Minimum-Distortion Embedding

Other titles in Foundations and Trends® in Machine Learning

Graph Kernels: State-of-the-Art and Future Challenges
Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray and Bastian Rieck
ISBN: 978-1-68083-770-4

Data Analytics on Graphs Part III: Machine Learning on Graphs, from Graph Topology to Applications
Ljubiša Stanković, Danilo Mandic, Miloš Daković, Miloš Brajović, Bruno Scalzo, Shengxi Li and Anthony G. Constantinides
ISBN: 978-1-68083-982-16

Data Analytics on Graphs Part II: Signals on Graphso
Ljubiša Stanković, Danilo Mandic, Miloš Daković, Miloš Brajović, Bruno Scalzo, Shengxi Li and Anthony G. Constantinides
ISBN: 978-1-68083-982-1

Data Analytics on Graphs Part I: Graphs and Spectra on Graphs
Ljubiša Stanković, Danilo Mandic, Miloš Daković, Miloš Brajović, Bruno Scalzo, Shengxi Li and Anthony G. Constantinides
ISBN: 978-1-68083-982-1

Minimum-Distortion Embedding

Akshay Agrawal
Stanford University
akshayka@cs.stanford.edu

Alnur Ali
Stanford University
alnurali@stanford.edu

Stephen Boyd
Stanford University
boyd@stanford.edu

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Machine Learning

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

A. Agrawal, A. Ali, and S. Boyd. *Minimum-Distortion Embedding*. Foundations and Trends® in Machine Learning, vol. 14, no. 3, pp. 211–378, 2021.

ISBN: 978-1-68083-889-3

Foundations and Trends® in Machine Learning
Volume 14, Issue 3, 2021

Editorial Board

Editorial Scope

Topics

Foundations and Trends® in Machine Learning publishes survey and tutorial articles in the following topics:

- Adaptive control and signal processing
- Applications and case studies
- Behavioral, cognitive and neural learning
- Bayesian learning
- Classification and prediction
- Clustering
- Data mining
- Dimensionality reduction
- Evaluation
- Game theoretic learning
- Graphical models
- Independent component analysis
- Inductive logic programming
- Kernel methods
- Markov chain Monte Carlo
- Model choice
- Nonparametric methods
- Online learning
- Optimization
- Reinforcement learning
- Relational learning
- Robustness
- Spectral methods
- Statistical learning theory
- Variational inference
- Visualization

Information for Librarians

Foundations and Trends® in Machine Learning, 2021, Volume 14, 6 issues. ISSN paper version 1935-8237. ISSN online version 1935-8245. Also available as a combined paper and online subscription.

Contents

Minimum-Distortion Embedding

Akshay Agrawal[1], Alnur Ali[2] and Stephen Boyd[3]

[1] *Stanford University; akshayka@cs.stanford.edu*
[2] *Stanford University; alnurali@stanford.edu*
[3] *Stanford University; boyd@stanford.edu*

ABSTRACT

We consider the vector embedding problem. We are given a finite set of items, with the goal of assigning a representative vector to each one, possibly under some constraints (such as the collection of vectors being standardized, *i.e.*, having zero mean and unit covariance). We are given data indicating that some pairs of items are similar, and optionally, some other pairs are dissimilar. For pairs of similar items, we want the corresponding vectors to be near each other, and for dissimilar pairs, we want the vectors to not be near each other, measured in Euclidean distance. We formalize this by introducing distortion functions, defined for some pairs of items. Our goal is to choose an embedding that minimizes the total distortion, subject to the constraints. We call this the *minimum-distortion embedding* (MDE) problem.

The MDE framework is simple but general. It includes a wide variety of specific embedding methods, such as spectral embedding, principal component analysis, multidimensional scaling, Euclidean distance problems, dimensionality reduction methods (like Isomap and UMAP), semi-supervised learning, sphere packing, force-directed layout, and others. It also includes new embeddings, and provides principled ways of validating or sanity-checking historical and new embeddings alike.

Akshay Agrawal, Alnur Ali and Stephen Boyd (2021), "Minimum-Distortion Embedding", Foundations and Trends® in Machine Learning: Vol. 14, No. 3, pp 211–378.
DOI: 10.1561/2200000090.

In a few special cases, MDE problems can be solved exactly. For others, we develop a projected quasi-Newton method that approximately minimizes the distortion and scales to very large data sets, while placing few assumptions on the distortion functions and constraints. This monograph is accompanied by an open-source Python package, PyMDE, for approximately solving MDE problems. Users can select from a library of distortion functions and constraints or specify custom ones, making it easy to rapidly experiment with new embeddings. Because our algorithm is scalable, and because PyMDE can exploit GPUs, our software scales to problems with millions of items and tens of millions of distortion functions. Additionally, PyMDE is competitive in runtime with specialized implementations of specific embedding methods. To demonstrate our method, we compute embeddings for several real-world data sets, including images, an academic co-author network, US county demographic data, and single-cell mRNA transcriptomes.

1

Introduction

An embedding of n items, labeled $1, \ldots, n$, is a function F mapping the set of items into $\mathbf{R}^m$. We refer to $x_i = F(i)$ as the embedding vector associated with item i. In applications, embeddings provide concrete numerical representations of otherwise abstract items, for use in downstream tasks. For example, a biologist might look for subfamilies of related cells by clustering embedding vectors associated with individual cells, while a machine learning practitioner might use vector representations of words as features for a classification task. Embeddings are also used for visualizing collections of items, with embedding dimension m equal to one, two, or three.

For an embedding to be useful, it should be faithful to the known relationships between items in some way. There are many ways to define faithfulness. A working definition of a faithful embedding is the following: if items i and j are similar, their associated vectors x_i and x_j should be near each other, as measured by the Euclidean distance $\|x_i - x_j\|_2$; if items i and j are dissimilar, x_i and x_j should be distant, or at least not close, in Euclidean distance. (Whether two items are similar or dissimilar depends on the application. For example two biological cells might be considered similar if some distance between their mRNA transcriptomes is small.) Many well-known embedding methods like principal component analysis (PCA), spectral embedding (Chung and Graham, 1997; Belkin and Niyogi, 2002), and multidimensional scaling (Torgerson, 1952; Kruskal, 1964a) use this basic notion of faithfulness, differing in how they make it precise.

The literature on embeddings is both vast and old. PCA originated over a century ago (Pearson, 1901), and it was further developed three decades later in the field of psychology (Hotelling, 1933; Eckart and Young, 1936). Multidimensional scaling, a family of methods for embedding items given dissimilarity scores or distances between items, was also developed in the field of psychology during the early-to-mid 20th century (Richardson, 1938; Torgerson, 1952; Kruskal, 1964a). Methods for embedding items that are vectors can be traced back to the early 1900s (Menger, 1928; Young and Householder, 1938), and more recently developed methods use tools from convex optimization and convex analysis (Biswas and Ye, 2004; Hayden *et al.*, 1991). In spectral clustering, an embedding based on an eigenvector decomposition of the graph Laplacian is used to cluster graph vertices (Pothen *et al.*, 1990; von Luxburg, 2007). During this century, dozens of embedding methods have been developed for reducing the dimension of high-dimensional vector data, including Laplacian eigenmaps (Belkin and Niyogi, 2002), Isomap (Tenenbaum *et al.*, 2000), locally-linear embedding (LLE) (Roweis and Saul, 2000), stochastic neighborhood embedding (SNE) (Hinton and Roweis, 2003), t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008), LargeVis (Tang *et al.*, 2016) and uniform manifold approximation and projection (UMAP) (McInnes *et al.*, 2018). All these methods start with either weights describing the similarity of a pair of items, or distances describing their dissimilarity.

In this monograph we present a general framework for faithful embedding. The framework, which we call *minimum-distortion embedding* (MDE), generalizes the common cases in which similarities between items are described by weights or distances. It also includes most of the embedding methods mentioned above as special cases. In our formulation, for some pairs of items, we are given distortion functions of the Euclidean distance between the associated embedding vectors. Evaluating a distortion function at the Euclidean distance between the vectors gives the distortion of the embedding for a pair of items. The goal is to find an embedding that minimizes the total or average distortion, possibly subject to some constraints on the embedding. We focus on three specific constraints: a centering constraint, which requires the embedding to have mean zero, an anchoring constraint, which fixes the positions of a subset of the embedding vectors, and a standardization constraint, which requires the embedding to be centered and have identity covariance.

MDE problems are in general intractable, admitting efficiently computable (global) solutions only in a few special cases like PCA and spectral embedding. In most other cases, MDE problems can only be approximately solved, using heuristic methods. We develop one such heuristic, a projected quasi-Newton method. The method we describe works well for a variety of MDE problems.

This monograph is accompanied by an open-source implementation for specifying MDE problems and computing low-distortion embeddings. Our software package, PyMDE, makes it easy for practitioners to experiment with different embeddings via different choices of distortion functions and constraint sets. Our implementation scales to very large datasets and to embedding dimensions that are much larger than two or three. This means that our package can be used for both visualizing large amounts of data and generating features for downstream tasks. PyMDE supports GPU acceleration and automatic differentiation of distortion functions by using PyTorch (Paszke *et al.*, 2019) as the numerical backend.

A preview of our framework. Here we give a brief preview of the MDE framework, along with a simple example of an MDE problem. We discuss the MDE problem at length in Chapter 2.

An embedding can be represented concretely by a matrix $X \in \mathbf{R}^{n \times m}$, whose rows $x_1^T, \ldots, x_n^T \in \mathbf{R}^m$ are the embedding vectors. We use $\mathcal{E}$ to denote the set of pairs, and $f_{ij} : \mathbf{R}_+ \to \mathbf{R}$ to denote the distortion functions for $(i, j) \in \mathcal{E}$. Our goal is to find an embedding that minimizes the average distortion

$$E(X) = \frac{1}{|\mathcal{E}|} \sum_{(i,j) \in \mathcal{E}} f_{ij}(d_{ij}),$$

where $d_{ij} = \|x_i - x_j\|_2$, subject to constraints on the embedding, expressed as $X \in \mathcal{X}$, where $\mathcal{X} \subseteq \mathbf{R}^{n \times m}$ is the set of allowable embeddings. Thus the MDE problem is

$$\begin{array}{ll} \text{minimize} & E(X) \\ \text{subject to} & X \in \mathcal{X}. \end{array}$$

We solve this problem, sometimes approximately, to find an embedding.

An important example is the quadratic MDE problem with standardization constraint. In this problem the distortion functions are quadratic $f_{ij}(d_{ij}) = w_{ij} d_{ij}^2$, where $w_{ij} \in \mathbf{R}$ is a weight conveying similarity (when $w_{ij} > 0$) or dissimilarity (when $w_{ij} < 0$) of items i and j. We constrain the embedding X to be standardized, *i.e.*, it must satisfy $(1/n)X^T X = I$ and $X^T \mathbf{1} = 0$, which forces the embedding vectors to spread out. While most MDE problems are intractable, the quadratic MDE problem is an exception: it admits an analytical solution via eigenvectors of a certain matrix. Many well-known embedding methods, including PCA, spectral embedding, and classical multidimensional scaling, are instances of quadratic MDE problems, differing only in their choice of pairs and weights. Quadratic MDE problems are discussed in Chapter 3.

Why the Euclidean norm? A natural question is why we use the Euclidean norm as our distance measure between embedding vectors. First, when we are embedding into $\mathbf{R}^2$ or $\mathbf{R}^3$ for the purpose of visualization or discovery, the Euclidean distance corresponds to actual physical distance, making it a natural choice. Second, it is traditional, and follows a large number of known embedding methods like PCA and spectral embedding that also use Euclidean distance. Third, the standardization constraint we consider in this monograph has a natural interpretation when we use the Euclidean distance, but would make little sense if we used another metric. Finally, we mention that the local optimization methods described in this monograph can be easily extended to the case where distances between embedding vectors are measured with a non-Euclidean metric.

1.1 Contributions

The main contributions of this monograph are the following:

1. We present a simple framework, MDE, that unifies and generalizes many different embedding methods, both classical and modern. This framework makes it easier to interpret existing embedding methods and to create new ones. It also provides principled ways to validate, or at least sanity-check, embeddings.

2. We develop an algorithm for approximately solving MDE problems (*i.e.*, for computing embeddings) that places very few assumptions on the distortion functions and constraints. This algorithm reliably produces good embeddings in practice and scales to large problems.

3. We provide open-source software that makes it easy for users to solve their own MDE problems and obtain custom embeddings. Our implementation of our solution method is competitive in runtime to specialized algorithms for specific embedding methods.

1.2 Outline

This monograph is divided into three parts, I *Minimum-Distortion Embedding*, II *Algorithms*, and III *Examples*.

Part I: Minimum-distortion embedding. We begin Part I by describing the MDE problem and some of its properties in Chapter 2. We introduce the notion of anchored embeddings, in which some of the embedding vectors are fixed, and standardized embeddings, in which the embedding vectors are constrained to have zero mean and identity covariance. Standardized embeddings are favorably scaled for many tasks, such as for use as features for supervised learning.

In Chapter 3 we study MDE problems with quadratic distortion, focusing on the problems with a standardization constraint. This class of problems has an analytical solution via an eigenvector decomposition of a certain matrix. We show that many existing embedding methods, including spectral embedding, PCA, Isomap, kernel PCA, and others, reduce to solving instances of the quadratic MDE problem.

In Chapter 4 we describe examples of distortion functions, showing how different notions of faithfulness of an embedding can be captured by different distortion functions. Some choices of the distortion functions (and constraints) lead to MDE problems solved by well-known methods, while others yield MDE problems that, to the best of our knowledge, have not appeared elsewhere in the literature.

Part II: Algorithms. In Part II, we describe algorithms for computing embeddings. We begin by presenting stationarity conditions for the MDE problem in Chapter 5, which are necessary but not sufficient for an embedding to be optimal. The stationarity conditions have a simple form: the gradient of the average distortion, projected onto the set of tangents of the constraint set at the current point, is zero. This condition guides our development of algorithms for computing embeddings.

In Chapter 6, we present a projected quasi-Newton algorithm for approximately solving MDE problems. For very large problems, we additionally develop a stochastic proximal algorithm that uses the projected quasi-Newton algorithm to solve a sequence of smaller regularized MDE problems. Our algorithms can be applied to MDE problems with differentiable average distortion, and any constraint set for which there exists an efficient projection onto the set and an efficient projection onto the set of tangents of the constraint set at the current point. This includes MDE problems with centering, anchor, or standardization constraints.

In Chapter 7, we present numerical examples demonstrating the performance of our algorithms. We also describe a software implementation of these methods, and briefly describe our open-source implementation PyMDE.

Part III: Examples. In Part III, we use PyMDE to approximately solve many MDE problems involving real datasets, including images (Chapter 8), co-authorship networks (Chapter 9), United States county demographics (Chapter 10), population genetics (Chapter 11), and single-cell mRNA transcriptomes (Chapter 12).

1.3 Related work

Dimensionality reduction. In many applications, the original items are associated with high-dimensional vectors, and we can interpret the embedding into the smaller dimensional space as *dimensionality reduction.* Dimensionality reduction can be used to reduce the computational burden of numerical tasks, compared to carrying them out with the original high-dimensional vectors. When the embedding dimension is two or three, dimension reduction can also be used to visualize the original high-dimensional data and facilitate exploratory data analysis. For example, visualization is an important first step in studying single-cell mRNA transcriptomes, a relatively new type of data in which each cell is represented by a high-dimensional vector encoding gene expression (Sandberg, 2014; Kobak and Berens, 2019).

Dozens of methods have been developed for dimensionality reduction. PCA, the Laplacian eigenmap (Belkin and Niyogi, 2002), Isomap (Tenenbaum *et al.*, 2000), LLE (Roweis and Saul, 2000), maximum variance unfolding (Weinberger and Saul, 2004), t-SNE (Maaten and Hinton, 2008), LargeVis (Tang *et al.*, 2016), UMAP (McInnes *et al.*, 2018), and the latent variable model (LVM) from (Saul, 2020) are all dimensionality reduction methods. With the exception of t-SNE and the LVM, these methods can be interpreted as solving different MDE problems, as we will see in Chapters 3 and 4. We exclude t-SNE because its objective function is not separable in the embedding distances; however, methods like LargeVis and UMAP have been observed to produce embeddings that are similar to t-SNE embeddings (Böhm *et al.*, 2020). We exclude the LVM because it fits some additional parameters, in addition to the embedding.

Dimensionality reduction is sometimes called manifold learning in the machine learning community, since some of these methods can be motivated by a hypothesis that the original data lie in a low-dimensional manifold, which the dimensionality reduction method seeks to recover (Ma and Fu, 2011; Cayton, 2005; Lin and Zha, 2008; Wilson *et al.*, 2014; Nickel and Kiela, 2017).

Finally, we note that dimensionality reduction methods have been studied under general frameworks other than MDE (Ham *et al.*, 2004; Yan *et al.*, 2006; Kokiopoulou *et al.*, 2011; Lawrence, 2011; Wang *et al.*, 2020).

Metric embedding. Another well-studied class of embeddings are those that embed one finite metric space into another one. There are many ways to define the distortion of such an embedding. One common definition is the maximum fractional error between the embedding distances and original distances, across all pairs of items. (This can be done by insisting that the embedding be non-contractive, *i.e.*, the embedding distances are at least the original distances, and then minimizing the maximum ratio of embedding distance to original distance.)

An important result in metric embedding is the Johnson-Lindenstrauss Lemma, which states that a linear map can be used to reduce the dimension of vector data, scaling distances by no more than $(1 \pm \epsilon)$, when the target dimension m is $O(\log n/\epsilon^2)$ (Johnson and Lindenstrauss, 1984). Another important result is due to Bourgain, who showed that any finite metric can be embedded in Euclidean space with at most a logarithmic distortion (Bourgain, 1985). A constructive method via semidefinite programming was later developed (Linial *et al.*, 1995). Several other results, including impossibility results, have been discovered (Indyk *et al.*, 2017), and some recent research has focused on embedding into non-Euclidean spaces, such as hyperbolic space (Sala *et al.*, 2018).

In this monograph, for some of the problems we consider, all that is required is to place similar items near each other, and dissimilar items not near each other; in such applications we may not even have original distances to preserve. In other problems we do start with original distances. In all cases we are interested in minimizing an *average* of distortion functions (not maximum), which is more relevant in applications, especially since real-world data is noisy and may contain outliers.

Force-directed layout. Force-directed methods are algorithms for drawing graphs in the plane in an aesthetically pleasing way. In a force-directed layout problem, the vertices of the graph are considered to be nodes connected by springs. Each spring exerts attractive or repulsive forces on the two nodes it connects, with the magnitude of the forces depending on the Euclidean distance between the nodes. Force-directed methods move the nodes until a static equilibrium is reached, with zero net force on each node, yielding an embedding of the vertices into $\mathbf{R}^2$. Force-directed methods, which are also called spring embedders, can be considered as MDE problems in which the distortion functions give the potential energy associated with the springs. Force-directed layout is a decades-old subject (Tutte, 1963; Eades, 1984; Kamada and Kawai, 1989), with early applications in VLSI layout (Fisk *et al.*, 1967; Quinn and Breuer, 1979) and continuing modern interest (Kobourov, 2012).

Low-rank models. A low-rank model approximates a matrix by one of lower rank, typically factored as the product of a tall and a wide matrix. These factors can be interpreted as embeddings of the rows and columns of the original matrix. Well-known examples of low-rank models include PCA and non-negative matrix factorization (Lee and Seung, 1999); there are many others (Udell *et al.*, 2016, §3.2). PCA (and its kernelized version) can be interpreted as solving an MDE problem, as we show in §3.2.

X2vec. Embeddings are frequently used to produce features for downstream machine learning tasks. Embeddings for this purpose were popularized with the publication of word2vec in 2013, an embedding method in which the items are words (Mikolov *et al.*, 2013). Since then, dozens of embeddings for different types of items have been proposed, such as doc2vec (Le and Mikolov, 2014), node2vec (Grover and Leskovec, 2016) and related methods (Perozzi *et al.*, 2014; Tang *et al.*, 2015), graph2vec (Narayanan *et al.*, 2017), role2vec (Ahmed *et al.*, 2020), (batter-pitcher)2vec (Alcorn, 2016), BioVec, ProtVec, and GeneVec (Asgari and Mofrad, 2015), dna2vec (Ng, 2017), and many others. Some of these methods resemble MDE problems, but most of them do not. Nonetheless MDE problems generically can be used to produce such X2vec-style embeddings, where X describes the type of items.

Neural networks. Neural networks are commonly used to generate embeddings for use in downstream machine learning tasks. One generic neural network based embedding method is the auto-encoder, which starts by representing items by (usually large dimensional) input vectors, such as one-hot vectors. These vectors are fed into an encoder neural network, whose output is fed into a decoder network. The output of the encoder has low dimension, and will give our embedding. The decoder attempts to reconstruct the original input from this low-dimensional intermediate vector. The encoder and decoder are both trained so the decoder can, at least approximately, reproduce the original input (Goodfellow *et al.*, 2016, §14).

More generally, a neural network may be trained to predict some relevant quantity, and the trained network's output (or an intermediate activation) can be used as the input's embedding. For example, neural networks for embedding words (or sequences of words) are often trained to predict masked words in a sentence; this is the basic principle underlying word2vec and BERT, two well-known word embedding methods (Mikolov *et al.*, 2013; Devlin *et al.*, 2019). Similarly, intermediate activations of convolutional neural networks like residual networks (He *et al.*, 2016), trained to classify images, are often used as embeddings of images. Neural networks have also been used for embedding single-cell mRNA transcriptomes (Szubert *et al.*, 2019).

Software. There are several open-source software libraries for specific embedding methods. The widely used Python library sci-kit learn (Pedregosa *et al.*, 2011) includes implementations of PCA, spectral embedding, Isomap, locally linear embedding, multi-dimensional scaling, and t-SNE, among others. The umap-learn package implements UMAP (McInnes, 2020b), the openTSNE package provides a more scalable variant of t-SNE (Poličar *et al.*, 2019), and GraphVite (which can exploit multiple CPUs and GPUs) implements a number of embedding methods (Zhu *et al.*, 2019). Embeddings for words and documents are available in gensim (Řehůřek and Sojka, 2010), Embeddings.jl (White and Ellison, 2019), HuggingFace transformers (HuggingFace, 2020), and BERT (Devlin, 2020). Force-directed layout methods are implemented in graphviz (Gansner and North, 2000), NetworkX (Hagberg *et al.*, 2008), qgraph (Epskamp *et al.*, 2012), and NetworkLayout.jl (*NetworkLayout.jl* 2020).

There are also several software libraries for approximately solving optimization problems with orthogonality constraints (which the MDE problem with standardization constraint has). Some examples include Manopt (and its related packages PyManopt and Manopt.jl) (Boumal *et al.*, 2014; Townsend *et al.*, 2016; Bergmann, 2020), Geoopt (Kochurov *et al.*, 2020), and McTorch (Meghwanshi *et al.*, 2018). More generally, problems with differentiable objective and constraint functions can be approximately solved using solvers for nonlinear programming, such as SNOPT (Gill *et al.*, 2002) (which is based on sequential quadratic programming) and IPOPT (Wächter and Biegler, 2006) (which is based on an interior-point method).

References

Absil, P.-A. and J. Malick. (2012). "Projection-like retractions on matrix manifolds". *SIAM Journal on Optimization*. 22(1): 135–158.

Absil, P.-A., R. Mahony, and R. Sepulchre. (2009). *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.

Ahmed, N., R. Rossi, J. Lee, T. Willke, R. Zhou, X. Kong, and H. Eldardiry. (2020). "Role-based graph embeddings". *IEEE Transactions on Knowledge and Data Engineering*.

Alcorn, M. (2016). "(batter|pitcher)2vec: Statistic-free talent modeling with neural player embeddings". In: MIT Sloan Sports Analytics Conference.

Andoni, A., P. Indyk, and I. Razenshteyn. (2018). "Approximate nearest neighbor search in high dimensions". *arXiv*.

Arrow, K. (1950). "A difficulty in the concept of social welfare". *Journal of Political Economy*. 58(4): 328–346.

Asgari, E. and M. Mofrad. (2015). "Continuous distributed representation of biological sequences for deep proteomics and genomics". *PLOS One*. 10(11): 1–15.

Asi, H. and J. Duchi. (2019). "Stochastic (approximate) proximal point methods: Convergence, optimality, and adaptivity". *SIAM Journal on Optimization*. 29(3): 2257–2290.

Barocas, S., M. Hardt, and A. Narayanan. (2019). *Fairness and Machine Learning*. URL: fairmlbook.org.

Beatson, R. and L. Greengard. (1997). “A short course on fast multipole methods”. In: *Wavelets, Multilevel Methods and Elliptic PDEs*. Oxford University Press. 1–37.

Belkin, M. and P. Niyogi. (2002). “Laplacian eigenmaps and spectral techniques for embedding and clustering”. In: *Advances in Neural Information Processing Systems*. 585–591.

Bender, E., T. Gebru, A. McMillan-Major, and S. Shmitchell. (2021). “On the dangers of stochastic parrots: Can language models be too big?” In: *Proceedings of the 2021 Conference on Fairness, Accountability, and Transparency*.

Bergmann, R. (2020). “Manopt.jl”. URL: https://manoptjl.org/stable/index.html.

Bernhardsson, E. (2020). “annoy”. URL: https://github.com/spotify/annoy.

Bernstein, M., V. De Silva, J. Langford, and J. Tenenbaum. (2000). “Graph approximations to geodesics on embedded manifolds”. *Tech. rep.* Department of Psychology, Stanford University.

Biswas, P. and Y. Ye. (2004). “Semidefinite programming for ad hoc wireless sensor network localization”. In: *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. 46–54.

Böhm, J. N., P. Berens, and D. Kobak. (2020). “A unifying perspective on neighbor embeddings along the attraction-repulsion spectrum”. *arXiv*.

Bolukbasi, T., K.-W. Chang, J. Zou, V. Saligrama, and A. Kalai. (2016). “Man is to computer programmer as woman is to homemaker? Debiasing word embeddings”. In: *Advances in Neural Information Processing Systems*. 4356–4364.

Borg, I. and P. Groenen. (2003). “Modern multidimensional scaling: Theory and applications”. *Journal of Educational Measurement*. 40(3): 277–280.

Boumal, N., B. Mishra, P.-A. Absil, and R. Sepulchre. (2014). “Manopt, a Matlab toolbox for optimization on manifolds”. *Journal of Machine Learning Research*. 15(1): 1455–1459.

Bourgain, J. (1985). “On Lipschitz embedding of finite metric spaces in Hilbert space”. *Israel Journal of Mathematics*. 52(1-2): 46–52.

Boyd, S. and L. Vandenberghe. (2004). *Convex Optimization*. New York, NY, USA: Cambridge University Press.

Boyd, S. and L. Vandenberghe. (2018). *Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares*. New York, NY, USA: Cambridge University Press.

Bradley, R. and M. Terry. (1952). "Rank analysis of incomplete block designs: The method of paired comparisons". *Biometrika*. 39(3/4): 324–345.

Broyden, C. G. (1970). "The convergence of a class of double-rank minimization algorithms, general considerations". *IMA Journal of Applied Mathematics*. 6(1): 76–90.

Burer, S. and R. Monteiro. (2003). "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization". *Mathematical Programming*. 95(2): 329–357.

Burer, S. and R. Monteiro. (2005). "Local minima and convergence in low-rank semidefinite programming". *Mathematical Programming*. 103(3, Ser. A): 427–444.

Carreira-Perpinán, M. and R. Zemel. (2005). "Proximity graphs for clustering and manifold learning". *Advances in Neural Information Processing Systems*. 17: 225–232.

Cayton, L. (2005). "Algorithms for manifold learning". *Tech. rep.* Department of Computer Science, University of California at San Diego.

Cayton, L. and S. Dasgupta. (2006). "Robust Euclidean embedding". In: *Proceedings of the 23rd International Conference on Machine Learning*. 169–176.

Chen, S., S. Ma, A. Man-Cho So, and T. Zhang. (2020). "Proximal gradient method for nonsmooth optimization over the Stiefel manifold". *SIAM Journal on Optimization*. 30(1): 210–239.

Chen, W., K. Weinberger, and Y. Chen. (2013). "Maximum variance correction with application to A* search". In: *International Conference on Machine Learning*. 302–310.

Chen, Y., C. Ding, J. Hu, R. Chen, P. Hui, and X. Fu. (2017). "Building and analyzing a global co-authorship network using Google Scholar Data". In: *Proceedings of the 26th International Conference on World Wide Web Companion*. 1219–1224.

Chung, F. and F. Graham. (1997). *Spectral Graph Theory*. No. 92. American Mathematical Society.

Corbett-Davies, S. and S. Goel. (2018). "The measure and mismeasure of fairness: A critical review of fair machine learning". *arXiv*.

Cox, T. and M. Cox. (2000). *Multidimensional Scaling*. CRC Press.

Devlin, J. (2020). "BERT". URL: https://github.com/google-research/bert.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.

Diakonikolas, I., G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart. (2017). "Being robust (in high dimensions) can be practical". In: *International Conference on Machine Learning*. 999–1008.

Dokmanic, I., R. Parhizkar, J. Ranieri, and M. Vetterli. (2015). "Euclidean distance matrices: Essential theory, algorithms, and applications". *IEEE Signal Processing Magazine*. 32(6): 12–30.

Dong, W., M. Charikar, and K. Li. (2011). "Efficient k-nearest neighbor graph construction for generic similarity measures". In: *Proceedings of the 20th International Conference on World Wide Web*. 577–586.

Dwork, C., M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. (2012). "Fairness through awareness". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. 214–226.

Dwork, C., R. Kumar, M. Naor, and D. Sivakumar. (2001). "Rank aggregation methods for the web". In: *Proceedings of the 10th International Conference on World Wide Web*. 613–622.

Eades, P. (1984). "A heuristic for graph drawing". In: *Proceedings of the 13th Manitoba Conference on Numerical Mathematics and Computing*. Vol. 42. 149–160.

Easley, D. and J. Kleinberg. (2010). *Networks, Crowds, and Markets*. Vol. 8. Cambridge University Press.

Eckart, C. and G. Young. (1936). "The approximation of one matrix by another of lower rank". *Psychometrika*. 1(3): 211–218.

Edelman, A., T. Arias, and S. Smith. (1998). "The geometry of algorithms with orthogonality constraints". *SIAM Journal on Matrix Analysis and Applications*. 20(2): 303–353.

El Alaoui, A., X. Cheng, A. Ramdas, M. Wainwright, and M. Jordan. (2016). "Asymptotic behavior of ℓ_p-based Laplacian regularization in semi-supervised learning". In: *Conference on Learning Theory*. 879–906.

Epskamp, S., A. Cramer, L. Waldorp, V. Schmittmann, and D. Borsboom. (2012). "qgraph: Network visualizations of relationships in psychometric data". *Journal of Statistical Software*. 48(4): 1–18.

Fan, K. and A. Hoffman. (1955). "Some metric inequalities in the space of matrices". *Proceedings of the American Mathematical Society*. 6(1): 111–116.

Fisk, C., d. Caskey, and L. West. (1967). "ACCEL: Automated circuit card etching layout". *Proceedings of the IEEE*. 55(11): 1971–1982.

Fletcher, R. (1970). "A new approach to variable metric algorithms". *The Computer Journal*. 13(3): 317–322.

Fligner, M. and J. Verducci. (1986). "Distance based ranking models". *Journal of the Royal Statistical Society: Series B (Methodological)*. 48(3): 359–369.

Gansner, E. and S. North. (2000). "An open graph visualization system and its applications to software engineering". *Software – Practice and Experience*. 30(11): 1203–1233.

Garg, N., L. Schiebinger, D. Jurafsky, and J. Zou. (2018). "Word embeddings quantify 100 years of gender and ethnic stereotypes". *Proceedings of the National Academy of Sciences*. 115(16): E3635–E3644.

Gill, P., W. Murray, and M. Saunders. (2002). "SNOPT: an SQP algorithm for large-scale constrained optimization". *SIAM Journal on Optimization*. 12(4): 979–1006.

Goldfarb, D. (1970). "A family of variable-metric methods derived by variational means". *Mathematics of Computation*. 24(109): 23–26.

Golub, G. and C. Van Loan. (2013). *Matrix Computations*. Fourth. *Johns Hopkins Studies in the Mathematical Sciences*. Johns Hopkins University Press, Baltimore, MD.

Goodfellow, I., Y. Bengio, and A. Courville. (2016). *Deep Learning*. MIT Press.

Google. "Google Scholar". URL: https://scholar.google.com/.

Greengard, L. and V. Rokhlin. (1987). "A fast algorithm for particle simulations". *Journal of Computational Physics*. 73(2): 325–348.

Groenen, P., J. de Leeuw, and R. Mathar. (1996). "Least squares multidimensional scaling with transformed distances". In: *From Data to Knowledge*. Springer. 177–185.

Grover, A. and J. Leskovec. (2016). "node2vec: Scalable feature learning for networks". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 855–864.

Hagberg, A., D. Schult, and P. Swart. (2008). "Exploring network structure, dynamics, and cunction using NetworkX". In: *Proceedings of the 7th Python in Science Conference*. 11–15.

Hall, K. (1970). "An r-dimensional quadratic placement algorithm". *Management Science*. 17(3): 219–229.

Ham, J., D. Lee, S. Mika, and B. Schölkopf. (2004). "A kernel view of the dimensionality reduction of manifolds". In: *International Conference on Machine Learning*. 47.

Hamilton, W., R. Ying, and J. Leskovec. (2017). "Representation learning on graphs: Methods and applications". *arXiv*.

Hayden, T., J. Wells, W.-M. Liu, and P. Tarazaga. (1991). "The cone of distance matrices". *Linear Algebra and its Applications*. 144: 153–169.

He, K., X. Zhang, S. Ren, and J. Sun. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

Higham, N. (1989). "Matrix nearness problems and applications". In: *Applications of Matrix Theory*. Vol. 22. Oxford University Press, New York. 1–27.

Hinton, G. and S. Roweis. (2003). "Stochastic neighbor embedding". In: *Advances in Neural Information Processing Systems*. 857–864.

Hiriart-Urruty, J.-B. and C. Lemaréchal. (1993). *Convex Analysis and Minimization Algorithms I. Fundamentals*. Vol. 305. *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin.

Hirsch, J. (2005). "An index to quantify an individual's scientific research output". *Proceedings of the National Academy of Sciences*. 102(46): 16569–16572.

Holstein, K., J. Wortman Vaughan, H. Daumé III, M. Dudik, and H. Wallach. (2019). "Improving fairness in machine learning systems: What do industry practitioners need?" In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–16.

Hosseini, S., W. Huang, and R. Yousefpour. (2018). "Line search algorithms for locally Lipschitz functions on Riemannian manifolds". *SIAM Journal on Optimization*. 28(1): 596–619.

Hotelling, H. (1933). "Analysis of a complex of statistical variables into principal components". *Journal of Educational Psychology*. 24(6): 417.

Hu, J., B. Jiang, L. Lin, Z. Wen, and Y. Yuan. (2019). "Structured quasi-Newton methods for optimization with orthogonality constraints". *SIAM Journal on Scientific Computing*. 41(4): A2239–A2269.

Huang, W., P.-A. Absil, and K. Gallivan. (2017). "Intrinsic representation of tangent vectors and vector transports on matrix manifolds". *Numerische Mathematik*. 136(2): 523–543.

Huang, W., P.-A. Absil, and K. Gallivan. (2018). "A Riemannian BFGS method without differentiated retraction for nonconvex optimization problems". *SIAM Journal on Optimization*. 28(1): 470–495.

Huang, W., K. Gallivan, and P.-A. Absil. (2015). "A Broyden class of quasi-Newton methods for Riemannian optimization". *SIAM Journal on Optimization*. 25(3): 1660–1685.

HuggingFace. (2020). "Transformers". URL: https://github.com/huggingface/transformers.

Hutchinson, B., V. Prabhakaran, E. Denton, K. Webster, Y. Zhong, and S. Denuyl. (2020). "Social Biases in NLP Models as Barriers for Persons with Disabilities". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5491–5501.

Hutchinson, M. (1989). "A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines". *Communications in Statistics – Simulation and Computation*. 18(3): 1059–1076.

Indyk, P., J. Matoušek, and A. Sidiropoulos. (2017). "Low-distortion embeddings of finite metric spaces". In: *Handbook of Discrete and Computational Geometry*. Ed. by C. D. Toth, J. O'Rourke, and J. E. Goodman. Chapman and Hall/CRC. Chap. 8. 211–231.

Jensen, T. and M. Diehl. (2017). "An approach for analyzing the global rate of convergence of quasi-Newton and truncated-Newton methods". *Journal of Optimization Theory and Applications*. 172(1): 206–221.

Ji, H. (2007). "Optimization approaches on smooth manifolds". *PhD thesis*. Australian National University.

Jiang, B. and Y.-H. Dai. (2015). "A framework of constraint preserving update schemes for optimization on Stiefel manifold". *Mathematical Programming*. 153(2): 535–575.

Joachims, T. (2002). "Optimizing search engines using clickthrough data". In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 133–142.

Johnson, W. and J. Lindenstrauss. (1984). "Extensions of Lipschitz mappings into a Hilbert space". *Contemporary Mathematics*. 26(189-206): 1.

Kamada, T. and S. Kawai. (1989). "An algorithm for drawing general undirected graphs". *Information Processing Letters*. 31(1): 7–15.

Knyazev, A. (2001). "Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method". *SIAM Journal on Scientific Computing*. 23(2): 517–541.

Knyazev, A. (2017). "Signed Laplacian for spectral clustering revisited". *arXiv*.

Knyazev, A. (2018). "On spectral partitioning of signed graphs". In: *2018 Proceedings of the Seventh SIAM Workshop on Combinatorial Scientific Computing*. SIAM. 11–22.

Kobak, D. and P. Berens. (2019). "The art of using t-SNE for single-cell transcriptomics". *Nature Communications*. 10(1): 1–14.

Kobourov, S. (2012). "Spring embedders and force directed graph drawing algorithms". *arXiv*.

Kochurov, M., R. Karimov, and S. Kozlukov. (2020). "Geoopt: Riemannian optimization in PyTorch". *arXiv*.

Kokiopoulou, E., J. Chen, and Y. Saad. (2011). "Trace optimization and eigenproblems in dimension reduction methods". *Numerical Linear Algebra with Applications*. 18(3): 565–602.

Koren, Y. (2003). "On spectral graph drawing". In: *International Computing and Combinatorics Conference*. Springer. 496–508.

Kruskal, J. (1964a). "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis". *Psychometrika*. 29(1): 1–27.

Kruskal, J. (1964b). "Nonmetric multidimensional scaling: A numerical method". *Psychometrika*. 29(2): 115–129.

Kunegis, J., S. Schmidt, A. Lommatzsch, J. Lerner, E. De Luca, and S. Albayrak. (2010). "Spectral analysis of signed graphs for clustering, prediction and visualization". In: *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM. 559–570.

Lanczos, C. (1951). "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators". In: *Proceedings of a Second Symposium on Large-Scale Digital Calculating Machinery*. Harvard University Press. 164–206.

Lawrence, N. (2011). "Spectral dimensionality reduction via maximum entropy". In: *International Conference on Artificial Intelligence and Statistics*. 51–59.

Le, Q. and T. Mikolov. (2014). "Distributed representations of sentences and documents". In: *International Conference on Machine Learning*. 1188–1196.

LeCun, Y., C. Cortes, and C. Burges. (1998). *The MNIST database of handwritten digits*. URL: http://yann.lecun.com/exdb/mnist/.

Lee, D. and S. Seung. (1999). "Learning the parts of objects by non-negative matrix factorization". *Nature*. 401(6755): 788–791.

Liberti, L., C. Lavor, N. Maculan, and A. Mucherino. (2014). "Euclidean distance geometry and applications". *SIAM Review*. 56(1): 3–69.

Lin, T. and H. Zha. (2008). "Riemannian manifold learning". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 30(5): 796–809.

Linial, N., E. London, and Y. Rabinovich. (1995). "The geometry of graphs and some of its algorithmic applications". *Combinatorica*. 15(2): 215–245.

Luce, R. (2012). *Individual choice behavior: A theoretical analysis*. Courier Corporation.

Ma, Y. and Y. Fu. (2011). *Manifold Learning Theory and Applications*. CRC press.

Maaten, L. van der and G. Hinton. (2008). "Visualizing data using t-SNE". *Journal of Machine Learning Research*. 9: 2579–2605.

Manton, J. (2002). "Optimization algorithms exploiting unitary constraints". *IEEE Transactions on Signal Processing*. 50(3): 635–650.

Martinet, B. (1970). "Brève communication. Régularisation d'inéquations variationnelles par approximations successives". *Revue française d'informatique et de recherche opérationnelle. Série rouge*. 4(R3): 154–158.

McInnes, L. (2020a). "pynndescent". URL: https://github.com/lmcinnes/pynndescent.

McInnes, L. (2020b). "UMAP". URL: https://github.com/lmcinnes/umap.

McInnes, L., J. Healy, and J. Melville. (2018). "UMAP: Uniform manifold approximation and projection for dimension reduction". *arXiv*.

Meghwanshi, M., P. Jawanpuria, A. Kunchukuttan, H. Kasai, and B. Mishra. (2018). "McTorch, a manifold optimization library for deep learning". *arXiv*.

Menger, K. (1928). "Untersuchungen über allgemeine Metrik". *Mathematische Annalen*. 100(1): 75–163.

Meyer, R., C. Musco, C. Musco, and D. Woodruff. (2020). "Hutch++: Optimal stochastic trace estimation". *arXiv*.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean. (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems*. 3111–3119.

Narayanan, A., M. Chandramohan, L. Rajasekar Venkatesan, Y.-L. Chen, and S. Jaiswal. (2017). "graph2vec: Learning distributed representations of graphs". In: *Workshop on Mining and Learning with Graphs*.

Nelson, M., K. Bryc, K. King, A. Indap, A. Boyko, J. Novembre, L. Briley, Y. Maruyama, D. Waterworth, G. Waeber, *et al.* (2008). "The Population Reference Sample, POPRES: a resource for population, disease, and pharmacological genetics research". *The American Journal of Human Genetics*. 83(3): 347–358.

"NetworkLayout.jl". (2020). URL: https://github.com/JuliaGraphs/NetworkLayout.jl.

Ng, P. (2017). "dna2vec: Consistent vector representations of variable-length k-mers". *arXiv*.

Nickel, M. and D. Kiela. (2017). "Poincaré embeddings for learning hierarchical representations". *Advances in Neural Information Processing Systems*. 30: 6338–6347.

Nocedal, J. (1980). "Updating quasi-Newton matrices with limited storage". *Mathematics of Computation*. 35(151): 773–782.

Nocedal, J. and S. Wright. (2006). *Numerical Optimization*. Second. *Springer Series in Operations Research and Financial Engineering*. Springer, New York.

Novembre, J., T. Johnson, K. Bryc, Z. Kutalik, A. Boyko, A. Auton, A. Indap, K. King, S. Bergmann, M. Nelson, *et al.* (2008). "Genes mirror geography within Europe". *Nature*. 456(7218): 98–101.

Page, L., S. Brin, R. Motwani, and T. Winograd. (1999). "The PageRank citation ranking: Bringing order to the web". *Tech. rep.* Stanford InfoLab.

Parikh, N. and S. Boyd. (2014). "Proximal algorithms". *Foundations and Trends in Optimization*. 1(3): 127–239.

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.* (2019). "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in Neural Information Processing Systems*. 8024–8035.

Pearson, K. (1901). "On lines and planes of closest fit to systems of points in space". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*. 2(11): 559–572.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.* (2011). "Scikit-learn: Machine learning in Python". *Journal of Machine Learning Research*. 12: 2825–2830.

Perozzi, B., R. Al-Rfou, and S. Skiena. (2014). "DeepWalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 701–710.

Plackett, R. (1975). "The analysis of permutations". *Journal of the Royal Statistical Society: Series C (Applied Statistics)*. 24(2): 193–202.

Poličar, P., M. Stražar, and B. Zupan. (2019). "openTSNE: A modular Python library for t-SNE dimensionality reduction and embedding". *bioRxiv*. DOI: 10.1101/731877.

Pothen, A., H. Simon, and K.-P. Liou. (1990). "Partitioning sparse matrices with eigenvectors of graphs". *SIAM Journal on Matrix Analysis and Applications*. 11(3): 430–452.

Quinn, N. and M. Breuer. (1979). "A forced directed component placement procedure for printed circuit boards". *IEEE Transactions on Circuits and systems*. 26(6): 377–388.

Řehůřek, R. and P. Sojka. (2010). "Software Framework for Topic Modelling with Large Corpora". In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA. 45–50.

Richardson, M. (1938). "Multidimensional psychophysics". *Psychological Bulletin*. 35: 659–660.

Ring, W. and B. Wirth. (2012). "Optimization methods on Riemannian manifolds and their application to shape space". *SIAM Journal on Optimization*. 22(2): 596–627.

Rockafellar, R. (1976). "Monotone operators and the proximal point algorithm". *SIAM Journal on Control and Optimization*. 14(5): 877–898.

Roweis, S. and L. Saul. (2000). "Nonlinear dimensionality reduction by locally linear embedding". *Science*. 290(5500): 2323–2326.

Ryu, E. and S. Boyd. (2014). "Stochastic proximal iteration: A non-asymptotic improvement upon stochastic gradient descent".

Sala, F., C. De Sa, A. Gu, and C. Ré. (2018). "Representation tradeoffs for hyperbolic embeddings". In: *International Conference on Machine Learning*. 4460–4469.

Sammon, J. (1969). "A nonlinear mapping for data structure analysis". *IEEE Transactions on Computers*. 100(5): 401–409.

Sandberg, R. (2014). "Entering the era of single-cell transcriptomics in biology and medicine". *Nature Methods*. 11(1): 22–24.

Saul, L. (2020). "A tractable latent variable model for nonlinear dimensionality reduction". *Proceedings of the National Academy of Sciences*. 117(27): 15403–15408.

Saul, L. and S. Roweis. (2001). "An introduction to locally linear embedding". *Tech. rep.*

Schönemann, P. (1966). "A generalized solution of the orthogonal Procrustes problem". *Psychometrika*. 31(1): 1–10.

Schouten, B., M. Calinescu, and A. Luiten. (2013). "Optimizing quality of response through adaptive survey designs". *Survey Methodology*. 39(1): 29–58.

Shanno, D. (1970). "Conditioning of quasi-Newton methods for function minimization". *Mathematics of Computation*. 24(111): 647–656.

Sherwani, N. (2012). *Algorithms for VLSI Physical Design Automation*. Springer Science & Business Media.

Sigl, G., K. Doll, and F. Johannes. (1991). "Analytical placement: A linear or a quadratic objective function?" In: *Proceedings of the 28th ACM/IEEE design automation conference*. 427–432.

Szubert, B., J. Cole, C. Monaco, and I. Drozdov. (2019). "Structure-preserving visualisation of high dimensional single-cell datasets". *Scientific Reports*. 9(1): 1–10.

Tang, J., J. Liu, M. Zhang, and Q. Mei. (2016). "Visualizing large-scale and high-dimensional data". In: *Proceedings of the 25th International Conference on World Wide Web*. 287–297.

Tang, J., M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. (2015). "LINE: Large-scale information network embedding". In: *Proceedings of the 24th International Conference on World Wide Web*. 1067–1077.

Tenenbaum, J., V. De Silva, and J. Langford. (2000). "A global geometric framework for nonlinear dimensionality reduction". *Science*. 290(5500): 2319–2323.

Torgerson, W. (1952). "Multidimensional scaling: I. Theory and method". *Psychometrika*. 17(4): 401–419.

Townsend, J., N. Koep, and S. Weichwald. (2016). "PyManopt: A python toolbox for optimization on manifolds using automatic differentiation". *The Journal of Machine Learning Research*. 17(1): 4755–4759.

Trefethen, L. and D. Bau. (1997). *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Tutte, W. T. (1963). "How to draw a graph". *Proceedings of the London Mathematical Society*. 3(1): 743–767.

Udell, M., C. Horn, R. Zadeh, S. Boyd, *et al.* (2016). "Generalized low rank models". *Foundations and Trends in Machine Learning*. 9(1): 1–118.

United States Census Bureau. "American Community Survey 2013–2017 5-Year Data". URL: https://www.census.gov/newsroom/press-kits/2018/acs-5year.html.

Von Ahn, L. and L. Dabbish. (2008). "Designing games with a purpose". *Communications of the ACM*. 51(8): 58–67.

von Luxburg, U. (2007). "A tutorial on spectral clustering". *Statistics and Computing*. 17(4): 395–416.

Wächter, A. and L. Biegler. (2006). "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". *Mathematical Programming*. 106(1, Series A): 25–57.

Wang, Y., H. Huang, C. Rudin, and Y. Shaposhnik. (2020). "Understanding how dimension deduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMAP, and PaCMAP for data visualization". *arXiv*.

Weinberger, K. and L. Saul. (2004). "Unsupervised learning of image manifolds by semidefinite programming". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2.

White, L. and D. Ellison. (2019). "Embeddings.jl: Easy access to pretrained word embeddings from Julia". *Journal of Open Source Software*. 4(36): 1013.

Wilk, A., A. Rustagi, N. Zhao, J. Roque, G. Martínez-Colón, J. McKechnie, G. Ivison, T. Ranganath, R. Vergara, T. Hollis, *et al.* (2020). "A single-cell atlas of the peripheral immune response in patients with severe COVID-19". *Nature Medicine*: 1–7.

Wilson, R., E. Hancock, E. Pekalska, and R. Duin. (2014). "Spherical and hyperbolic embeddings of data". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 36(11): 2255–2269.

Xu, Y. (2010). "Semi-supervised Learning on Graphs: A Statistical Approach". *PhD thesis*. Stanford University.

Yan, S., D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. (2006). "Graph embedding and extensions: A general framework for dimensionality reduction". *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 29(1): 40–51.

Young, G. and A. Householder. (1938). "Discussion of a set of points in terms of their mutual distances". *Psychometrika*. 3(1): 19–22.

Zhou, S., N. Xiu, and H.-D. Qi. (2019). "Robust Euclidean embedding via EDM optimization". *Mathematical Programming Computation*: 1–51.

Zhu, Z., S. Xu, M. Qu, and J. Tang. (2019). "GraphVite: A high-performance CPU-GPU hybrid system for node embedding". In: *Proceedings of the World Wide Web Conference*. 2494–2504.