euclidean ground distance that satisfies*

$$\forall x \in \mathcal{X}, \exists x_i \in \mathcal{E},\ \textit{WMD}(x, x_i) \leq \epsilon.$$

with $|\mathcal{E}| \leq (\frac{2}{\epsilon})^{dL_{\max}}$, where $L_{\max}$ is a bound on the length of document $x \in \mathcal{X}$.

Theorem 2. *Let $\Delta_R(x, y)$ be the difference between the exact kernel and the random approximation with R samples, we have uniform convergence*

$$P\left\{\max_{x,y\in\mathcal{X}} |\Delta_R(x, y)| > 2t\right\} \leq 2\left(\frac{12\gamma}{t}\right)^{2dL_{\max}} \exp(-Rt^2/2).$$

where d is the dimension of word embedding and $L_{\max}$ is a bound on the document length.

Experiments and Setup

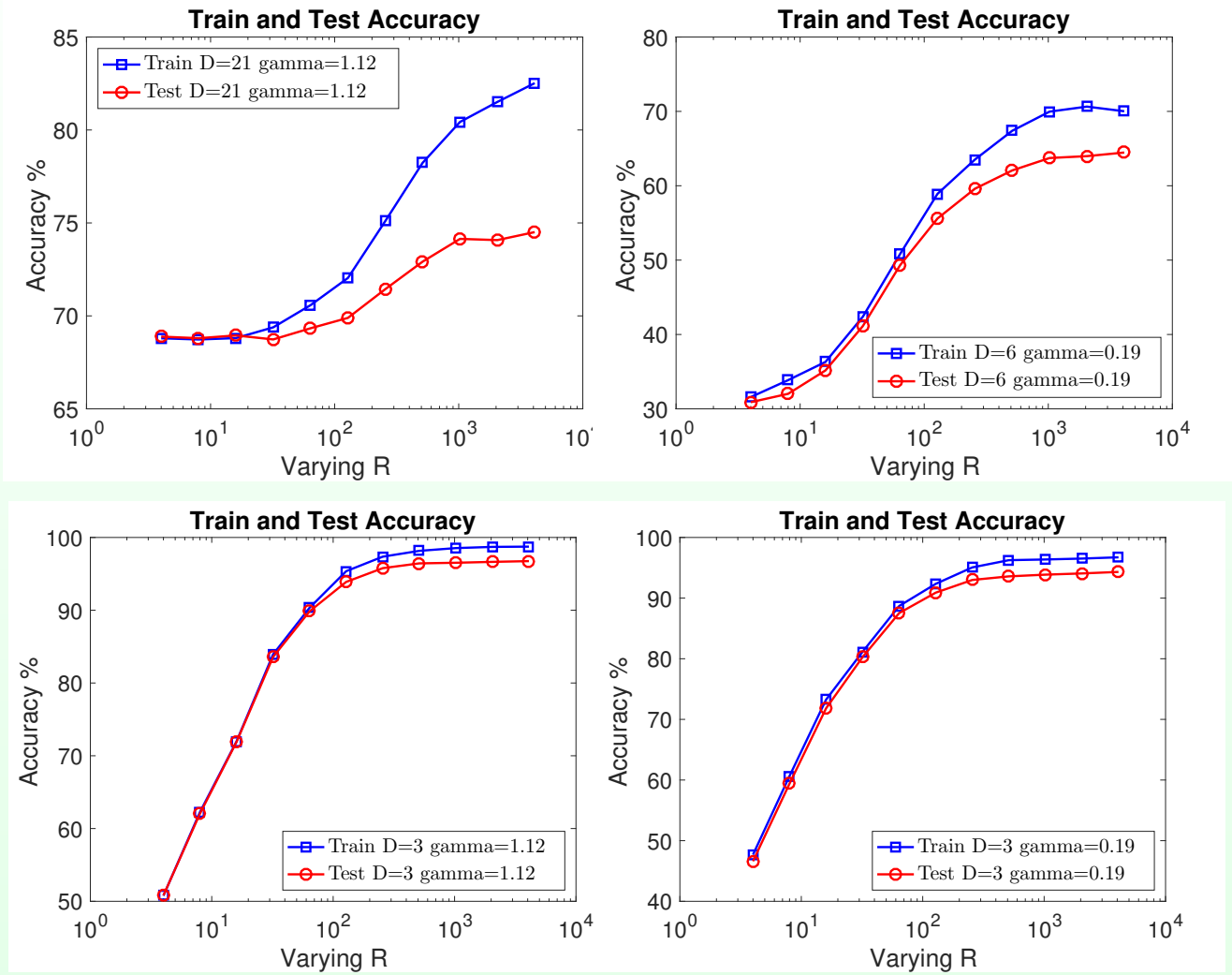
We first compare WME against 7 baselines on 9 datasets over a wide range of text classification tasks, including sentiment analysis, news categorization, amazon review, recipe identification.

We further compare our method against 10 baselines on the 22 datasets from SemEval semantic textual similarity (STS) tasks.

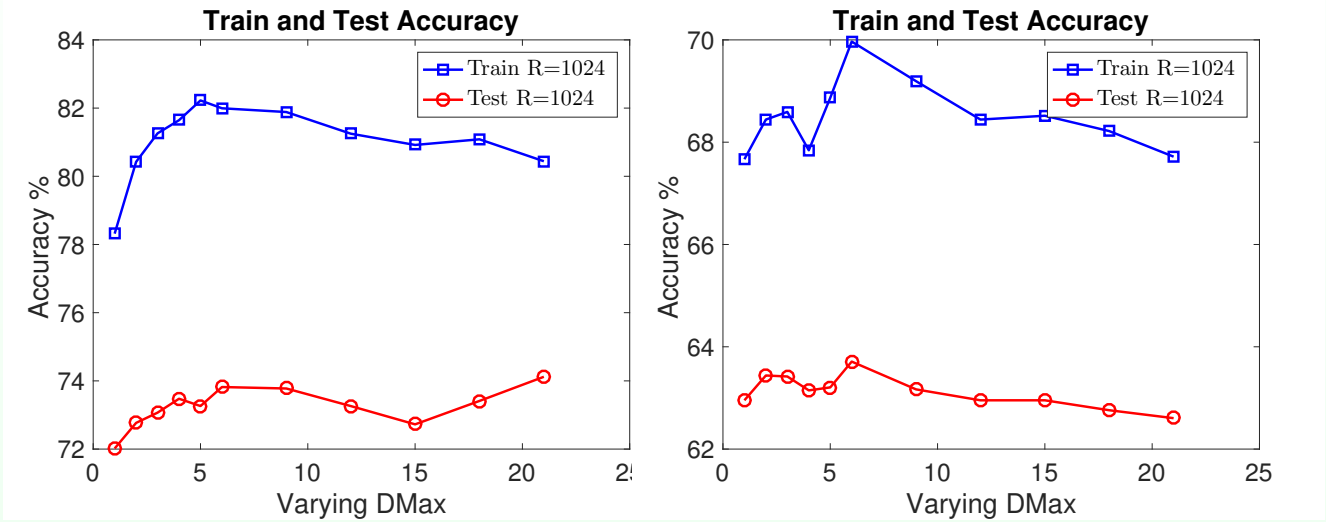
Table 4: Properties of the datasets

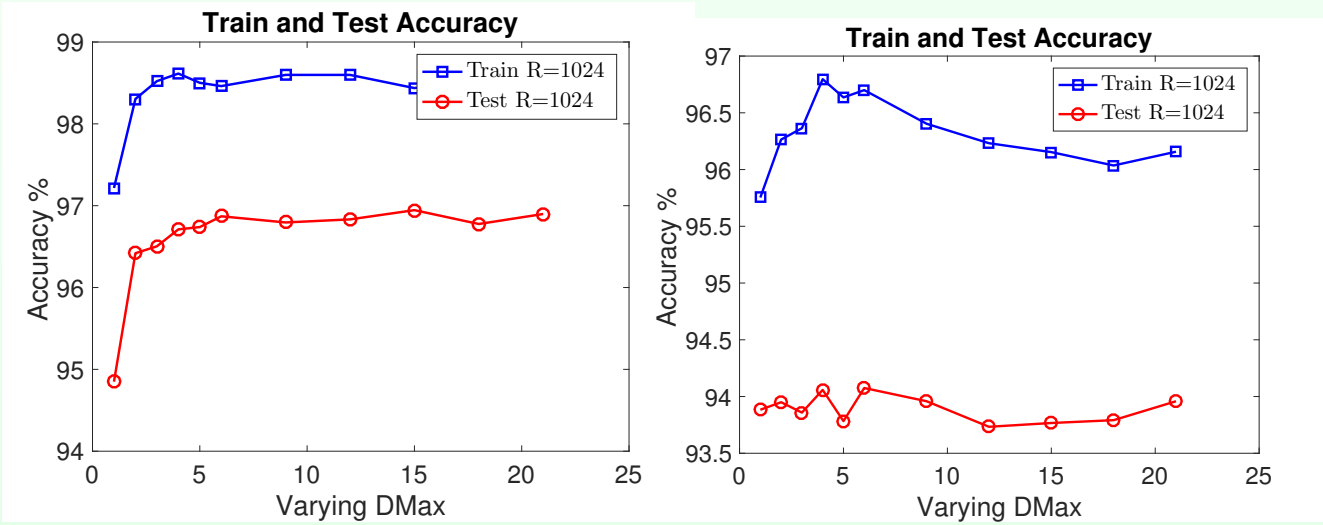
Dataset	C:Classes	N:Train	M:Test	BOW Dim	L:Length
BBCSPORT	5	517	220	13243	117
TWITTER	3	2176	932	6344	9.9
RECIPE	15	3059	1311	5708	48.5
OHSUMED	10	3999	5153	31789	59.2
CLASSIC	4	4965	2128	24277	38.6
REUTERS	8	5485	2189	22425	37.1
AMAZON	4	5600	2400	42063	45.0
20NEWS	20	11293	7528	29671	72
RECIPE_L	20	27841	11933	3590	18.5

The effect of R on Random Features



The effect of D on Random Features





Comparisons against KNN-based classification

Table 1: Testing accuracy comparing WME against KNN-based methods

Dataset	BOW	TF-IDF	BM25	LSI
BBCSPORT	79.4 ± 1.2	78.5 ± 2.8	83.1 ± 1.5	95.7 ± 0.6
TWITTER	56.4 ± 0.4	66.8 ± 0.9	57.3 ± 7.8	68.3 ± 0.7
RECIPE	40.7 ± 1.0	46.4 ± 1.0	46.4 ± 1.9	54.6 ± 0.5
OHSUMED	38.9	37.3	33.8	55.8
CLASSIC	64.0 ± 0.5	65.0 ± 1.8	59.4 ± 2.7	93.3 ± 0.4
REUTERS	86.1	70.9	67.2	93.7
AMAZON	71.5 ± 0.5	58.5 ± 1.2	41.2 ± 2.6	90.7 ± 0.4
20NEWS	42.2	45.6	44.1	71.1

Dataset	LDA	mSDA	KNN-WMD	WME
BBCSPORT	93.6 ± 0.7	91.6 ± 0.8	95.4 ± 0.7	**98.2 ± 0.6**
TWITTER	66.2 ± 0.7	67.7 ± 0.7	71.3 ± 0.6	**74.5 ± 0.5**
RECIPE	48.7 ± 0.6	52 ± 1.4	57.4 ± 0.3	**61.8 ± 0.8**
OHSUMED	49.0	50.7	55.5	**64.5**
CLASSIC	95.0 ± 0.3	93.1 ± 0.4	**97.2 ± 0.1**	97.1 ± 0.4
REUTERS	93.1	91.9	96.5	**97.2**
AMAZON	88.2 ± 0.6	82.9 ± 0.4	92.6 ± 0.3	**94.3 ± 0.4**
20NEWS	68.5	60.5	73.2	**78.3**

Comparisons against KNN-WMD in runtime

Table 2: Testing accuracy, and total training and testing time (Seconds) of WME against KNN-WMD

Classifier	KNN-WMD	KNN-WMD+P	WME(SR)	WME(SR)+P		WME/KNN-WMD
Dataset	Accu	Time	Time	Accu	Time	Time	Speedup
BBCSPORT	94.5	147	122	95.5	3	1	122
TWITTER	72.3	25	4	72.5	10	2	2
RECIPE	56.1	448	326	57.4	18	4	82
OHSUMED	55.8	3530	2807	55.8	24	7	401
CLASSIC	96.9	777	520	96.6	49	10	52
REUTERS	96.1	814	557	96.0	50	24	23
AMAZON	92.6	2190	1319	92.7	31	8	165
20NEWS	73.2	37988	32610	72.9	205	69	472
RECIPE_L	71.4	5942	2060	72.5	113	20	103

Comparisons against Word2Vec and Doc2Vec

Table 3: Testing accuracy of WME against Word2Vec-based and Doc2Vec-based methods

Dataset	SIF(GloVe)	Word2Vec+nbow	Word2Vec+tf-idf
BBCSPORT	97.3 ± 1.2	97.3 ± 0.9	96.9 ± 1.1
TWITTER	57.8 ± 2.5	72.0 ± 1.5	71.9 ± 0.7
OHSUMED	**67.1**	63.0	60.6
CLASSIC	92.7 ± 0.9	95.2 ± 0.4	93.9± 0.4
REUTERS	87.6	96.9	95.9
AMAZON	94.1 ± 0.2	94.0 ± 0.5	92.2 ± 0.4
20NEWS	72.3	71.7	70.2
RECIPE_L	71.1 ± 0.5	74.9 ± 0.5	73.1 ± 0.6

Dataset	PV-DBOW	PV-DM	Doc2VecC	WME
BBCSPORT	97.2 ± 0.7	97.9 ± 1.3	90.5 ± 1.7	**98.2 ± 0.6**
TWITTER	67.8 ± 0.4	67.3 ± 0.3	71.0 ± 0.4	**74.5 ± 0.5**
OHSUMED	55.9	59.8	63.4	64.5
CLASSIC	97.0 ± 0.3	96.5 ± 0.7	96.6 ± 0.4	**97.1 ± 0.4**
REUTERS	96.3	94.9	96.5	**97.2**
AMAZON	89.2 ± 0.3	88.6 ± 0.4	91.2 ± 0.5	**94.3 ± 0.4**
20NEWS	71.0	74.0	78.2	**78.3**
RECIPE_L	73.1 ± 0.5	71.1 ± 0.4	76.1 ± 0.4	**79.2 ± 0.3**

Comparisons for textual similarity tasks

Table 4: Pearson's scores of WME against other unsupervised and supervised methods on 22 textual similarity tasks

Approaches	Supervised					
Tasks	PP	Dan	RNN	iRNN	LSTM(no)	LSTM(o.g.)
STS'12	58.7	56.0	48.1	58.4	51.0	46.4
STS'13	55.8	54.2	44.7	**56.7**	45.2	41.5
STS'14	**70.9**	69.5	57.7	**70.9**	59.8	51.5
STS'15	**75.8**	72.7	57.2	75.6	63.9	56.0
SICK'14	71.6	70.7	61.2	71.2	63.9	59.0
Twitter'15	52.9	**53.7**	45.1	52.9	47.6	36.1

Approaches	Unsupervised				
Tasks	ST	GV+ave	GV+tf-idf	SIF	WME
STS'12	30.8	52.5	58.7	56.2	**60.6**
STS'13	24.8	42.3	52.1	56.6	54.5
STS'14	31.4	54.2	63.8	68.5	65.5
STS'15	31.0	52.7	60.6	71.7	61.8
SICK'14	49.8	65.9	69.4	**72.2**	68.0
Twitter'15	24.7	30.3	33.8	48.0	41.6

Thank you for your attention!

Any Question?