Large-Scale Heterogeneous Feature Embedding

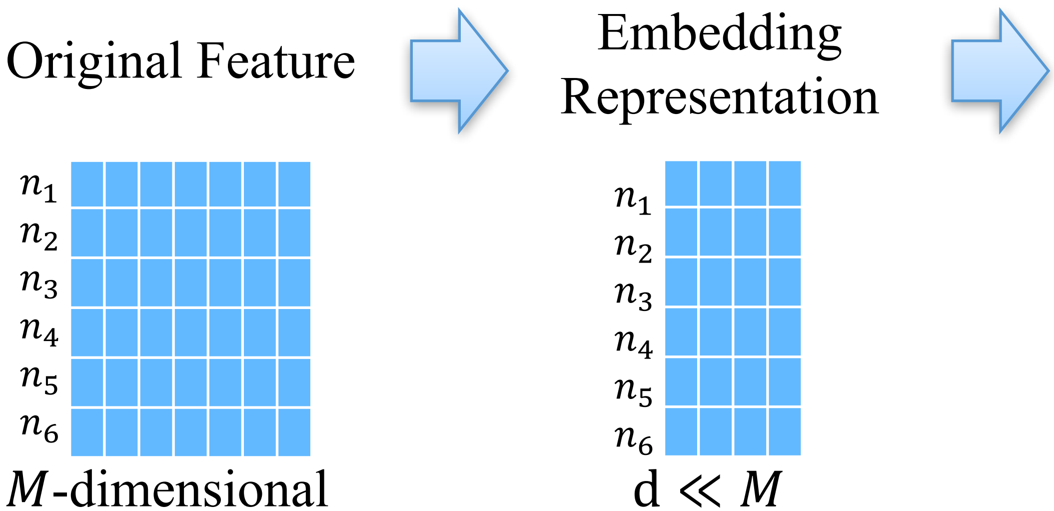
Xiao Huang, Qingquan Song, Fan Yang, and Xia (Ben) Hu

Computer Science & Engineering, Texas A&M University, College Station, TX, USA

Emails: {xhuang, song_3134, nacoyang, xiahu}@tamu.edu

AAAI 2019, Honolulu, USA

What is Feature Embedding



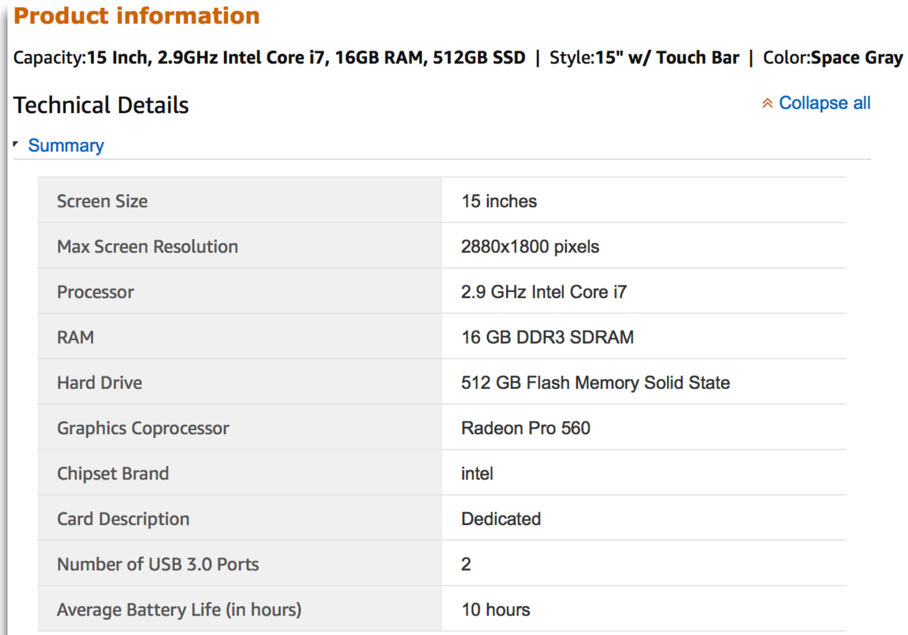
Applications

- Syntactic Analysis
- Action Recognition
- Anomaly Detection
- …

➢ Goal: Learn a low-dimensional vector representation for each instance, such that all original information is preserved.

➢ The learned vector representations could be directly applied to and benefit real-world applications.

Multiple Types of Correlated Features are Available



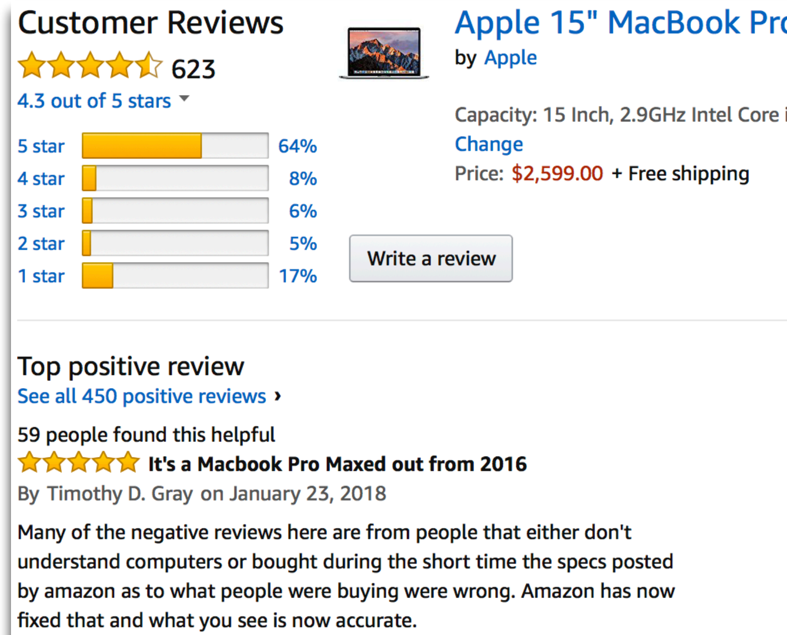
Product information

Capacity:15 Inch, 2.9GHz Intel Core i7, 16GB RAM, 512GB SSD | Style:15" w/ Touch Bar | Color:Space Gray

Technical Details — Collapse all

Summary

Screen Size	15 inches
Max Screen Resolution	2880x1800 pixels
Processor	2.9 GHz Intel Core i7
RAM	16 GB DDR3 SDRAM
Hard Drive	512 GB Flash Memory Solid State
Graphics Coprocessor	Radeon Pro 560
Chipset Brand	intel
Card Description	Dedicated
Number of USB 3.0 Ports	2
Average Battery Life (in hours)	10 hours



- Amazon products: product info, customer reviews, etc.
  - Network 1: customer purchase records
  - Network 2: customer viewing history
- Real-world instances often contain multiple types of correlated features or even data of a distinct modality such networks.

Example of Multiple Types of Features

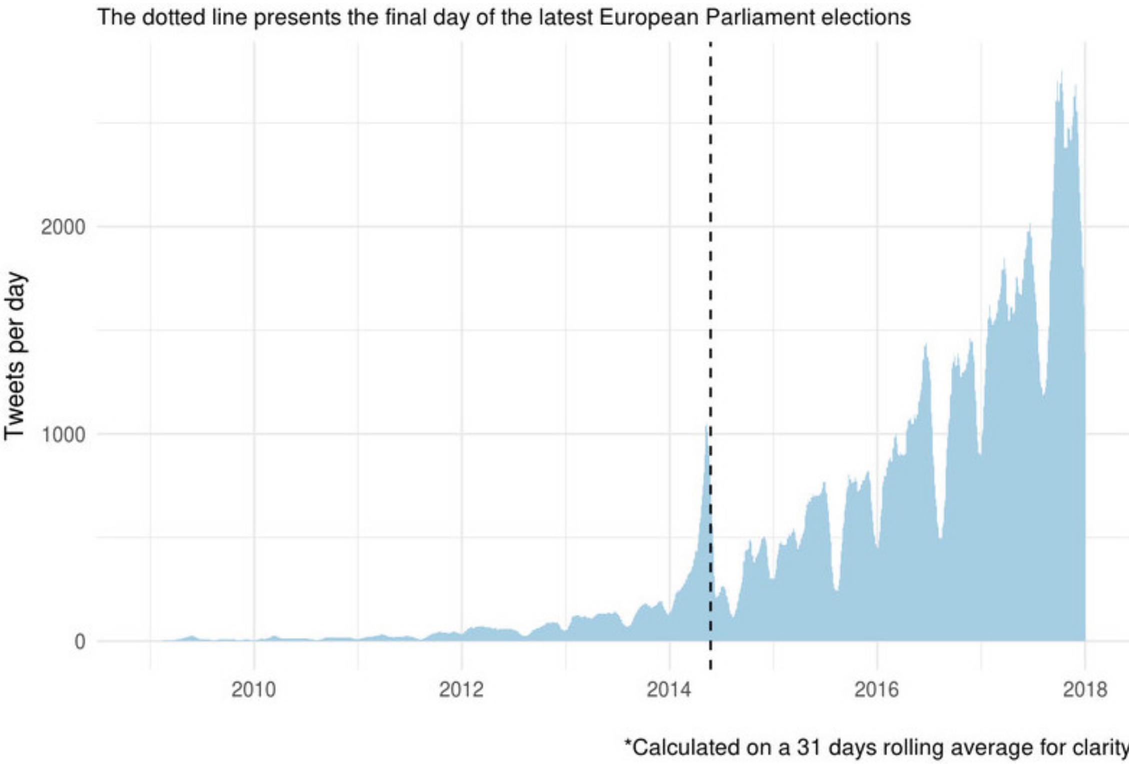
- Twitter users: attributes in introduction, words in tweets, content in photos, etc.

Joint Learning Benefits Embedding

- **Inherently Correlated:**
  - Posts reflect status
  - Social status impact words
  - Friends tend to share posts with similar topics

- **Multiview learning & Attributed network embedding:**
  It is promising to perform feature embedding based on features collected from multiple aspects.

Challenges

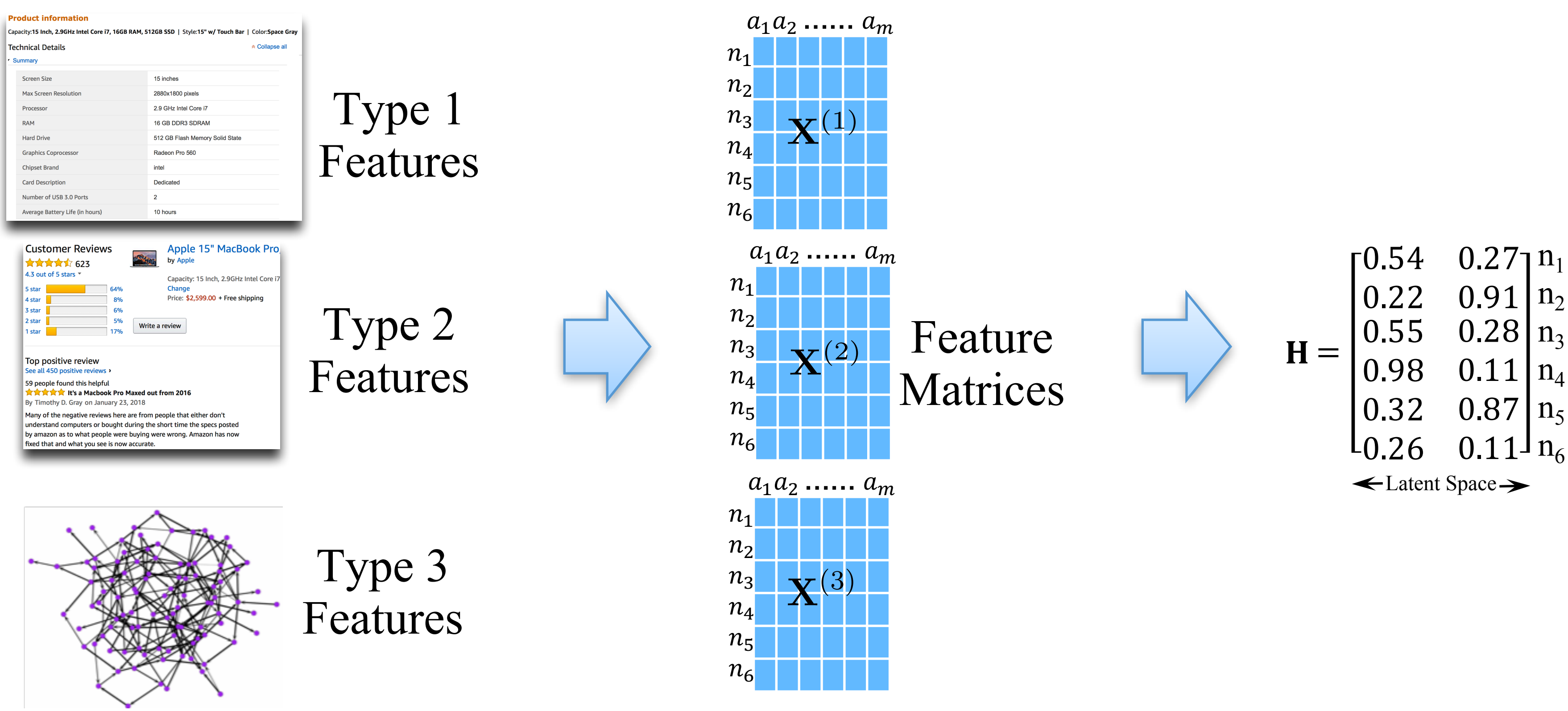


- Number of tweets posted by European Parliament per day
- Dotted line: European Parliament elections

➢ Ever-growing data volume along with the complex data properties put demands on the scalability of algorithms.

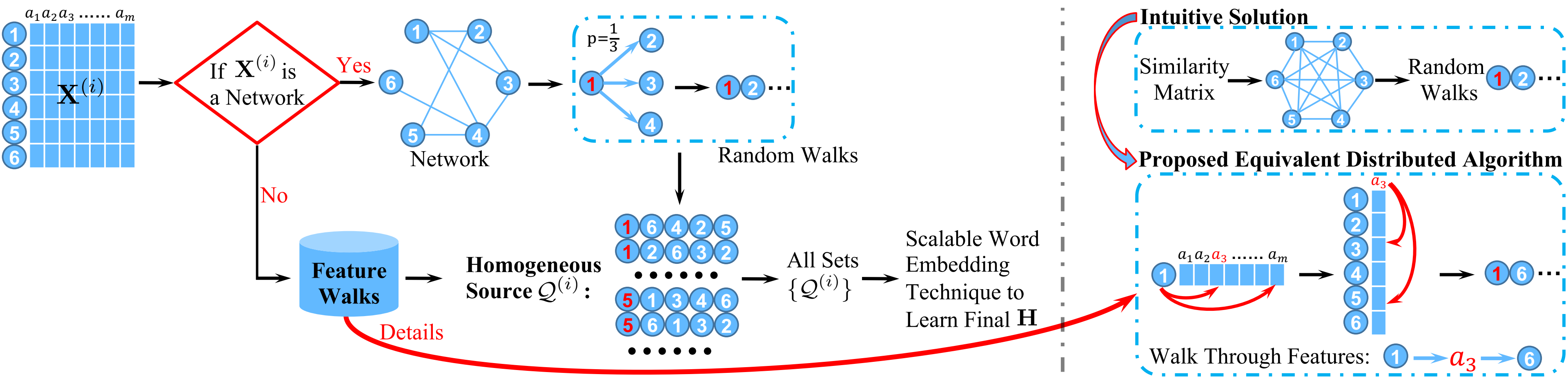
➢ Real-world features are often heterogeneous sources or even within a different modality such as networks.

Large-scale Heterogeneous Feature Embedding



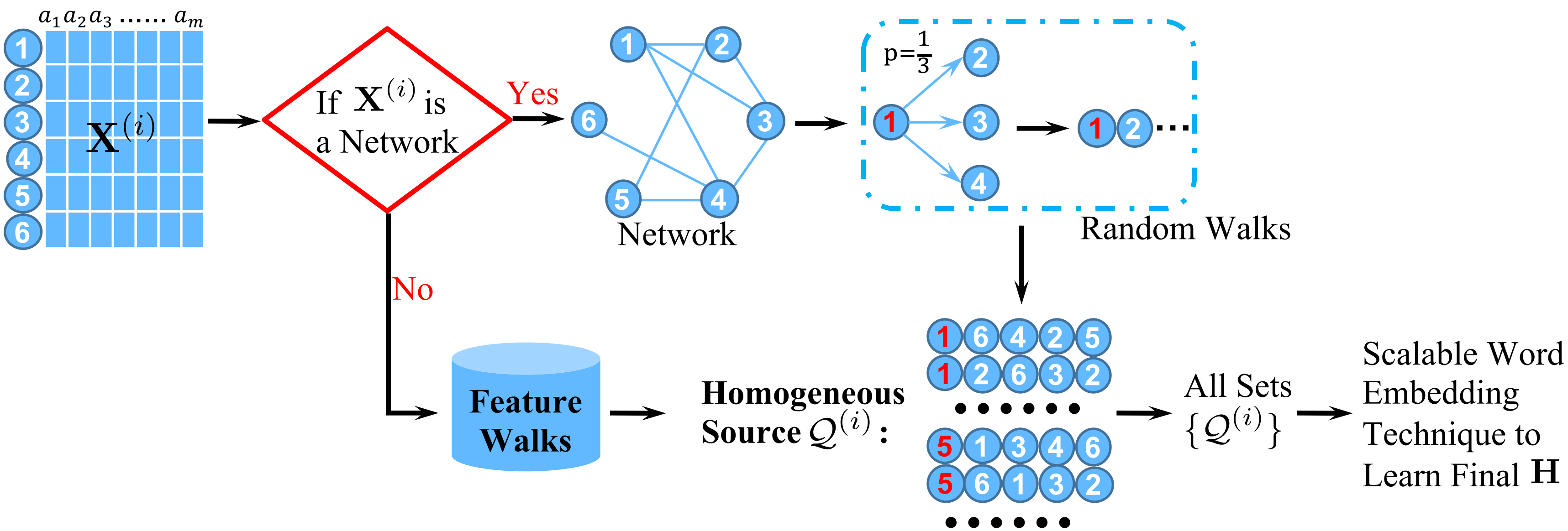
- **Input:** A large number of instances, associated with a set of instance feature matrices and an instance relation network.
- **Output:** A low-dimensional representation $\mathbf{h}_i$ for each instance.
- **Goal:** All meaningful information are well preserved in $\mathbf{H}$.

Proposed Framework FeatWalk



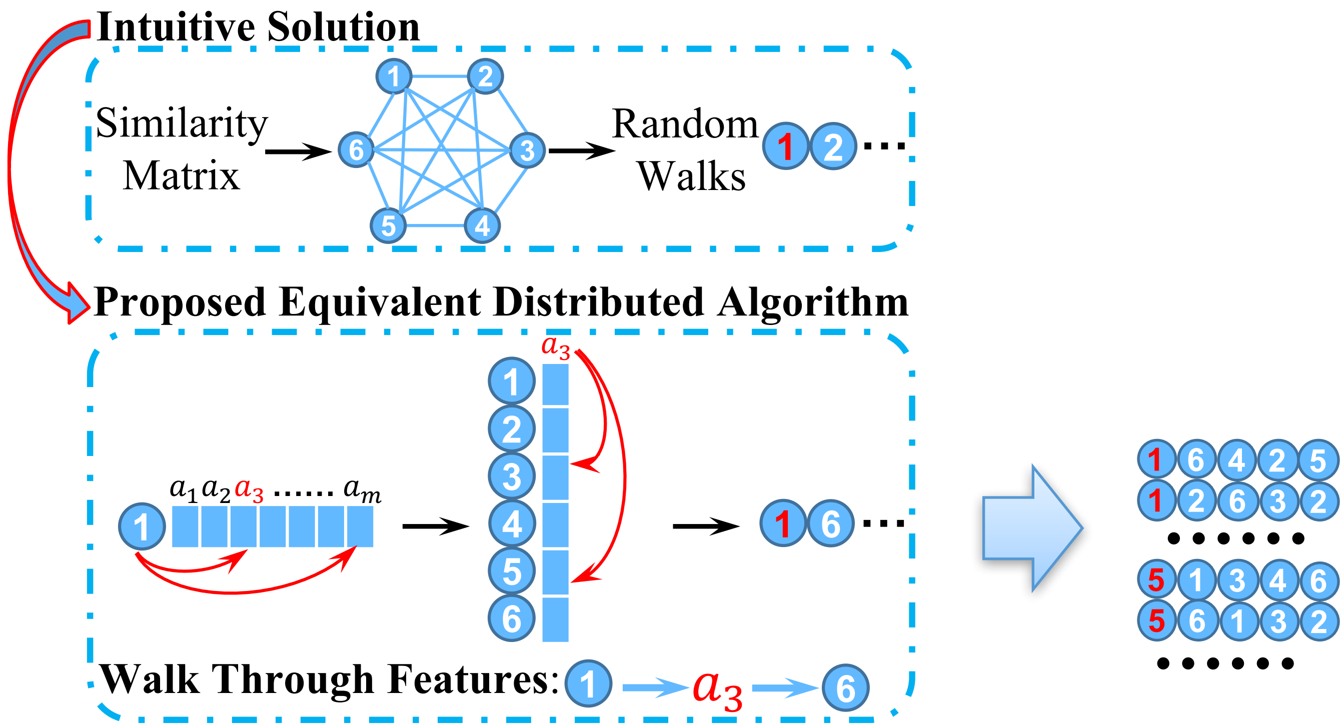
- **Goal:** Incorporate multiple types of high-dimensional feature matrices & networks into unified vector representations.
- **Key Ideas:**
  - Avoid computing similarity measure
  - Alternative way to simulate the similarity-based random walks among instances to sample the local instance proximity.

Learn Instance Proximities to Handle Heterogeneity



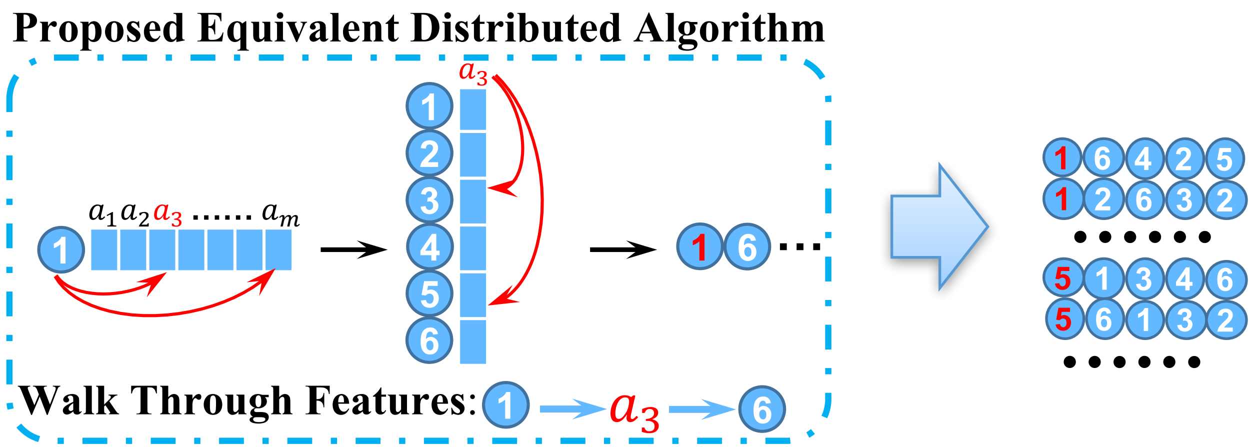
- **Instance proximity:** Similarities between instances defined by the features of instances, i.e., rows of each $\mathbf{X}^{(i)}$.
- Though $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, and $\mathbf{X}^{(3)}$ are heterogenous, the instance proximities learned from them are homogeneous.
- FeatWalk projects each instance proximity into a sequence of instance indices $\mathcal{Q}^{(i)}$, and learns $\mathbf{H}$ from $\{\mathcal{Q}^{(i)}\}$.

Intuitive Solution



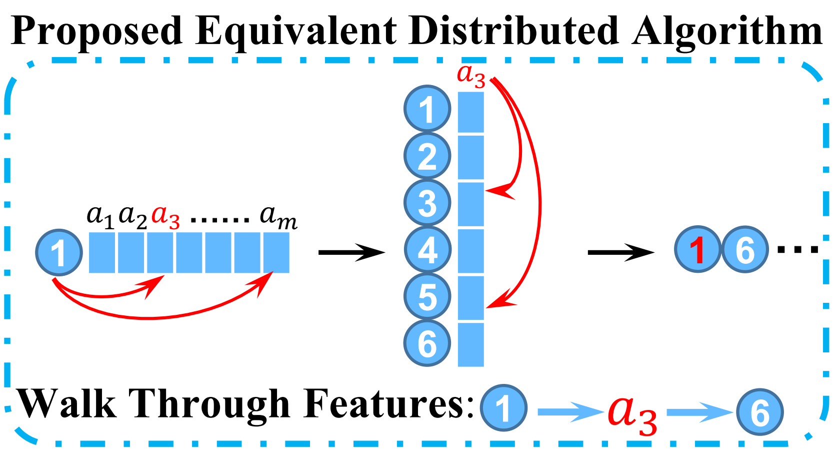
- To learn $\mathcal{Q}^{(i)}$, intuitive solution is to compute instance similarity matrix $\mathbf{S}$ based on $\mathrm{X}^{(i)}$, and perform random walks on $\mathbf{S}$.
  - Random Walks: In $\mathcal{Q}^{(i)}$, a sequence of instance indices, probability of i follows j approaches their similarity in $\mathbf{S}$.
- Expensive: $\mathbf{S}$ is dense with $\mathrm{n} \times n$ dimensions.

Equivalent to Similarity-based Random Walks



- FeatWalk has same results as the intuitive solution but avoid the computation of instance similarities **S**.
- **Theorem 1.** Probability of walking from i to j via FeatWalk is equal to the one via random walks on **S**, where
$$\mathbf{S} = \mathbf{Y}\mathbf{D}\mathbf{Y}^{\mathrm{T}}.$$
- **Y** is the feature matrix after two special normalizations.

FeatWalk Walks via Features



I. Given the initial (i), we walk to the m^{th} attribute category with probability

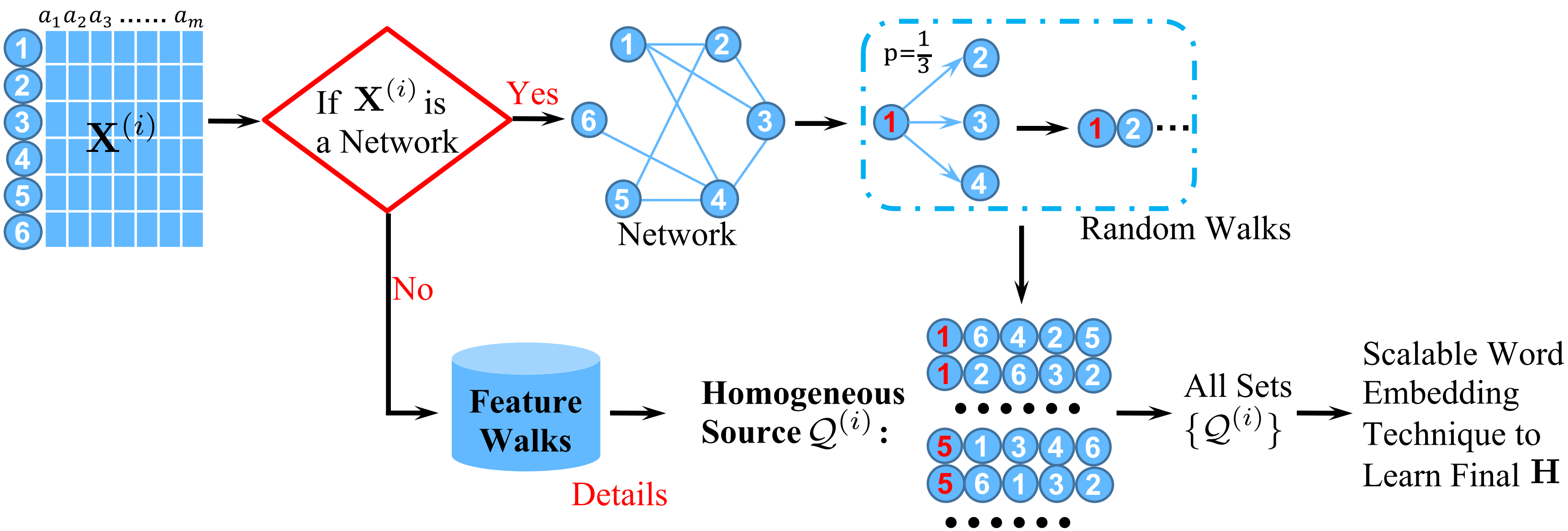
$$P(i \rightarrow a_m) = \frac{\hat{x}_{im}}{\sum_{p=1}^{M} \hat{x}_{ip}}.$$

II. Focus on the m^{th} attribute category and walk from a_m to (j) with probability

$$P(a_m \rightarrow j) = \frac{y_{jm}}{\sum_{n=1}^{N} y_{nm}}.$$

➢ $\hat{x}_{im}$ and y_{jm} are normalized instance features.

Strategies of FeatWalk



- FeatWalk projects each instance proximity into a sequence of instance indices $\mathcal{Q}^{(i)}$.
- Consider instance indices as words and sequences as sentences, a scalable word embedding technique is applied to all $\{\mathcal{Q}^{(i)}\}$ to learn a joint embedding representation $\mathbf{H}$.

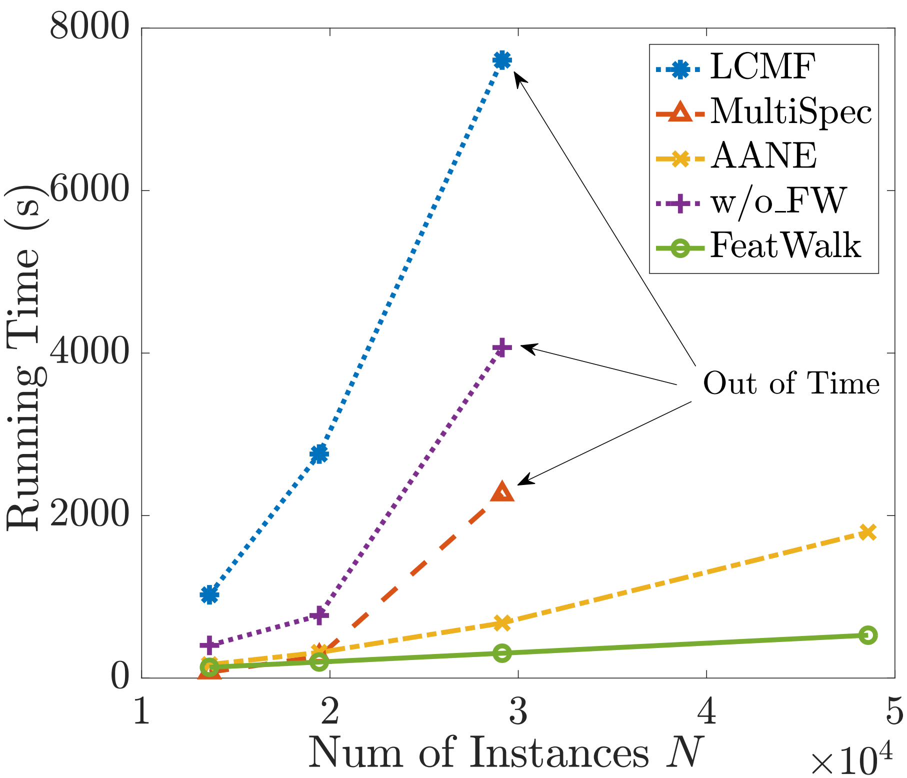
Experimental Settings

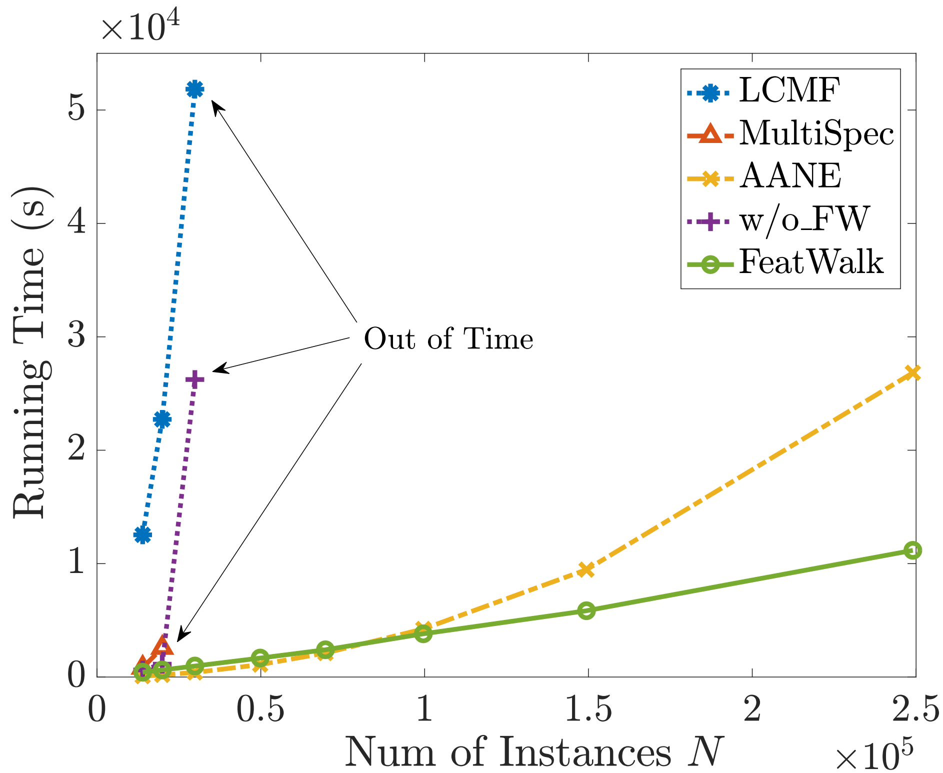
- Classification on four real-world datasets.
  - Reuters (18,758 documents)
  - Flickr (7,564 users)
  - ACM (48,579 papers)
  - Yelp (249,012 users, 1,779,803 edges, 20,000 feature categories, 47,216,356 entities)

- Three types of baselines.
  - Single feature embedding: NMF, Spectral, and FeatWalk_X
  - Network embedding: DeepWalk and LINE
  - Heterogeneous feature embedding: LCMF, MultiSpec, and AANE

Codes and datasets are available at http://people.tamu.edu/~xhuang/Code.html

Efficiency Evaluation





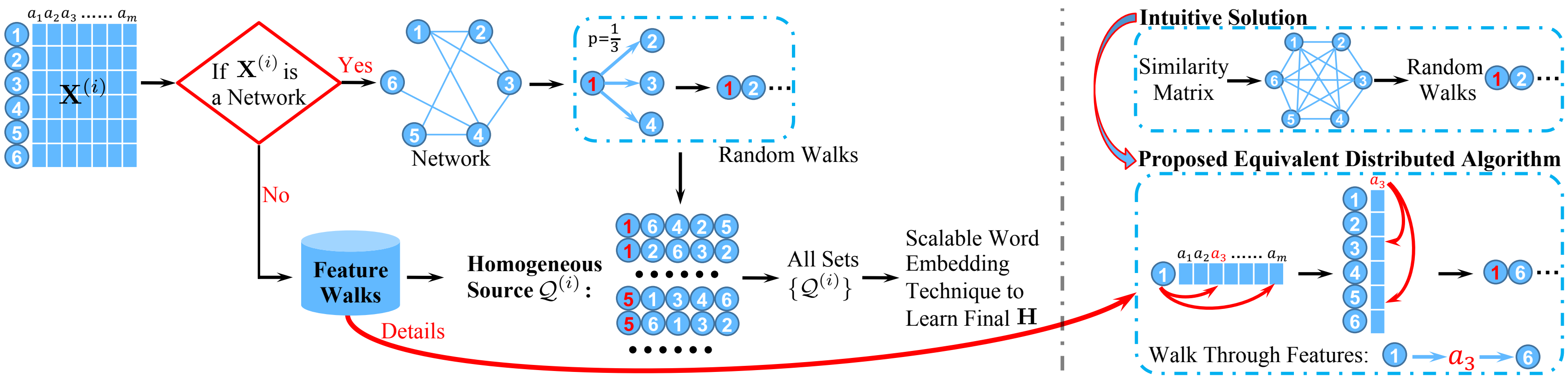
- Running time of FeatWalk is almost linear to N.
- FeatWalk achieves a significant acceleration compared to the intuitive solution w/o_FW.
- FeatWalk has the least running time when N is large.

Effectiveness Evaluation

	Flickr			ACM			Yelp-sub		
Training	25%	50%	100%	25%	50%	100%	25%	50%	100%
# Instances	3,026	4,538	7,564	19,432	29,147	48,579	19,921	29,881	49,802
NMF	0.629	0.718	0.773	0.653	0.660	0.664	0.680	0.686	0.688
Spectral	0.771	0.813	0.846	0.688	0.700	N.A.	0.683	N.A.	N.A.
FeatWalk_X	0.803	0.841	0.868	0.676	0.675	0.667	**0.701**	0.710	0.714
DeepWalk	0.373	0.465	0.535	0.576	0.630	0.684	0.310	0.318	0.350
LINE	0.332	0.421	0.516	0.549	0.624	0.693	0.243	0.264	0.294
LCMF	0.676	0.725	0.749	0.690	0.706	N.A.	0.680	0.686	N.A.
MultiSpec	0.720	0.800	0.859	0.709	0.719	N.A.	0.667	N.A.	N.A.
AANE	0.811	0.854	0.885	0.701	0.715	0.722	0.694	0.703	0.711
FeatWalk	**0.831**	**0.865**	**0.893**	**0.722**	**0.738**	**0.751**	0.700	**0.710**	**0.717**

- FeatWalk_X performs better than all single feature embedding and network embedding baselines.
- FeatWalk outperforms the state-of-the-art heterogeneous feature embedding baselines.

Conclusions



- Propose an effective framework FeatWalk to incorporate multiple types of high-dimensional instance features into a joint embedding representation.
- Design an efficient algorithm that avoids to compute similarity measure, and provides an alternative way to simulate the similarity-based random walks among instances to sample the local instance proximity.

Acknowledgement

- DATA Lab and collaborators

COMPUTER SCIENCE & ENGINEERING
TEXAS A&M UNIVERSITY

Data Analytics at Texas A&M (DATA Lab)

- Funding agencies
  - National Science Foundation
  - Defense Advanced Research Projects Agency

- Everyone attending the talk