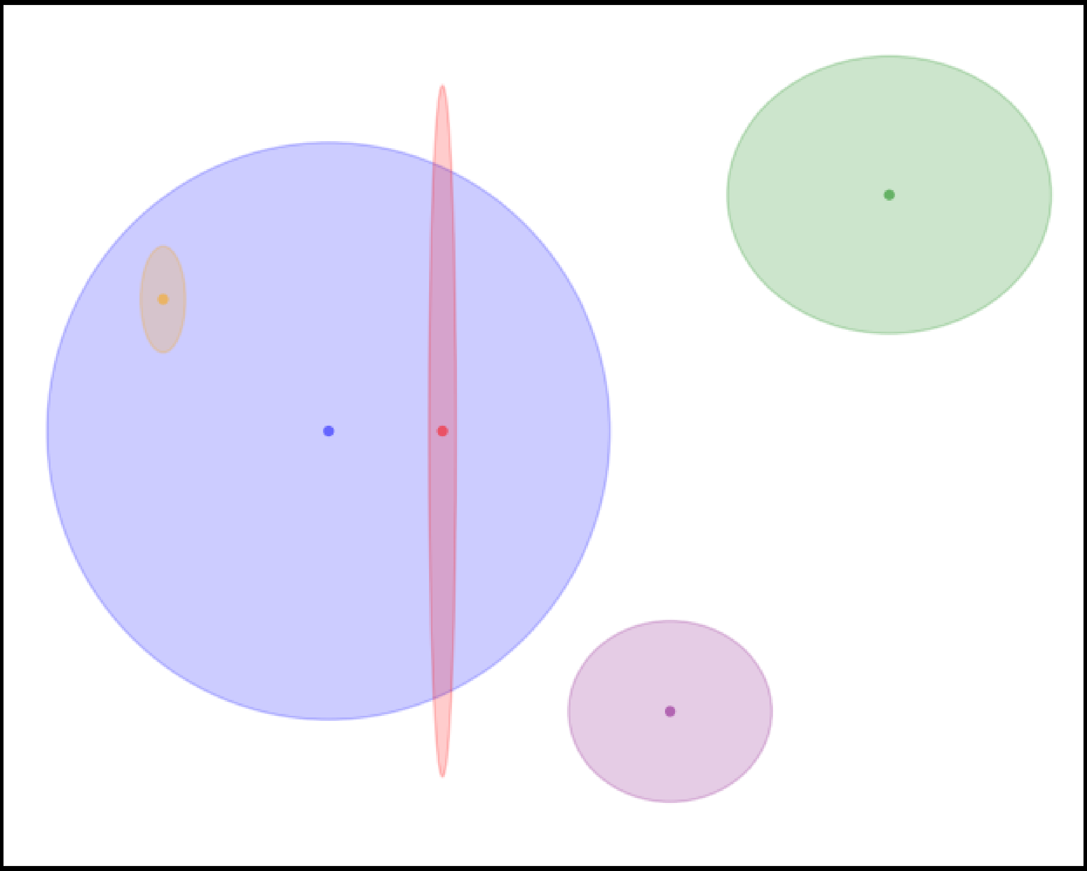
# Extending Dimensionality Reduction Techniques for Probabilistic Data



Yanto Christoffel

Layout: typeset by the author using LaTeX.
Cover illustration: Yanto Christoffel

# Extending Dimensionality Reduction Techniques for Probabilistic Data

Yanto Christoffel
12480800

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 900
1098 XH Amsterdam

*Supervisor*
Dr. Vlad Niculae

Informatics Institute
Faculty of Science
University of Amsterdam
Science Park 900
1098 XH Amsterdam

Semester 2, 2022-2023

# Abstract

In this research, we consider the problem of learning 2D probabilistic embeddings given high-dimensional distribution data. To deal with this problem, we present extensions of two well-known dimensionality reduction techniques: principal component analysis (PCA), and t-distributed neighbor embedding (t-SNE). These methods have previously been used on vectors, but we are extending them to examine their efficacy on distribution data. Using a data set consisting of words represented as high-dimensional Gaussian distributions, our extended techniques output 2D ellipses that each represent a word. Visualizing these elliptical representations of probabilistic data can capture concepts that point representations cannot do, such as the uncertainty of a word. We evaluate the extended techniques to determine their capabilities and find that they provide a richer interpretation of the high-dimensional space than their conventional counterparts.

# Contents

# Chapter 1

# Introduction

Dimensionality reduction techniques are a key part of high-dimensional data visualization. These techniques represent high-dimensional data in a low-dimensional space while preserving as much of the original information as possible. They produce faithful visualizations of the data, offering a means to understand intricate data sets that would otherwise be challenging to interpret.

In natural language processing, researchers use dimensionality reduction techniques to visualize high-dimensional word embeddings for intuitive interpretations of a complicated space (Liu et al., 2018). Through these visualizations, researchers may examine the overall structure of the data set and analyze relationships between words.

Current dimensionality reduction techniques have proven to be effective with words embedded as vectors, not only for visualization purposes (Young and Rusli, 2019), but also for reducing memory requirements (Raunak, 2017). Nevertheless, it is important to note that the visualizations produced by these techniques depict each word solely as a point in the projected space, which may not fully capture certain properties of words, such as uncertainty (Vilnis and McCallum, 2015). In order for dimensionality reduction techniques to be able to capture these properties, they have to be extended to incorporate probabilistic data, or in other words, data where every word is embedded as a probability distribution. Vilnis and McCallum (2015) argue that embeddings represented as probability distributions can capture these notions of uncertainty as well as enable more expressive relationships between objects.

To tackle this, we present extensions of two popular existing dimensionality reduction techniques, principal component analysis (Pearson, 1901, PCA) and t-distributed stochastic neighbor embedding (Hinton and van der Maaten, 2008, t-SNE), so that they may be applied to probabilistic embeddings. For our probability distribution we choose a multivariate Gaussian distribution, providing each embedded word with a mean vector and a covariance vector. The outputs of our extensions enable visualizations where each low-dimensional word embedding has an ellipse associated with it, describing its variance. Given Gaussian embeddings, our extended techniques are able to visualize a high-dimensional data set with more expressivity than the original techniques.

# Chapter 2

# Background

## 2.1 PCA

Principal component analysis (Pearson, 1901, PCA) is a linear dimensionality reduction technique, extensively applied in wide variety of domains, including bioinformatics (Ringnér, 2008), chemistry (Joswiak et al., 2019), natural language processing (Young and Rusli, 2019), and many more.

PCA is able to construct a low-dimensional representation of a high-dimensional data set. It achieves this by finding the principal components, which are orthogonal axes representing the directions that maximize the variance in the data set. By projecting the data onto these directions, PCA effectively reduces the dimensionality of the data set.

The central idea behind PCA is grounded in linear algebra. Essentially, PCA finds a linear subspace that preserves the structure of the data. This subspace is spanned by the eigenvectors that correspond to the largest eigenvalues, obtained via eigendecomposition of the data's covariance matrix.

Besides maximizing the variance, there is another method of obtaining the principal components. Projecting the data onto the eigenvectors inevitably results in information loss, provided that we project onto a number of eigenvectors that is less than the original dimensionality of the data. By mapping the low-dimensional projection back to $\mathbb{R}^d$, we can measure how much information was lost by the projection. This loss of information is called the reconstruction error, and minimization of it leads to the same principal component solutions as maximizing the variance (Murphy, 2022).

## 2.2 t-SNE

t-Distributed stochastic neighbor embedding (t-SNE) is a popular and powerful dimensionality reduction technique introduced by Hinton and van der Maaten (2008) as a variation of stochastic neighbor embedding (Hinton and Roweis, 2002, SNE). The optimization objective of t-SNE focuses on preserving the local structure of the data rather than the global structure that PCA is focused on.

The core idea of t-SNE is to map high-dimensional data points to a lower-dimensional space while keeping similar data points from the high-dimensional space close together in the lower-dimensional space.

t-SNE is non-parametric, meaning that it does not construct an explicit function that maps the high-dimensional points to 2D. Instead, it randomly places the data points in the space, and shifts their positions according to pairwise similarities. More specifically, we measure the distance for each pair of high-dimensional data points. We then convert this distance between two neighboring data points to form a probability distribution. For two high-dimensional data points $x_i$ and $x_j$, we have the conditional probability:

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma^2)}{\sum_{k\neq i}\exp(-||x_i - x_k||^2/2\sigma^2)} \tag{2.1}$$

and for the two corresponding points $y_i$ and $y_j$ in the low-dimensional space, we have the conditional probability:

$$q_{j|i} = \frac{\exp(-||y_i - y_j||^2/2\sigma^2)}{\sum_{k\neq i}\exp(-||y_i - y_k||^2/2\sigma^2)} \tag{2.2}$$

t-SNE aims to minimize the Kullback-Leibler (KL) divergence between these two probabilities, which is defined as:

$$D_{KL}(p_{j|i}, q_{j|i}) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{2.3}$$

Minimization of this cost function encourages similar data points from the original space to stay close together in the reduced space, while dissimilar data points are modeled as far apart as possible. This optimization process is done iteratively over all data points, adjusting their positions at each step (Hinton and van der Maaten, 2008).

Due to the fact that t-SNE places emphasis on preserving the local structure of the data, it is highly effective at visualizing clusters and identifying outliers. The extent to which t-SNE clusters data points is controlled by the perplexity parameter. This value roughly corresponds to the number of neighbors that t-SNE tries to preserve.

Unlike PCA, t-SNE is a nonlinear method. The distinction between linear and nonlinear methods is important to recognize because nonlinear methods are often more powerful than linear ones. This is attributed to the fact that the relationship between the latent variables and observed ones can have a much more complex structure than a linear transformation is capable of describing (Lee and Verleysen, 2007, p. 40).

## 2.3 Probabilistic Embeddings

Probabilistic embeddings, in which each embedded object is represented as a probability distribution, can capture the uncertainty of objects as well as relationships between objects (Vilnis and McCallum, 2015). In our case, we have probabilistic word embeddings, which means that we assign a probability distribution over the possible contextualized representations of every word in our vocabulary. In some cases, these contextualized representations may be able to capture the meaning of a word (Mikolov et al., 2013a).

To construct probabilistic embeddings, each embedded object needs to satisfy the parameters for the chosen probability distribution. For instance, consider a data set consisting of word embeddings. If our chosen distribution is a multivariate Gaussian distribution, each embedded word requires a corresponding mean vector and covariance vector. Tol (2023) extracted probabilistic word embeddings from a BERT-based model, which is a pre-trained contextual language model (Devlin et al., 2019). Due to the effectiveness of these probabilistic word embeddings, we use a similar model for extracting our probabilistic embeddings, which we cover in Section 3.1.

## 2.4 Wasserstein Distance

Vilnis and McCallum (2015) embedded words as Gaussians, endowed with the Kullback-Leibler (KL) divergence for measuring similarity between the distributions. This metric, however, has a disadvantage when the covariances of a Gaussian distribution collapse: its probability $p(x)$ goes to $\infty$, causing the KL divergence to also go to $\infty$ (Muzellec and Cuturi, 2019). For comparing distributions, Muzellec and Cuturi (2019) propose to use the squared 2-Wasserstein distance, which can handle such degenerate measures. For two random variables $X, Y$, their squared 2-Wasserstein distance is defined by Peyré et al. (2019) as:

$$W_2^2(X, Y) = \min_{G \in \Gamma(X,Y)} \mathbb{E}_{(x,y)\sim G}[||x - y||_2^2] \tag{2.4}$$

where the minimization is over the set $\Gamma(X, Y)$ of joint couplings $G$, whose margins match $X$ and $Y$. For two Gaussians $\alpha$ and $\beta$, the definition is simplified (Peyré et al., 2019, Remark 2.31):

$$W_2^2(\alpha, \beta) = ||\mathbf{a} - \mathbf{b}||_2^2 + \mathfrak{B}^2(\Sigma_\alpha, \Sigma_\beta) \tag{2.5}$$

We have that the squared 2-Wasserstein distance is equal to the squared Euclidean distance, plus $\mathfrak{B}^2$, which is the squared Bures metric (Bhatia et al., 2018). It is defined as:

$$\mathfrak{B}^2(\Sigma_1, \Sigma_2) = \mathrm{Tr}(\Sigma_1) + \mathrm{Tr}(\Sigma_2) - 2\,\mathrm{Tr}((\Sigma_1^{\frac{1}{2}}\Sigma_2\Sigma_1^{\frac{1}{2}})^{\frac{1}{2}}) \tag{2.6}$$

This formula has a constraint, however, because the covariance matrices must be symmetric and positive semidefinite. Furthermore, computing matrix square roots is required. A convenient workaround is using diagonal covariance matrices instead of full covariance matrices, which simplifies the definition of the squared 2-Wasserstein distance once again:

$$W_2^2(\alpha, \beta) = ||\mathbf{a} - \mathbf{b}||_2^2 + ||\sqrt{\mathbf{d_A}} - \sqrt{\mathbf{d_B}}||_2^2 \tag{2.7}$$

We see that the squared 2-Wasserstein distance in the case of diagonal covariances is equal to the Euclidean distance between the means, plus the Hellinger metric between the diagonals of the scale matrices (Muzellec and Cuturi, 2019). For measuring similarity between the Gaussian distributions in our extension of t-SNE, we use the 2-Wasserstein distance with diagonal covariance matrices precisely because of this convenience. This is not possible for the PCA extension, since we are not able to achieve diagonal 2D covariance matrices. For distributional PCA, we instead compute the 2-Wasserstein distances using the formula from Eq. 2.5.

# Chapter 3

# Method

## 3.1 Probabilistic Data Set

To extract our embeddings, we use a pre-trained RoBERTa base model from Hugging Face Transformers (Liu et al., 2019). For each word, we extract 100 contexts (paragraphs) containing that word. These contexts are chosen at random from the Wikipedia data set as preprocessed by Meng et al. (2019). For each word, we compute the contextual subword embeddings from all of its contexts and average over them. These contextual subword embeddings are outputs of the RoBERTa base model.

The vocabulary of our data set is the same vocabulary used in Tol (2023), and contains 3195 words. Each word is assigned a 768-dimensional mean vector and a 768-dimensional covariance vector that serve as the parameters for its multivariate Gaussian distribution.

## 3.2 Extensions to Distribution Data

### 3.2.1 Distributional PCA

Standard PCA requires a data set

$$\{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\} \in \mathbb{R}^d$$

of which we want to find directions $\boldsymbol{u} \in \mathbb{R}^d$ that maximize the variance of the projected data:

$$\underset{\boldsymbol{u}\in\mathbb{R}^d, ||\boldsymbol{u}||=1}{\arg\max} \sum_i (\boldsymbol{u}^\top \boldsymbol{x}_i)^2$$

In our case, we have words embedded as multivariate Gaussian distributions where each word $i$ corresponds to random variable $\mathrm{X}_i$, defined as:

$$\mathrm{X}_i = \mathcal{N}(\mu_i, \Sigma_i)$$

where $\mu_i$ and $\Sigma_i$ are the mean and covariance matrix of a word $i$, respectively. For the derivation of PCA on these distributions, we use the following proof by Niculae (2023). We aim to maximize the expected projected variance:

$$\underset{\boldsymbol{u} \in \mathbb{R}^d, ||\boldsymbol{u}||=1}{\arg\max} \; \mathbb{E}\left[\sum_i (\boldsymbol{u}^\top X_i)^2\right]$$

By rearranging and using linearity of expectation, we derive:

$$\begin{aligned} \mathbb{E}\left[\sum_i (\boldsymbol{u}^\top X_i)^2\right] &= \mathbb{E}\left[\sum_i \boldsymbol{u}^\top (X_i X_i^\top) \boldsymbol{u}\right] \\ &= \sum_i \boldsymbol{u}^\top (\mathbb{E}[X_i X_i^\top]) \boldsymbol{u} \\ &= \boldsymbol{u}^\top \left(\sum_i \mu_i \mu_i^\top + \Sigma_i\right) \boldsymbol{u} \end{aligned} \tag{3.1}$$

This proof shows that PCA on probabilistic embeddings can be achieved by eigendecomposition of the composite matrix $\sum_i \mu_i \mu_i^\top + \Sigma_i$. In other words, we add the covariances of each word to the symmetric matrix $\sum_i \mu_i \mu_i^\top$, before performing eigendecomposition to obtain the principal components.

Once we have obtained the principal components using this formula, we have a linear mapping $\boldsymbol{U}$ from the high-dimensional space to the low-dimensional space. In general, we can say that if $X = \mathcal{N}(\mu, \Sigma)$, then $\boldsymbol{U}X = \mathcal{N}(\boldsymbol{U}\mu, \boldsymbol{U}\Sigma\boldsymbol{U}^\top)$. Note that even if $\Sigma$ is diagonal, the 2D covariance matrix $\boldsymbol{U}\Sigma\boldsymbol{U}^\top$ will contain full covariances. This means that, for a word $i$, its 2D covariance matrix $\bar{\Sigma}_i$ is of the form:

$$\bar{\Sigma}_i = \begin{pmatrix} u & w \\ w & v \end{pmatrix}$$

Unlike with diagonal covariance matrices, ellipses from full covariance matrices are not necessarily axis-aligned, and allow rotation. Rotations in 2D by an angle $\theta \in [-\pi, \pi]$ can be done with an orthogonal matrix of the form:

$$\boldsymbol{R}_\theta = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$

with $\boldsymbol{R_\theta} = \boldsymbol{R_\theta}^\top = \boldsymbol{R_\theta}^{-1}$ (Axler, 1997). Furthermore, any 2D covariance matrix $\Sigma$ can be diagonalized using some rotation matrix $\boldsymbol{R_\theta}$:

$$\Sigma = \boldsymbol{R_\theta} \boldsymbol{\Lambda} \boldsymbol{R_\theta}^\top \tag{3.2}$$

Hence, given $\bar{\Sigma}$, we can compute its eigendecomposition to get the eigenvalue $\lambda$ and the angle $\theta$. Since we obtained the 2D eigenvectors and eigenvalues for every word in our data set with Eq. 3.2, we essentially obtained the rotation and scale for every word's ellipse. Specifically, the orientation of each word's ellipsis is contained within $\boldsymbol{R_\theta}$, and the semiaxes of each word's ellipsis are contained within $\boldsymbol{S} = \sqrt{\boldsymbol{\Lambda}}$.

We wish to only plot a certain amount of probability mass of each word's 2D distribution. This is determined by the interval (John, 1968). For the multivariate Gaussian distribution, we can describe the interval as a region consisting of $\mathbf{x}$ satisfying:

$$(\mathbf{x} - \mu)^\top \Sigma^{-1} (\mathbf{x} - \mu) \leq \chi_k^2(p)$$

where $\chi_k^2(p)$ is the quantile function for probability $p$ of the chi-squared distribution with $k$ degrees of freedom (John, 1968). In our case, since we are mapping to 2D, we have $k = 2$. We can then say that, for any word $i$:

$$\frac{(x_{i1} - \mu_{i1})^2}{\chi_k^2(p) s_{i1}} + \frac{(x_{i2} - \mu_{i2})^2}{\chi_k^2(p) s_{i2}} = 1 \tag{3.3}$$

where $s_{i1}$ and $s_{i2}$ are contained in $\boldsymbol{S}_i$. With the equation for an ellipse in mind, we have derived that the semiaxes of the ellipse of a word $i$ are equal to $(\sqrt{\chi_k^2(p) s_{i1}}, \sqrt{\chi_k^2(p) s_{i2}})$. The only parameter left to decide on is $p$, which decides what percentage of the probability mass we want the ellipses to represent. We set $p$ to 0.5, such that each ellipse shows 50% of the probability mass of its Gaussian distribution.

Having procured the semiaxes and orientations of the ellipses, we now possess the necessary elements to produce visualizations for distributional PCA.

### 3.2.2 Distributional t-SNE

Standard t-SNE learns a vector for each data point $x_i$, and its optimization objective depends on pairwise distances $d(x_i, x_j)$. Our data consists of words embedded as Gaussian distributions, parametrized by a mean and covariance. We represent each data point as a random variable $X_i \sim \mathcal{N}(\mu_i, \text{diag}(s_i))$ where $s_i$ is a non-negative vector. Using Eq. 2.1 and 2.2, we have the conditional probability:

$$p_{j|i} = \frac{\exp(-d(X_i - X_j)^2 / 2\sigma^2)}{\sum_{k \neq i} \exp(-d(X_i - X_k)^2 / 2\sigma^2)} \tag{3.4}$$

for two high-dimensional data points $X_i$ and $X_j$. The two corresponding points $Y_i$ and $Y_j$ in the low-dimensional space have the conditional probability:

$$q_{j|i} = \frac{\exp(-d(Y_i - Y_j)^2/2\sigma^2)}{\sum_{k \neq i} \exp(-d(Y_i - Y_k)^2/2\sigma^2)} \tag{3.5}$$

To treat $X$ and $Y$ as embeddings, we take $d_2 = W_2$. Conveniently, the computation of these distances is equivalent to the squared Euclidean distances between the means concatenated to the square roots of the diagonal covariances:

$$d(X_i, X_j) = ||\mu_i - \mu_j||^2 + ||\sqrt{s_i} - \sqrt{s_j}||^2 \tag{3.6}$$

Therefore, this concatenation can be given as input to a t-SNE model that optimizes the squared Euclidean distances, and we will have a solution that is optimized in Wasserstein space.

We keep the same cost function as in standard t-SNE, the KL divergence. For minimizing this cost function, t-SNE employs gradient descent updates of the form:

$$y_{i+1} \leftarrow y_i - \gamma \nabla_{y_i} l(x, y)$$

We require part of the output, specifically the covariances, to be non-negative. To achieve this, we instead use projected gradient descent (Bertsekas, 1997) to learn our means and covariances. In our particular scenario, the updates can be written in the form:

$$\begin{aligned} \mu_{i+1} &\leftarrow \mu - \gamma \nabla_{\mu_i} l(x, \mu, s) \\ s_{i+1} &\leftarrow \Pi(s_i - \gamma \nabla_{s_i} l(x, \mu, s)) \end{aligned}$$

where $[\Pi(x)]_i = \max(x_i, \epsilon)$, which denotes the projection on the set $[\epsilon, \infty)$. Following Muzellec and Cuturi (2019), we set the regularization term $\epsilon$ to 0.001 to ensure that all gradients exist. We use an existing implementation of t-SNE,[1] and alter it by introducing the diagonal covariance parameters.

Unlike in distributional PCA, we have no eigenvectors $\boldsymbol{U}$ and instead learn the 2D covariance matrices for every word during the optimization process. We obtain the semiaxes using these learned covariance matrices in combination with Eq. 3.3. As mentioned in Section 2.4, we use diagonal covariance matrices so that computing the 2-Wasserstein distances is easier. Consequently, the visualizations of distributional t-SNE will represent words as axis-aligned ellipses, while the ellipses in the distributional PCA visualizations can take any shape.

[1] t-SNE implementation in Python by Xiao Li (2020): https://github.com/mxl1990/tsne-pytorch/tree/master

## 3.3 Evaluation Metrics

The goal of dimensionality reduction is to preserve the structural information from the high-dimensional data. For evaluation, we have divided this goal into two parts: global structure preservation and local structure preservation. These two parts highlight the qualities of both PCA and t-SNE, as the two techniques optimize the preservation of structure in different ways.

### 3.3.1 Spearman's Rank Correlation Coefficient

For our global metric, we use Spearman's rank correlation coefficient, also called Spearman's $\rho$, which is defined as the Pearson correlation coefficient between the rank variables (Myers and Well, 2003). The term "rank variables" refers to the rank of each data point when the data are sorted. In other words, the numerical value of each data point is replaced by its position in the sorted data list. For two variables $X$ and $Y$, the Spearman's $\rho$ is mathematically defined as:

$$\rho_{\mathrm{R}(X),\mathrm{R}(Y)} = \frac{\mathrm{cov}(\mathrm{R}(X), \mathrm{R}(Y))}{\sigma_{\mathrm{R}(X)}\sigma_{\mathrm{R}(Y)}}$$

where $\mathrm{R}(X)$ and $\mathrm{R}(Y)$ are the rank variables of $X$ and $Y$, respectively.

The scale of distances between words in PCA and t-SNE varies wildly, so we require a qualitative measure that does not compare the distance values directly. Spearman's $\rho$ is a fitting choice as the distance values are converted to ranks before calculating the coefficient. This way, the scale of the projected data does not matter.

The coefficient ranges from $-1$ to 1, where a negative coefficient indicates a negative correlation, a positive coefficient indicates a positive correlation, and a coefficient close to zero indicates that there is no correlation between the original distances and the model distances.

To obtain coefficients for each model, we first compute the original pairwise distances from the high-dimensional embeddings learned by the RoBERTa base model to form a distance matrix. Second, we compute the pairwise distances between every word for every model so that each model has its own distance matrix. As all of these distance matrices are symmetrical, we use their upper triangles as vectors. We compute the Spearman's $\rho$ between the original distances and the distances of each model to measure how correlated the low-dimensional representations are with the high-dimensional data. We consider this a global metric because it uses the entirety of the pairwise distances per model for calculating the coefficients.

### 3.3.2 Average nDCG

For our local metric we use normalized discounted cumulative gain (Järvelin and Kekäläinen, 2002, nDCG), regularly used for evaluation of search engines (Gubanov and Pyayt, 2016). In search engine evaluation, we often are interested in the relevance of the top-$k$ results from a web search. With our dimensionality reduction techniques, we care about how well the top-$k$ neighbors of a word are preserved in the low-dimensional projection. The nDCG score is defined as:

$$\text{nDCG}_k = \frac{DCG_k}{IDCG_k}$$

where $DCG_k$ is the discounted cumulative gain at position $k$, defined as:

$$DCG_k = \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)}$$

In this formula, $\text{rel}_i$ refers to the relevance score of the item at position $i$. The $IDCG_k$ is the ideal discounted cumulative gain, which represents the $DCG_k$ of a perfect ranking of the top-$k$ neighbors. Often, this perfect ranking is equal to the top $k$ neighbors in descending order of their relevance scores. The nDCG score ranges from 0 to 1, with a score of 1 being a perfect score.

In our case, this metric can be calculated for each word by taking their $k$ closest neighbors and assigning relevance scores. The relevance scores of words are dependent on their positions in the $k$ closest neighbors from the original high-dimensional space. For example, with $k = 10$, the closest neighbor of a word has a relevance score of 10, the second closest neighbor has a score of 9, and so on. Every word that is not in the $k$ closest neighbors is assigned a score of 0.

For every word, we compute the nDCG score by comparing the negative distances of its nearest neighbors in the model with their true relevance scores. We take the negative distances because low distance values correspond to high relevance values. This is done for each word in the vocabulary, after which we take the average of the scores, resulting in a singular score. In short, this score will be high when the low-dimensional output has preserved the local distances from the high-dimensional space well.

We also consider a different calculation of this metric. Instead of the true relevance scores ranging from 1 to $k$, we compute the relevance scores from the full ranking. This means that we still have $k$ neighbors for each word, but their relevance scores range from 1 to $N$, which is the vocabulary size. In this case, the penalty for not preserving close neighbors is reduced.

# Chapter 4

# Results and Discussion

We compare a total of six different models on our evaluation metrics, two of which are the standard techniques, and four of which are our extended techniques. To assess the impact of using the Wasserstein distance metric, our extended techniques are evaluated in both Euclidean and Wasserstein space. To clarify, in distributional PCA with distances in Euclidean, we discard the covariances and compute the distances between the projected means. In distributional t-SNE with distances in Euclidean, we give distributions as input and learn points as output.

The optimization of t-SNE's objective function relies on a gradient-descent approach that is initialized randomly. Consequently, different runs of t-SNE may yield different solutions. To ensure consistency in our evaluation, we generated multiple t-SNE visualizations, and selected the visualizations that had the lowest error values. PCA does not require this consideration as it is deterministic.

## 4.1 Experiments on Synthetic Data

To gain intuition for our distributional methods, we conduct experiments with synthetic data. Our synthetic data set consists of two data points with means:

$$\mu_1 = \mu_2 = [0, 0, 0, 0]$$

and covariances:

$$\Sigma_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}, \Sigma_2 \begin{pmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The means of the two data points are equal, while their covariance matrices are opposite. Distributional PCA linearly transforms the data points, not allowing their means to shift. We would like to show, however, that our method is capable of distinguishing between the different covariances. With these specific means and covariances, we essentially perform eigendecomposition on an identity matrix, multiplied by a constant $c$. All the eigenvalues resulting from this decomposition will be equal to $c$, so any two eigenvectors can be picked as the principal components. To circumvent the deterministic nature of PCA for the purpose of this experiment, we add a small amount of noise to the diagonals of the covariance matrices. This added randomness makes the eigenvalues fluctuate, causing distributional PCA to pick different eigenvectors at different runs.

Our synthetic data set is 4-dimensional, so we get four eigenvectors $(e_1, e_2, e_3, e_4)$ from decomposition. Depending on the combination of eigenvectors that are selected as the leading principal components, we either get ellipses nested inside each other, or ellipses that form a cross shape, as seen in Figure 4.1.

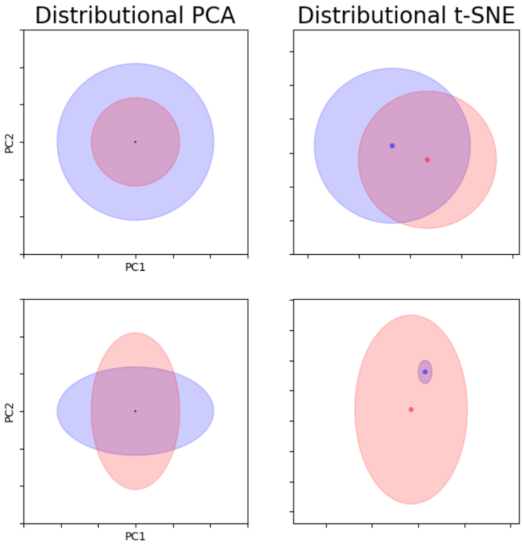


Figure 4.1: Solutions from distributional PCA (column 1) and distributional t-SNE (column 2) of the same synthetic data set. Coordinate values on the axes are hidden because we are interested in relative distances rather than numerical distances.

The principal components consisting of $(e_1, e_2)$, and $(e_3, e_4)$, lead to the nested ellipses, while principal components $(e_1, e_3), (e_1, e_4), (e_2, e_3)$, and $(e_2, e_4)$ lead to a cross-shape. The latter solution retains the covariances from the original space. Given that every combination of eigenvectors is equally likely to be selected, the cross-shaped solution will occur $\frac{2}{3}$ of the time, while the nested ellipses will occur $\frac{1}{3}$ of the time.

Distributional t-SNE, on the other hand, has infinitely many solutions, two of which are shown in Figure 4.1. We see that distributional t-SNE preserves the 2-Wasserstein distance between distributions from the high-dimensional space, by adjusting the mean and the variance of the 2D distributions. The non-linearity of t-SNE, despite adding additional expressivity, may cause the 2D space to contain properties and relationships that do not exist in the high-dimensional space. With a large number of data points, however, this does not present as much of a problem.

## 4.2 Performance on Evaluation Metrics

In the following tables, we assess the performance of our models on the discussed evaluation metrics. In these tables, the column "Distance metric" refers to the distance metric used for computing pairwise distances within a model.

With regards to the Spearman's rank correlation coefficients, all the PCA models outperform the t-SNE models (Table 4.1). This difference can be attributed to the fact that the Spearman's rank correlation coefficient is a global metric, and PCA is designed to optimize global variance.

| **Model** | **Distance metric** | **Spearman's $\rho$-E** | **Spearman's $\rho$-W** |
|---|---|---|---|
| Standard PCA | Euclidean | **0.4173** | **0.4185** |
| Distributional PCA | Euclidean | 0.3661 | 0.3722 |
| Distributional PCA | Wasserstein | 0.4022 | 0.4174 |
| Standard t-SNE | Euclidean | 0.2750 | 0.3058 |
| Distributional t-SNE | Euclidean | 0.3002 | 0.3512 |
| Distributional t-SNE | Wasserstein | 0.3061 | 0.3597 |

Table 4.1: Spearman's rank correlation coefficients or Spearman's $\rho$'s per model. $\rho$-E refers to $\rho$ computed with original distances in Euclidean space, and $\rho$-W refers to $\rho$ computed with original distances in Wasserstein space.

Counterintuitively, however, distributional PCA shows a lower correlation with the original distances than standard PCA, even when evaluating with the Wasserstein distance. To validate the correctness of our PCA extension, we computed the reconstruction errors of both standard PCA and distributional PCA in Table 4.2. Indeed, standard PCA is optimal in terms of minimizing the Euclidean reconstruction error, while distributional PCA is optimal in terms of minimizing the Wasserstein reconstruction error.

| | Standard PCA | Distributional PCA |
|---|---|---|
| Euclidean | **14.99** | 15.72 |
| Wasserstein | 30.77 | **22.06** |

Table 4.2: Euclidean and Wasserstein reconstruction errors.

The Euclidean reconstruction error is defined as:

$$\frac{1}{N}\sum ||x - U^\top Ux||_2^2$$

while the Wasserstein reconstruction error is equal to:

$$\frac{1}{N}\sum\left(||x - U^\top Ux||_2^2 + \mathcal{B}^2(\Sigma, U\Sigma U^\top)\right)$$

We suspect that the reconstruction error is not necessarily correlated with the Spearman's $\rho$.

Table 4.3 shows the average nDCG scores per model with $k$, as defined in Section 3.3.2, set to 10. The t-SNE models are markedly better than the PCA models for this metric, which is exactly as expected, as the nDCG metric virtually corresponds to t-SNE's optimization objective. Within the t-SNE models, distributional t-SNE outperforms the other models with original distances in Wasserstein space. Based on these observations our extended method was optimal for preserving the 2-Wasserstein distances between the closest neighboring distributions. When distances are in Euclidean space, on the other hand, standard t-SNE has a higher average nDCG score than the other models.

| **Model** | **Distance metric** | **Average nDCG-E** | **Average nDCG-W** |
|---|---|---|---|
| Standard PCA | Euclidean | 0.0773 | 0.0796 |
| Distributional PCA | Euclidean | 0.0569 | 0.0585 |
| Distributional PCA | Wasserstein | 0.0889 | 0.1021 |
| Standard t-SNE | Euclidean | **0.7699** | 0.7735 |
| Distributional t-SNE | Euclidean | 0.7587 | 0.7723 |
| Distributional t-SNE | Wasserstein | 0.7694 | **0.7816** |

Table 4.3: Average nDCG scores per model. The relevance scores per word can range from 1 to $k$. nDCG-E refers to the nDCG score computed with original distances in Euclidean space, and nDCG-W refers to the nDCG score computed with original distances in Wasserstein space.

As described in Section 3.3.2, we also compute the average nDCG scores per model where the true relevance scores are calculated from the full ranking. Table 4.4 shows these results. This change in computation drastically reduces the penalty of not having retained many of the closest $k$ neighbors from the original space. As anticipated, the t-SNE models exhibit inferior performance on this metric compared to the previous metric. Surprisingly, however, distributional PCA with Euclidean distances achieves the best results in both columns.

| Model | Distance metric | Average nDCG-E | Average nDCG-W |
|---|---|---|---|
| Standard PCA | Euclidean | 0.7734 | 0.7716 |
| Distributional PCA | Euclidean | **0.7814** | **0.7788** |
| Distributional PCA | Wasserstein | 0.7636 | 0.7546 |
| Standard t-SNE | Euclidean | 0.5370 | 0.5379 |
| Distributional t-SNE | Euclidean | 0.5388 | 0.5370 |
| Distributional t-SNE | Wasserstein | 0.5373 | 0.5363 |

Table 4.4: Average nDCG scores per model using the full ranking, as described in Section 3.3.2. The relevance scores per word can range from 1 to $N$, the vocabulary size. nDCG-E refers to the nDCG score computed with original distances in Euclidean space, and nDCG-W refers to the nDCG score computed with original distances in Wasserstein space.

To assess the impact of the parameter $k$, we examine how the average nDCG scores of the models change as $k$ increases, shown in Figure 4.2.

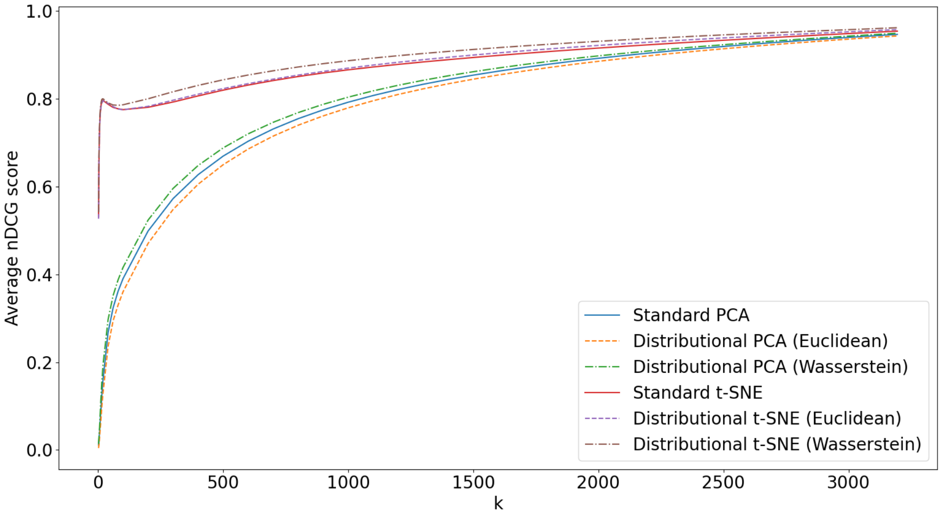


Figure 4.2: Trends of average nDCG scores per model as $k$ increases. The scores are computed with original distances in Wasserstein space.

The t-SNE models attain higher average nDCG scores than the PCA models for all values of $k$, until the scores of all models eventually converge at $k$ = vocabulary size. Both distributional PCA and distributional t-SNE outperform their original counterparts in Wasserstein space. Around $k = 20$, we see a notice-able drop-off for the scores of the t-SNE models. This can be attributed to the perplexity value, which was set to 20 in all of our t-SNE experiments.

## 4.3 Visualizations of Distributions

To determine whether or not distributional PCA plots provide any advantages compared to standard PCA visualizations, we show a side-by-side plot of a couple words in Figure 4.3.

The distributional PCA plot gives us two added dimensions of information that are not accessible by standard PCA. Since distributional PCA attempts to replicate the variance from the high-dimensional space, we can say that the sizes of the ellipses in the plot are related to the sizes of the high-dimensional distributions.

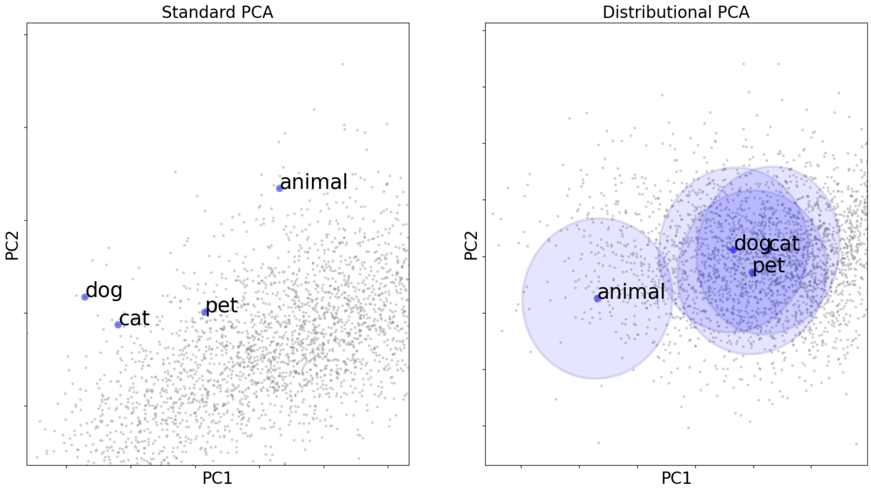


Figure 4.3: 2D scatterplot of a small set of words learned by PCA on means (left) and PCA on distributions (right). The gray dots in the plot are other words in the vocabulary that have been omitted for visibility. Coordinate values on the axes are hidden because we are interested in relative distances rather than numerical distances.

Moreover, when we examine the areas of the ellipses, we see that the ellipses of words that are very specific in meaning tend to be small in size. Accordingly, ellipses of words that are more ubiquitous tend to be large in size, as seen in Figure 4.4. Taking an example from this visualization, the word "enzymology", which refers to the field of study that deals with enzymes, has the smallest ellipse in our data set. The word "do", on the other hand, has one of the largest ellipses in our data set.

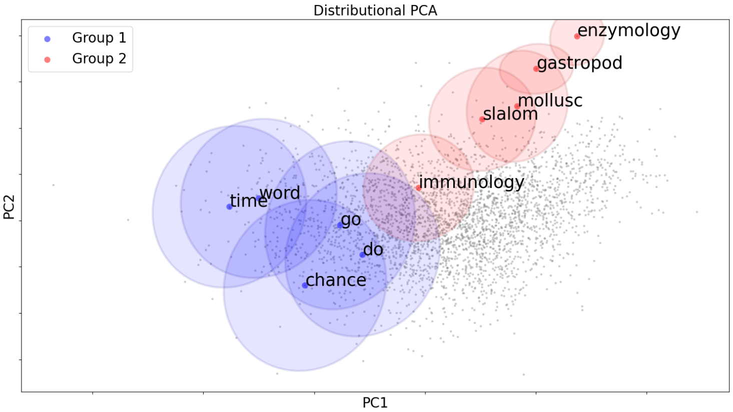


Figure 4.4: 2D scatterplot of two subsets of our vocabulary of word embeddings learned by distributional PCA. Group 1 contains 5 words with some of the largest ellipses, while Group 2 contains 5 words with relatively small ellipses. The gray dots in the plot are other words in the vocabulary that have been omitted for visibility. Coordinate values on the axes are hidden because we are interested in relative distances rather than numerical distances.

The size of an ellipse depends on the size of its high-dimensional distributions, which in turn depends on the contexts that served as input for the RoBERTa model. For instance, if a word appears in a wide variety of contexts, it will be assigned a relatively large covariance by the RoBERTa model, and our extended techniques try to preserve this large covariance. This also means that a word with multiple meanings is more likely to have a distribution with high covariance, and thus more likely to have a large ellipse in the low-dimensional space.

In Figure 4.5, we have another side-by-side plot with the same set of words, but generated by t-SNE models. Unlike in distributional PCA, we see that the ellipses in distributional t-SNE are axis-aligned because of diagonal covariances. Again, we have more information from the extended technique than we have in standard t-SNE.

Comparing the ellipses in this plot to the ellipses in Figure 4.3, we notice that the variances of the words are different. Just as we demonstrated in our synthetic experiments, distributional t-SNE can adjust both the mean and the variance in order to optimize its solution.

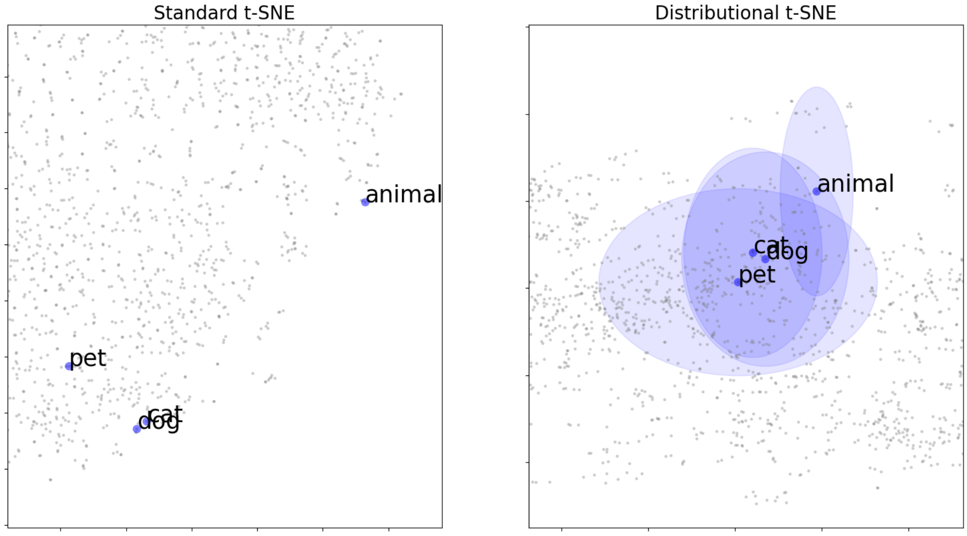


Figure 4.5: 2D scatterplot of a small set of words learned by t-SNE on means (left) and t-SNE on distributions (right). The gray dots in the plot are other words in the vocabulary that have been omitted for visibility. Coordinate values on the axes are hidden because we are interested in relative distances rather than numerical distances.

Overall, it is evident that the visualizations from our extensions contribute more information than the visualizations produced by standard PCA and t-SNE. The low-dimensional ellipses offer a clear depiction of the high-dimensional variance.

# Chapter 5

# Conclusion

We have shown extensions of PCA and t-SNE that allow for Gaussian distributions as input. Our methods enable visualizations that faithfully represent the original space, and can describe the high-dimensional variance of data points through the sizes of ellipses. Examining the visualizations produced by our techniques can offer a deeper understanding of the complicated space, providing valuable information that is otherwise inaccessible through standard PCA and t-SNE.

The strength of distributional t-SNE lies in its expressiveness and in its ability to maintain neighboring distributions from the original space. Its non-linearity, however, may sometimes cause the reduced space to contain properties and relationships that do not exist in the high-dimensional space. The portrayals of ellipses in distributional PCA are more reliable in comparison. Furthermore, while distributional PCA does not boast t-SNE's expressivity of non-linearity, it does have the benefit of expressivity via full covariance matrices, allowing for rotated ellipses. In summary, together, distributional PCA and distributional t-SNE are powerful tools for analyzing and interpreting high-dimensional data.

# Chapter 6

# Future Research

We have shown that the extended methods can be applied to probabilistic word embeddings. In practice, these extended methods can be used on any data set grouped by individuals. If each individual has multiple entries in the data set, we can derive a mean and covariance for each of them, forming the parameters of a multivariate Gaussian distribution per individual.

For example, an application could be a data set containing physiological measurements of patients in a hospital. Every patient would have multiple measurement rows from which we can derive a mean and covariance. By visualizing this data set using our extensions, we can group patients with similar measurements, as well as represent each patient with an ellipse describing how much their measurements may vary. Depending on the type of structure we want to retain, we can choose either distributional PCA or distributional t-SNE for the visualization method.

Regarding future research, a promising direction is the implementation of distributional t-SNE with full covariances. This would enable ellipses to take any shape, leading to more expressivity in the visualizations. This could, for example, be achieved by learning the angle $\theta$ directly during the optimization process. Additionally, there is potential value in extending other dimensionality reduction algorithms besides PCA and t-SNE for distribution data, such as minimum-distortion embedding (Agrawal et al., 2021, MDE).

In conclusion, this research has shed light on the potential of extending dimensionality reduction techniques for probabilistic embeddings. There are plenty of exciting avenues to explore, any of which may lead to novel ways to analyze and interpret distribution data.

# Bibliography

Agrawal, A., Ali, A. and Boyd, S. (2021). Minimum-distortion embedding.

Axler, S. J. (1997). *Linear Algebra Done Right*, Undergraduate Texts in Mathematics, Springer, New York.

Bertsekas, D. P. (1997). Nonlinear programming, *Journal of the Operational Research Society* 48(3): 334–334.

Bhatia, R., Jain, T. and Lim, Y. (2018). On the bures-wasserstein distance between positive definite matrices, *Expo. Math* 37(2): 165–191.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Gubanov, M. N. and Pyayt, A. (2016). Type-aware web-search., *EDBT*, pp. 656–657.

Hinton, G. and Roweis, S. (2002). Stochastic neighbor embedding, *in* S. Becker, S. Thrun and K. Obermayer (eds), *Advances in Neural Information Processing Systems*, Vol. 15, MIT Press.

Hinton, G. and van der Maaten, L. (2008). Visualizing data using t-SNE, *Journal of Machine Learning Research* 9(86): 2579–2605.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques, *ACM Trans. Inf. Syst.* 20(4): 422–446.

John, S. (1968). A central tolerance region for the multivariate normal distribution, *Journal of the Royal Statistical Society. Series B (Methodological)* 30(3): 599–601.

Joswiak, M., Peng, Y., Castillo, I. and Chiang, L. H. (2019). Dimensionality reduction for visualizing industrial chemical process data, *Control Engineering Practice* 93: 104189.

Lee, J. and Verleysen, M. (2007). Nonlinear dimensionality reduction.

Liu, S., Bremer, P.-T., Thiagarajan, J. J., Srikumar, V., Wang, B., Livnat, Y. and Pascucci, V. (2018). Visual exploration of semantic relationships in neural word embeddings, *IEEE Transactions on Visualization and Computer Graphics* 24(1): 553–562.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach, *CoRR* abs/1907.11692.

Meng, Y., Huang, J., Wang, G., Zhang, C., Zhuang, H., Kaplan, L. and Han, J. (2019). Spherical text embedding.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality.

Murphy, K. P. (2022). *Probabilistic Machine Learning: An introduction*, MIT Press.

Muzellec, B. and Cuturi, M. (2019). Generalizing point embeddings using the wasserstein space of elliptical distributions.

Myers, J. and Well, A. (2003). Research design and statistical analysis.

Niculae, V. (2023). Two derivations of principal component analysis on datasets of distributions.

Pearson, K. (1901). LIII. 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.

Peyré, G., Cuturi, M. et al. (2019). Computational optimal transport: With applications to data science, *Foundations and Trends in Machine Learning* 11(5-6): 355–607.

Raunak, V. (2017). Simple and effective dimensionality reduction for word embeddings, *arXiv:1708.03629*.

Ringnér, M. (2008). What is principal component analysis?, *Nature biotechnology* 26(3): 303–304.

Tol, R. (2023). Word-level entailment with probertilistic word embeddings.

Vilnis, L. and McCallum, A. (2015). Word representations via gaussian embedding, *arXiv:1412.6623*.

Young, J. C. and Rusli, A. (2019). Review and visualization of facebook's fasttext pretrained word vector model, *2019 International Conference on Engineering, Science, and Industrial Applications (ICESI)*, pp. 1–6.