Towards Incremental Construction of Graph Embeddings

Anton Tsitsulin
University of Bonn

Marina Munkhoeva
Skoltech

Davide Mottin
Aarhus University

Panagiotis Karras
Aarhus University

Ivan Oseledets
Skoltech

Lior Kamma
Aarhus University

Emmanuel Müller
University of Bonn

ABSTRACT

Low-dimensional representations, or *embeddings*, of a graph's nodes facilitate tasks such as link prediction and node classification. Embeddings learn similarities among nodes, either implicitly, as a side-effect of adapting word embedding methods to graphs, or explicitly, by reconstructing a similarity measure. As the similarity matrix is large and dense, past research has resorted to heuristic or linear factorization methods, compromising the solution quality.

In this paper we propose FREDE (*FRequent Directions Embeddings*), a nonlinear sketching-based embedding method that provides, at any iteration, an *error guarantee* that renders it practically indistinguishable from the optimal SVD factorization.

Starting out from the observation that embeddings strive to preserve the covariance among rows of the similarity matrix, FREDE operates on each row individually, progressively improving the overall embedding quality; thereby, it is the first, to our knowledge, embedding method that combines (i) linear space complexity, (ii) a nonlinear transform as its basis, and (iii) nontrivial error guarantees. Our experimental evaluation on variably sized networks shows that FREDE performs almost as well as SVD and clearly outperforms previous embedding algorithms in various data mining tasks.

ACM Reference Format:
Anton Tsitsulin, Marina Munkhoeva, Davide Mottin, Panagiotis Karras, Ivan Oseledets, Lior Kamma, and Emmanuel Müller. 2019. Towards Incremental Construction of Graph Embeddings . In *DLG'19: The First International Workshop on Deep Learning on Graphs: Methods and Applications, August 5, 2019, Anchorage, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1234/56789012.345678

1 INTRODUCTION

Graph embeddings [5, 13] empower data practitioners with a multi-purpose tool for effectively solving different graph tasks, such as node classification and link prediction. Neural graph embeddings [5, 13, 17, 19] computing *nonlinear* transformations via unsupervised learning outperform their linear counterparts [11, 21] in quality.

Recently, NetMF [14] established a connection between neural graph embeddings and the factorization of a matrix of nonlinear pairwise similarities among nodes, under certain conditions on the algorithm parameters. NetMF performs Singular Value Decomposition (SVD) on a dense similarity matrix. Contrary to neural methods such as DeepWalk and node2vec, NetMF achieves the global optimum of its objective function by virtue of the SVD.

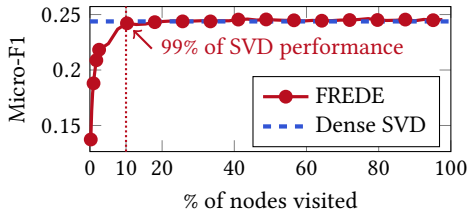

Figure 1: FREDE reaches 99% of full SVD performance whilst processing only 10% of a similarity matrix on PPI dataset.

In this paper, we propose FREDE, an *incremental, linear-space* algorithm that produces embeddings with *error guarantees*. We observe that embeddings seek to preserve the covariance among embeddings of nodes; we thus adapt the *Frequent Directions (FD)* algorithm [4, 8] to *sketch* a similarity matrix on per-row basis. FREDE provides error guarantees even after it has accessed a subset of similarity matrix rows. Figure 1 shows that it already achieves the same accuracy as SVD after it has only processed 10% of rows. We summarize our contributions as follows:

- we show that matrix factorization based embedding methods only need optimize for the covariance of embeddings;
- we propose FREDE, a novel graph embedding method that directly minimizes covariance error via *sketching* and is much lighter than matrix-factorization-based methods.
- we design FREDE as an *anytime* algorithm with time complexity linear in the number of processed rows;
- we confirm that FREDE is competitive against the state-of-the-art graph embedding methods.

	Algorithm		**Complexity**	
method	Nonlinear	Error bounds	Space	Time
DeepWalk [13]	✔	✘	$O(dn)$	$O(dn \log n)$
NODE2VEC [5]			$O(n^3)$	$O(dnb)$
LINE [17]	✔	✘	$O(dn)$	$O(dnb)$
HOPE [11]	✘	✔	$O(dn)$	$O(d^2m)$
AROPE [21]	✘	✘	$O(dn)$	$O(dm + d^2n)$
VERSE [19]	✔	✘	$O(dn)$	$O(dnb)$
NETMF [14]	✔	✔	$O(n^2)$	$O(dn^2)$
FREDE (ours)	✔	✔	$O(dn)$	$O(dn^2)$

Table 1: Comparison with previous work in terms of fullfilled (✔) and missing (✘) properties; complexities in terms of the dimensionality d, number of nodes n, edges m, and negative samples b.

DLG'19, August 5, 2019, Anchorage, USA

ACM ISBN 123-4-567-8901-2/34/56…$15.00
https://doi.org/10.1234/56789012.345678

2 RELATED WORK

We focus on embedding nodes of simple graphs with no additional information on nodes or edges. We discuss two major categories in which related work belongs to in terms of their desirable properties:

- **nonlinearity:** applying nonlinear transformations; most embedding methods except HOPE [11] and AROPE [21] are nonlinear.
- **error-boundness:** returning a solution with optimal or non-trivial error guarantees; closed-form methods are error-bounded, except AROPE [21] that abandons such guarantees for scalability.

Neural graph embeddings. Advances in natural language processing [6, 10] provided scalable methods that derive vector representations of words. DeepWalk [13] imported such methods to graphs by materializing a corpus of random walks, treating nodes encountered on the walk as words in a text. LINE [17] extended DeepWalk by exploiting graph edges rather than random walks; Node2vec [5] boosted it with a customizable generation of random walks; and VERSE [19] generalized it to a method that preserves any similarity measure among nodes, with Personalized PageRank [12] as the default option. Such neural graph embeddings reach scalability through stochastic gradient descent and sampling; they provide no closed-form solution, and hence do not offer any comprehensible quality guarantees either.

Matrix factorization embeddings. In another vein than neural embeddings, matrix factorization relies on the explicit decomposition of similarity matrices among nodes. GraRep [2] factorizes, by Singular Value Decomposition (SVD), the concatenation of dense log-transformed DeepWalk transition probability matrices over different numbers of steps. GraRep is neither scalable ($O(n^3)$), nor interpretable, as the concatenation provides no guarantees on the resulting representation. HOPE [11] overcomes the scalability drawback by applying a generalized form of SVD on special similarity matrices in the form AB^{-1}; HOPE achieves optimality due to the SVD guarantees of the Eckart–Young–Mirsky theorem, but its overall performance is hindered by its linear nature [19]. AROPE [21] applies eigenvalue filtering on symmetric similarity matrices; it forfeits the theoretical guarantees for the sake of performance.

Connecting the neural and factorization worlds. NetMF [14] extended an analysis of word embeddings [6] to suggest a connection between matrix factorization and neural graph embeddings: under certain probability independence assumptions, DeepWalk, LINE, and node2vec implicitly apply SVD on dense log-transformed similarity matrices. NetMF proposes novel closed-form solutions to compute such similarity matrices directly with optimal error guarantees. However, it yields a quadratic space complexity that hinders its application to large graphs with more than 100 000 nodes.

3 PRELIMINARIES AND PROBLEM SETTING

Here, we show that graph embedding methods implicitly minimize the covariance error in relation to a similarity matrix among graph nodes. Arguably, this interpretation is more economic than explaining embeddings in terms of minimizing reconstruction error [14]. Then, in Section 4, we describe FREDE, a novel, incremental algorithm that offers linear space complexity and error guarantees through covariance sketching.

3.1 Problem setting and notation

A *graph* is a pair $G = (V, E)$ with n vertices $V = (v_1, \ldots, v_n)$, $|V| = n$, and edges $E \subseteq V \times V$, $|E| = m$. A graph is represented by an *adjacency matrix* A for which $\mathrm{A}_{ij} = 1$ if $(i, j) \in E$ is an edge between node i and node j, otherwise $\mathrm{A}_{ij} = 0$. The matrix D is the diagonal matrix with the degree of node i as entry D_{ii}, i.e, $D_{ii} = \sum_{j=1}^{n} A_{ij}$. The *normalized adjacency matrix* is the matrix $\mathrm{P} = \mathrm{D}^{-1}\mathrm{A}$ that represents the transition probability from one node to any of its neighbors. We represent arbitrary interactions among nodes with a *similarity matrix* S, as done in previous work [11, 19, 21]. The row i of a matrix A is denoted as A_i.

We seek to find a d-dimensional embedding, represented as a $n \times d$ matrix W that provably retains most of the information in S.

3.2 Graph embeddings as matrix factorization

We now cast the problem of learning graph embeddings as *matrix factorization*. We first introduce the problem of factorization by minimization of the approximation error to pave the way on the study of the covariance error and its properties.

One way to preserve the similarities contained in the matrix S is to find an approximate matrix $\tilde{\mathrm{S}}$ that minimizes the *reconstruction error* [11, 14, 21].

Definition 1 (Reconstruction error). *The* reconstruction error *between* S *and* $\tilde{\mathrm{S}}$ *is the Frobenius norm of the difference among the* S *and* $\tilde{\mathrm{S}}$*, i.e.,* $\|\mathrm{S} - \tilde{\mathrm{S}}\|_F^2 = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} (\mathrm{S}_{ij} - \tilde{\mathrm{S}}_{ij})^2}$.

In other words, the reconstruction error acts element-wise and discards row dependencies.

In the case of *symmetric* S, there exists an eigendecomposition $\mathrm{S} = \mathrm{U}\Lambda\mathrm{U}^\top$. The optimal rank-$k$ approximation $[\mathrm{S}]_k = \mathrm{W}\mathrm{W}^\top$ of S minimizing the reconstruction error is $\mathrm{W} = \mathrm{U}_{:k}\sqrt{\Lambda_{:k}}$, obtained by the product of the first k eigenvectors $\mathrm{U}_{:k}$ and a diagonal matrix of square roots of the first k eigenvalues $\Lambda_{:k}$.

In the *non-symmetric* case, the best rank-k approximation can be obtained by taking the first k singular vectors and values $[\mathrm{S}]_k = \mathrm{U}_{:k}\Sigma_{:k}\mathrm{V}_{:k}^\top$ on the Singular Value Decomposition $\mathrm{S} = U\Sigma V^\top$.

The analysis in [14], for instance, shows that Deepwalk [13] objective is equivalent to SVD on the dense similarity matrix created from large powers of the adjacency matrix

$$\mathrm{S} = \log\left(\frac{|E|}{bT}\Big(\sum_{r=1}^{T}\mathrm{P}^r\Big)\mathrm{D}^{-1}\right),$$

where T is the window size for the random-walk and b is the number of negative samples. The $d-$dimensional Deepwalk embedding is the result of multiplying the d left singular vectors U_d by the square root of the first d singular values Σ_d, i.e., $\mathrm{U}_d\sqrt{\Sigma_d}$. The major shortcoming of such approach is the $O(n^2)$ **memory requirement** to store the dense similarity matrix S.

One solution to the scalability challenge is to apply SVD on the similarity matrix directly [11, 18] discarding the nonlinearity; however, that incurs a significant quality loss [19].

3.3 Matrix sketching

As the non-linear matrix factorization can not eschew its quadratic memory complexity, we turn our attention to an alternative approach, namely *matrix sketching*. Matrix sketching aims at a low-dimensional *sketch* of a matrix S that retains most of the information in S without striving for matrix reconstruction. One representative objective for matrix sketching is *covariance:*

Definition 2 (Covariance error). *The* covariance error *is the normalized difference between covariance matrices*

$$\mathsf{ce}_k(\mathrm{S},\mathrm{W}) = \frac{\|\mathrm{S}^\top\mathrm{S} - \mathrm{W}^\top\mathrm{W}\|_2}{\|\mathrm{S}-[\mathrm{S}]_k\|_F^2} \leq \frac{\|\mathrm{S}^\top\mathrm{S} - \mathrm{W}^\top\mathrm{W}\|_2}{\|\mathrm{S}\|_F^2} = \mathsf{ce}(\mathrm{S},\mathrm{W}).$$

Many sketching techniques allow *row-wise* processing of the matrix S with guarantees on the quality of the covariance. This is a valuable property as it allows to build the sketch incrementally (Section 4.3).

Frequent Directions [4, 8] sketches a matrix by iteratively filling the sketch with the incoming rows, performing SVD on the sketch when the sketch cannot add more rows, and shrinking the accumulated vectors with a low-rank SVD approximation. This allows incremental computation with bounded covariance error $\mathsf{ce} \leq \epsilon$ with setting $d = O(n/\epsilon)$.

4 INCREMENTAL GRAPH EMBEDDINGS

We show that SVD-based embedding methods aim to preserve covariance; this fact suggests that we could apply a sketching algorithm as an embedding method. Still, it would be inefficient to apply sketching algorithms to previously introduced dense similarity matrices that cannot be partially materialized. Instead, we use the PPR matrix as S and incrementally construct a graph embedding aiming to preserve the covariance of S.

4.1 Embedding as covariance optimization

Here we show that SVD based methods (notably, HOPE and NetMF) instead of preserving reconstruction loss aim to preserve covariance by dropping either the left or right singular vectors. First note that the columns of U are eigenvectors of the *covariance* matrix $\mathrm{SS}^\top = \mathrm{U}\Sigma^2\mathrm{U}^\top$, while the columns of V are eigenvectors of $\mathrm{S}^\top\mathrm{S} = \mathrm{V}\Sigma^2\mathrm{V}^\top$. Methods like HOPE, NetMF and GraRep have optimization objectives expressed as:

$$\min_{\mathrm{W}} \frac{\|\mathrm{SS}^\top - \mathrm{WW}^\top\|_2}{\|\mathrm{S}-[\mathrm{S}]_k\|_F^2}.$$

Since the similarity matrix is a square matrix, both rows and columns describe nodes. In what follows we apply sketching algorithms that optimize the *covariance among columns*,

$$\min_{\mathrm{W}} \frac{\|\mathrm{S}^\top\mathrm{S} - \mathrm{W}^\top\mathrm{W}\|_2}{\|\mathrm{S}-[\mathrm{S}]_k\|_F^2}.$$

When the similarity matrix is symmetric, as in AROPE, the two formulations are equivalent. The unnormalized form of the above objective is the PIP similarity [20], i.e., the unnormalized covariance error of observations defined over rows: $\|\mathrm{S}^\top\mathrm{S} - \mathrm{W}^\top\mathrm{W}\|_2$. While PIP was defined merely as a similarity measure between embeddings, we upgrade it to an optimization objective and show that previous work explicitly optimizes for it.

4.2 Effective and efficient similarity matrices

Previous work [11, 19] pointed out that embeddings based on higher-order similarity matrices achieve good performance in downstream tasks. VERSE [19] suggests using Personalized PageRank (PPR), which encodes higher-order similarity information and allows efficient row-wise computation.

Definition 3. *Given a starting node distribution* s*, damping factor* α*, and the transition probability matrix* P*, the Personalized PageRank vector* $\mathrm{PPR}_{i\cdot}$ *is defined by the recursive equation:*

$$\mathrm{PPR}_{i\cdot} = \alpha s + (1-\alpha)\mathrm{PPR}_{i\cdot}^\top\mathrm{P} \tag{1}$$

Following [6, 14] we prove that, under mild assumptions, VERSE equipped with the PPR similarity implicitly factorizes the log(PPR) matrix up to an additive constant.

Theorem 4.1. *VERSE implicitly factorizes a matrix of the form*

$$\log(\mathrm{PPR}) + \log n - \log b = \mathrm{XX}^\top. \tag{2}$$

Proof. Consider the VERSE objective function for the uniform sampling distribution and PPR similarity:

$$\mathcal{L} = \sum_{i=1}^{n}\sum_{j=1}^{n}\left[\mathrm{PPR}_{ij}\log\sigma(\mathbf{x}_i^\top\mathbf{x}_j) + b\mathbb{E}_{j\prime\sim Q_i}\log\sigma(-\mathbf{x}_i^\top\mathbf{x}_{j\prime})\right]. \tag{3}$$

where $\sigma(x) = (1+e^{-x})^{-1}$ is the sigmoid and Q_i is the noise sample distribution. Since PPR is right-stochastic and Q_i is uniform, i.e. $\Pr(Q_i = j) = \frac{1}{n}$, we separate two terms in Eq. 3:

$$\mathcal{L} = \sum_{i=1}^{n}\sum_{j=1}^{n}\mathrm{PPR}_{ij}\log\sigma(\mathbf{x}_i^\top\mathbf{x}_j) + \frac{b}{n}\sum_{i=1}^{n}\sum_{j\prime=1}^{n}\log\sigma(-\mathbf{x}_i^\top\mathbf{x}_{j\prime}).$$

Writing down an individual loss term for vertices i and j,

$$\mathcal{L}_{ij} = \mathrm{PPR}_{ij}\log\sigma(\mathbf{x}_i^\top\mathbf{x}_j) + \frac{b}{n}\log\sigma(-\mathbf{x}_i^\top\mathbf{x}_j).$$

We now substitute $z_{ij} = \mathbf{x}_i^\top\mathbf{x}_j$ and, following [6, 14], we operate under the assumption that individual z_{ij} terms are independent given a sufficiently large embedding dimension. Taking the derivative with respect to z_{ij}, we solve for the condition

$$\frac{\partial\mathcal{L}_{ij}}{\partial z_{ij}} = \mathrm{PPR}_{ij}\log\sigma(-z_{ij}) + \frac{b}{n}\log\sigma(z_{ij}) = 0,$$

to obtain the solution $z_{ij} = \log\frac{n\cdot\mathrm{PPR}_{ij}}{b} = \mathbf{x}_i^\top\mathbf{x}_j$ in the real domain. In matrix form, that is

$$\log(\mathrm{PPR}) + \log n - \log b = \mathrm{XX}^\top.$$

□

Note that a solution is algebraically impossible, as it implies approximating a non-symmetric matrix by a symmetric one. Thus, Equation (2) is foreordained to fail — it does not ultimately explain what a VERSE embedding does. Still, it provides a matrix whose covariance we can sketch efficiently in row-streaming fashion.

Equipped with the right algorithms and similarity, we can now introduce FREDE, the first incremental graph embedding algorithm.

Algorithm 1 FREDE algorithm

function FREDE(G, n, d)
  $\mathrm{W} \leftarrow \texttt{zeros}(2d, n)$ ▷ all zeros matrix $\mathrm{W} \in \mathrm{R}^{2d \times n}$
  $\hat{\Sigma} \leftarrow \mathrm{I}(2d)$ ▷ identity matrix $\hat{\Sigma} \in \mathrm{R}^{2d \times 2d}$
  for $v \in V$ **do**
    $x \leftarrow \texttt{PersonalizedPageRank}(v)$
    $x \leftarrow \log x + \log n$ ▷ VERSE similarity row
    Insert x into the last zero valued row of W
    if W has no zero valued rows **then**
      $\mathrm{U}, \Sigma, \mathrm{V}^\top \leftarrow \mathsf{SVD}(\hat{\Sigma}\mathrm{W}), \sigma \leftarrow \Sigma_{d,d}$
      $\hat{\Sigma} \leftarrow \sqrt{\mathsf{ReLU}(\Sigma^2 - \sigma^2 \mathrm{I}_d)}$ ▷ $\mathsf{ReLU}(\cdot) = \max(\cdot, 0)$
      $\hat{\Sigma}_{d:} \leftarrow \mathrm{I}_d$
      $\mathrm{W} \leftarrow \mathrm{V}^\top$
  return $\hat{\Sigma}, \mathrm{W}[:d, :]$
function GetEmbedding($k \leq d$) ▷ Anytime
  return $\hat{\Sigma}^{1/2}\mathrm{W}^\top[:k, :]$ ▷ first k rows

4.3 FREDE algorithm

FREDE lowers the $O(d^2 n)$ retrieval time and obtains a more accurate covariance estimation as follows. The matrix $\mathrm{X}\mathrm{X}^\top$ has equal row and column ranks, hence we can rewrite the decomposition commutatively, as $\log(\mathrm{PPR}) + \log n - \log b = \mathrm{X}^\top \mathrm{X}$. Then, we treat the rows of the PPR matrix as a column basis for our embedding ($n \times d$), and apply the row-stream sketching algorithm described in Section 3.3.

Algorithm 1 presents the details of FREDE; it passes through the rows of the PPR matrix, applies Frequent Directions as a sketch, and conveniently keeps track of singular values so to avoid recalculations at the time of output.

Quite significantly, FREDE can produce valid embeddings after it has processed only a part of the graph. Since the bounds on the covariance error still hold for the subset of the processed rows, the sketch embedding maintains optimality guarantees as it progresses through the input matrix.

The weighting of singular values in embedding methods has been discussed in [7, 16]. It has been shown that raising singular values to a value from the interval $[0, 0.5]$ is beneficial for the downstream tasks, although not explained theoretically. In line with previous work, we use the square root of singular values for the embeddings.

5 EXPERIMENTS

We evaluate FREDE against state-of-the-art graph embedding algorithms, NetMF [14], and SVD exact matrix factorization. We run all experiments on commodity servers with 20 cores and 384Gb RAM. For each dataset, we repeat the experiment 10 times and report the average among the results.

Competing methods and baselines. We evaluate FREDE against the following state-of-the-art graph embedding methods:

- DeepWalk [13]: This approach learns an embedding by sampling fixed-length random walks from each node and applying word2vec-based learning on those walks. We use the default parameters described in the paper, i.e., walk length $t = 80$, number of walks per node $\gamma = 80$, and window size $T = 10$.

	Size			Statistics	
dataset	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert\mathcal{L}\rvert$	Avg. deg.	Density
PPI	4k	77k	50	19.9	5.1×10^{-3}
POS	5k	185k	40	38.7	8.1×10^{-3}
BlogCatalog	10k	334k	39	64.8	6.3×10^{-3}
CoAuthor	52k	178k	—	6.94	1.3×10^{-4}
VK	79k	2.7M	—	34.1	8.7×10^{-4}

Table 2: Dataset characteristics: number of vertices $|V|$, number of edges $|E|$; number of node labels $|\mathcal{L}|$; average node degree; density defined as $|E|/\binom{|V|}{2}$.

- VERSE [19]: This approach learns similarity measures via sampling by optimizing a single-layer neural network. We use personalized PageRank as similarity measure with the default parameters described in the paper, i.e., $\alpha = 0.85$ and nsamples= 10^6.
- NetMF [14]: This approach computes the DeepWalk matrix with a closed-form solution and performs SVD on that matrix. To compare NetMF on quality, we use their optimal method NetMF-small, as NetMF-large is heuristic and considerably slower than DeepWalk. We use the same parameters as in DeepWalk – window size $T = 10$, and set bias $b = 1$ as in the original paper.
- SVD: full SVD decomposition of the precomputed PPR similarity matrix. We use the same PPR settings as VERSE, namely $\alpha = 0.85$.

Parameter settings. We set default embedding dimension $d = 128$ for all the methods unless indicated otherwise [5, 13], and use $\alpha = 0.85$ for personalized PageRank for the default value [12, 19]. We use Intel MKL library to perform SVD using the function `gesdd`. For classification we use LIBLINEAR [3]. We repeat each experiment 10 times and evaluate each embedding 10 times in order to reduce the noise introduced by the embedding and classification processes.

Datasets. We assess our methods on 4 real datasets; the data characteristics are described in Table 2.

- PPI [5, 15]: a protein-protein interaction dataset, where labels represent hallmark gene sets of specific biological states.
- POS [5, 9]: a word co-occurrence network built from Wikipedia data. The labels indicate parts of speech (POS) induced by Stanford NLP parser.
- MS citation graph [1, 19]: a paper citation graph generated from Microsoft Academic graph. We extract the induced graph of papers published in the 15 major data mining conferences and use the conference identifiers as labels.
- VK [19]: a Russian all-encompassing social network. Labels represent self-identified sex of users. The graph is available as two snapshots: one made in November 2016 and one in May 2017. We predict the links that appeared in this timeframe.

5.1 Sketching quality

The optimal rank-k covariance approximation can be obtained via SVD on the full similarity matrix $\tilde{\mathrm{S}}^\top \tilde{\mathrm{S}} = \mathrm{V}_d \Sigma_d^2 \mathrm{V}_d^\top$. We numerically compute the covariance using the definition in Section 4.1 on the smallest dataset, PPI. Figure 2 reports the covariance error (ce) as a function of the dimension d. FREDE approaches the optimal SVD solution as the dimensionality d increases.

Operator	Result
Average	$(\mathbf{a}+\mathbf{b})/2$
Concat	$[\mathbf{a}_1,\ldots,\mathbf{a}_d,\mathbf{b}_1,\ldots,\mathbf{b}_d]$
Hadamard	$[\mathbf{a}_1*\mathbf{b}_1,\ldots,\mathbf{a}_d*\mathbf{b}_d]$
Weighted L1	$[\lvert\mathbf{a}_1-\mathbf{b}_1\rvert,\ldots,\lvert\mathbf{a}_d-\mathbf{b}_d\rvert]$
Weighted L2	$[(\mathbf{a}_1-\mathbf{b}_1)^2,\ldots,(\mathbf{a}_d-\mathbf{b}_d)^2]$

Table 3: Edge embedding strategies for link prediction task for nodes $u, v \in V$ and corresponding embeddings $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$.

5.2 Node classification

We now turn our attention on the quality of FREDE on node classification task. In a partially labelled graph, node classification aims at predicting the correct labels for the unlabelled nodes. We report results for all the methods on PPI, POS, and BlogCatalog graphs.

We evaluate the accuracy in terms of Micro-F1 measure as it is common in the literature [13, 17]. Tables 4, 5, and 6 report the results with the different datasets.

Our results confirm the superiority of our similarity matrix choice over NetMF across the tested datasets. Surprisingly, FREDE outperforms its exact solution given by SVD on two out of three datasets tested. We attribute that to the fact that the sketching can be seen as iterative filtering of the data, allowing us to reduce the noise.

5.3 Link prediction

Link prediction is the task of predicting the appearance of a link between pairs of nodes in a graph. We evaluate link prediction on two datasets – CoCit and VK (Tables 7, 8).

As a baseline, we train a logistic regression classifier on traditional link prediction features (node degree, number of common neighbors, Adamic-Adar index, Jaccard coefficient, and preferential attachment index). Features are constructed from embeddings according to the rules outlined in the Table 3. We then run a logistic regression classifier to distinguish between links that will appear in the graph versus those that will not. Negative sampling of the non appearing links is used for them to be equally represented in the training data. We use 50% of links for training and remaining 50% for testing. FREDE outperforms all competing methods on CoCit.

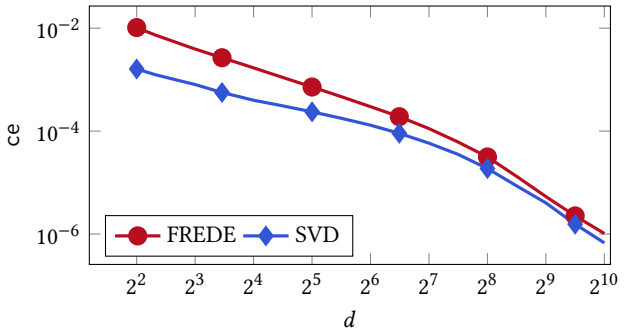

Figure 2: Covariance error wrt. embedding dimensionality d.

	labelled nodes, %				
method	10%	30%	50%	70%	90%
DeepWalk	16.33	19.74	21.34	22.39	23.38
NetMF	18.58	22.01	23.87	24.65	25.30
VERSE	16.45	19.89	21.64	23.08	23.84
FREDE	19.56	23.11	24.38	25.11	25.52
SVD	18.31	22.12	23.66	25.03	25.78

Table 4: Micro-F1 classification results in PPI dataset.

	labelled nodes, %				
method	10%	30%	50%	70%	90%
DeepWalk	43.42	47.12	48.96	49.86	50.18
NetMF	43.42	46.98	48.52	49.23	49.72
VERSE	40.80	44.70	46.60	47.65	48.24
FREDE	46.59	49.23	50.45	51.02	51.30
SVD	44.69	48.86	50.57	51.53	52.20

Table 5: Micro-F1 classification results in POS dataset.

	labelled nodes, %				
method	10%	30%	50%	70%	90%
DeepWalk	36.22	39.84	41.22	42.06	42.53
NetMF	36.62	39.80	41.05	41.70	42.17
VERSE	35.82	40.06	41.63	42.63	43.14
FREDE	35.69	38.88	39.98	40.54	40.75
SVD	37.60	40.99	42.10	42.66	43.47

Table 6: Micro-F1 classification results in BlogCatalog dataset.

method	Average	Concat	Hadamard	L1	L2
DeepWalk	68.97	68.43	66.61	78.80	77.89
VERSE	79.62	79.25	86.27	75.15	75.32
FREDE	81.28	80.95	86.83	81.70	82.37
Baseline			77.53		

Table 7: Link prediction results on the MS coauthorship graph. Best results per method are underlined.

method	Average	Concat	Hadamard	L1	L2
DeepWalk	69.98	69.83	69.56	78.42	77.42
VERSE	74.56	74.42	80.94	77.16	77.47
FREDE	74.68	74.59	77.63	74.25	73.60
Baseline			78.84		

Table 8: Link prediction results on the VK social graph. Best results per method are underlined.

6 CONCLUSION

We introduced FREDE, an incremental graph embedding algorithm that represents a nonlinear similarity matrix among nodes optimally with respect to an objective close to the optimal low-rank approximation by SVD. We have shown that previous graph embedding methods aim to preserve the covariance (i.e., dot-product among rows) of the similarity matrix. We took this analysis to its logical conclusion by adapting row-wise matrix sketching to graph embeddings, achieving both scalability and a closed-form optimal solution for covariance preservation.

Our experiments show that FREDE outperforms previous embedding methods on graph mining tasks, while it fares comparably to SVD in covariance preservation, and better in node classification thanks to its sketch-based nature that filters out noise in the data.

REFERENCES

[1] 2016. Microsoft Academic Graph - KDD cup 2016. https://kddcup2016.azurewebsites.net/Data.

[2] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning Graph Representations with Global Structural Information. In *CIKM*.

[3] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of machine learning research* (2008).

[4] Mina Ghashami, Edo Liberty, Jeff M. Phillips, and David P. Woodruff. 2016. Frequent Directions : Simple and Deterministic Matrix Sketching. *SIAM J. Comput.* 45 (2016), 1762–1792.

[5] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *KDD*.

[6] Omer Levy and Yoav Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *NIPS*.

[7] Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics* 3 (2015), 211–225.

[8] Edo Liberty. 2013. Simple and deterministic matrix sketching. In *KDD*.

[9] Matt Mahoney. 2011. Large text compression benchmark. http://www.mattmahoney.net/text/text.html.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS*.

[11] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric Transitivity Preserving Graph Embedding. In *KDD*.

[12] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The PageRank citation ranking: bringing order to the web. (1999).

[13] Bryan Perozzi, Rami Al-Rfou', and Steven Skiena. 2014. DeepWalk: online learning of social representations. In *KDD*.

[14] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network Embedding as Matrix Factorization: Unifying DeepWalk, LINE, PTE, and node2vec. In *WSDM*.

[15] Chris Stark, Bobby-Joe Breitkreutz, Teresa Reguly, Lorrie Boucher, Ashton Breitkreutz, and Mike Tyers. 2006. BioGRID: a general repository for interaction datasets. *Nucleic Acids Research* (2006).

[16] Karl Stratos, Michael Collins, and Daniel Hsu. 2015. Model-based word embeddings from decompositions of count matrices. In *ACL*, Vol. 1. 1282–1291.

[17] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *WWW*.

[18] Mikkel Thorup and Uri Zwick. 2005. Approximate distance oracles. *JACM* 52, 1 (2005), 1–24.

[19] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, and Emmanuel Müller. 2018. VERSE: Versatile Graph Embeddings from Similarity Measures. In *WWW*.

[20] Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. In *NeurIPS*.

[21] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. 2018. Arbitrary-Order Proximity Preserved Network Embedding. In *KDD*.