Representing Hierarchical Structure by Using Cone Embedding

Daisuke Takehara\(^1,2\)
Kei Kobayashi

\(^1\)Department of Mathematics, Keio University, Yokohama, Kanagawa, Japan
\(^2\)ALBERT.inc, Shinjuku, Tokyo, Japan,

Abstract

Graph embedding is becoming an important method with applications in various areas, including social networks and knowledge graph completion. In particular, Poincaré embedding has been proposed to capture the hierarchical structure of graphs, and its effectiveness has been reported. However, most of the existing methods have isometric mappings in the embedding space, and the choice of the origin point can be arbitrary. This fact is not desirable when the distance from the origin is used as an indicator of hierarchy, as in the case of Poincaré embedding. In this paper, we propose cone embedding, embedding method in a metric cone, which solve these problems, and we gain further benefits: 1) we provide an indicator of hierarchical information that is both geometrically and intuitively natural to interpret, and 2) we can extract the hierarchical structure from a graph embedding output of other methods by learning additional one-dimensional parameters.

In recent years, machine learning for graph-structured data has attracted significant attention. A recently developed method is using graph convolutional neural networks [Kipf and Welling, 2017], which apply convolutional neural networks to graphs. In this study, we focus on graph embedding, which has become an important method with applications in social networks [Hoff et al., 2002], knowledge graph completion [Bordes et al., 2013], and other fields. In particular, various methods of embedding in non-Euclidean spaces have been proposed to capture the hierarchical structure of graphs. In Klimovskaia et al. [2020], Poincaré embedding [Nickel and Kiela, 2017] has been applied to extract hierarchies from biological cell data.

However, there are two major problems with existing graph embedding methods that are used for extracting the hierarchical structures.

The first problem concerns the selection of the origin point. The distance from the origin is used as an indicator of hierarchy when applying the embedding method to the space where an isometric map exists. In non-Euclidean spaces, existing graph embedding methods use a loss function that depends on the distance between two nodes in the embedding. Because the distance from the origin is changed by the isometric transformation as the value of the loss function is not changed, it is not appropriate to use the distance from the origin as an indicator of hierarchy. In other words, when we define the indicator of the hierarchy in the space where the isometric map exists, it is necessary to be invariant to the isometric map.

The second problem is a lack of extensibility. If we need to solve a problem by adding hierarchical information to a graph embedding that has already been learned for another purpose, you will need to learn another embedding method or solve the problem as an independent process. Therefore, we cannot directly apply existing embedding method for extracting the hierarchical structure.

In this paper, we propose a method of embedding graphs in metric cones as a solution to the two problems described above. A metric cone can be defined on any length metric space—a wide class of metric spaces (including normed vector spaces, manifolds, and metric graphs). This space’s dimension is larger than that of the original space. It is known that the curvature of this space can be varied, and a method of changing the structure of the data space for analysis has also been proposed [Kobayashi and Wynn, 2020].

First, we show that it is effective to use the coordinate corresponding to “the height of the metric cone” (a one-dimensional parameter added to the original space) as an indicator of hierarchy. Second, we demonstrate that the hierarchical structure can be extracted by learning the embedding of the metric cone, which optimizes the only one-dimensional parameter, in case we have a graph embedding pretrained. By keeping the coordinates of the original space fixed, we show that only the one-dimensional parameter...
corresponding to the height of the cone is learned, and the computation time is reduced.

In cone embedding, we show that the curvature of the embedding space can be optimized by data driven method, like Wilson et al. [2014] and Chami et al. [2019]. The curvature of a metric cone varies with one parameter that corresponds to the generatrix of the cone. If we have the pretrained embedding, we show that the learning method is also computationally efficient, because curvature can be optimized by learning one-dimensional parameters without changing the structure of the original space (i.e., without learning the embedding in the original space).

The remainder of this paper is organized as follows. Section 2 describes related research. In Section 3, we propose the method of graph embedding in a metric cone, with the introduction of 1) graph embedding in non-Euclidean spaces and 2) the definition and properties of metric cones. Section 4 presents the experimental results using graph data, followed by a conclusion and future perspectives in Section 5.

1 RELATED WORK

1.1 RANDOM WALK MODELS

DeepWalk [Perozzi et al. 2014] samples series by random walks on graphs and applies a method for embedding series data, such as the skip-gram model [word2vec Mikolov et al. 2013]. The sampling method of DeepWalk’s random walk series has been modified to depend on edge weights in node2vec [Grover and Leskovec 2016], and a method that simultaneously optimizes hyperparameters (such as random walk length) has been proposed in Abu-El-Haija et al. [2018]. Unlike our proposed method, these methods were not designed to represent a hierarchical structure but were simply designed to embed the structure of a graph with high accuracy, which is evaluated by predicting edge links.

1.2 DIMENSIONALITY REDUCTION MODELS

Dimensionality compression methods exist for data in Euclidean space, such as multidimensional scaling (MDS) [Kruskal 1964], IsoMAP [Tenenbaum et al. 2000], and locally linear embedding (LLE) [Roweis and Saul 2000]. By applying it to the adjacency matrix of a graph, an embedded representation of the graph can be obtained. There are also methods to obtain an embedded representation of a graph by applying dimensionality reduction to the graph’s Laplacian matrix [Belkin and Niyogi 2002] instead of the adjacency matrix. As an alternative to random walk models, these graph embedding methods are also used to accurately embed the structure of the graph but not to represent a hierarchical structure.

1.3 GRAPH NEURAL NETWORKS

Methods to compress dimensionality of adjacency matrices [Cao et al. 2016] [Wang et al. 2016] or the graph’s Laplacian matrices [Kipf and Welling 2016] using a neural network (autoencoder) to obtain a representation of the graph embedding have been proposed. These methods can be easily applied to other tasks. However, as with the above categories, these methods are designed for simply embedding the structure of a graph but not for representing a hierarchical structure.

1.4 NON-EUCLIDEAN MODELS (HYPERBOLIC EMBEDDING MODELS)

Poincaré embedding [Nickel and Kiela 2017] is a method to embed the adjacency matrix of a graph in a skip-gram model, large-scale information network embedding (LINE) [Tang et al. 2015], which is constructed on a Poincaré sphere. In addition, there are other methods to embed graphs into Lorentz models [Nickel and Kiela 2018] and to embed each node of a graph as a cone instead of a point [Ganea et al. 2018]. These methods exist to describe the hierarchy of each graph according to the inclusion relation of the cones. In previous research, the hierarchical structure could be accurately captured, but the problem of invariance for isometric maps prevented the natural definition of a metric that would represent the hierarchical structure. Therefore, embedding parameters in non-Euclidean spaces makes them difficult to use in other tasks. In contrast, our proposed method can use a Euclidean embedded representation of all parameters except for the one-dimensional parameters. The purpose is to provide an embedding method that is easy to use for other tasks and provides a natural indicator of hierarchical information.

2 METHODS

From this point onwards, the set of edges in an undirected graph $G$ is denoted by $E$, the set of vertices by $V$, and the embedded space by $X$.

2.1 GRAPH EMBEDDING IN NON-EUCLIDEAN SPACES

Following Poincaré embedding, we learn the embedding of a graph $G$ by maximizing the following objective function:

$$L = \sum_{(u,v) \in E} \log \frac{\exp(-d(u,v))}{\sum_{v' \in N^c(u)} \exp(-d(u,v'))},$$  \hspace{1cm} (1)$$

where $N^c(u) : \{v' \in V | (u,v') \notin E\}$ denotes the set of points not adjacent to node $u$ (including $u$ itself), and $d$ denotes the distance function of the embedded space (for
With the metric tensor of the embedding space as $\beta > 0$, where $\eta > 0$.

Length metric space is a metric space where $\eta$. The metric cone has been studied as an analogy to the length metric space, including Euclidean spaces, normed vector spaces, manifolds and the tangent cone for differential manifolds.

The metric cone is similar to ordinary cones (e.g. circle cones) in the sense that it is defined as a collection of line segments connecting an apex point to a given set. However, the metric cone has a notable property such that every point on the metric cone has a distance from the apex point and this is a desirable property for hierarchical structure extraction.

The metric cone has been studied as an analogy to the length metric spaces of the tangent cone for differential manifolds. However, an isometric transformation on Poincaré ball can be described as Algorithm 1. Instead of the exponential map of metric cone, we use the first-order approximation using $\text{proj}(x, s)$:

$$\tilde{g}(x, s) = \begin{pmatrix} \pi^2 s^2 g_x & 0 \\ 0 & \beta^2 \end{pmatrix},$$

where $g_x$ represents the metric of $Z$ at $x$. Combining this metric and the argument in Sec 3.1, the algorithm of cone embedding can be described as Algorithm 1. Instead of the exponential map of metric cone, we use the first-order approximation using $\text{proj}(x, s)$:

$$\text{proj}(x, s) = \begin{cases} (x, s) & \text{if } \epsilon < s < 1 - \epsilon, \\ (x, 1 - \epsilon) & \text{if } s \geq 1 - \epsilon, \\ (x, \epsilon) & \text{if } s \leq \epsilon. \end{cases}$$  

### 2.2.1 Score Function of Hierarchy

The Poincaré embedding defines an index in [Nickel and Kiela 2017], which is aimed to be an indicator of the hierarchical structure, that depends on the distance from the origin:

$$\text{score}(u, v) = -\alpha(\|v\| - \|u\|)d(u, v)$$

This score function is penalized by the part after $\alpha$, so if $v$ is closer to the origin than $u$, then it is easier to obtain larger values. In other words, $v$ is higher in the hierarchy than $u$ (i.e., “$u$ is a $v$” relationship holds). However, it is not appropriate to use this indicator for the Poincaré embedding. This model learns the embedding by maximizing (1), where

$$d(x, y) := \text{arccosh} \left(1 + \frac{||x - y||^2}{(1 - ||x||^2)(1 - ||y||^2)}\right).$$

This loss function only depends on the distance between the two embeddings. However, an isometric transformation in Poincaré ball exists, known as M"obius transformation [Loustau 2020]. M"obius transformation is defined as a map $f : \mathbb{B}^n$ (open unit ball) $\to \mathbb{B}^n$, which can be written as a product of the inversions of $\mathbb{R}^n := \mathbb{R}^n \cup \{\infty\}$ through a sphere $S$ that preserves $\mathbb{B}^n$.

In contrast to Poincaré ball, the isometric transformation on the metric cone does not exist when the coordinate in the
As mentioned above, for Poincaré embedding, there is an isometric transformation on the Poincaré ball, and the heights of the learned hierarchy are not invariant to such transformation. Here, we show that such a phenomenon does not occur for the cone embedding, i.e., the heights of the hierarchy are (almost) uniquely determined from the distance between the embedded data points in a metric cone.

Let $Z$ be an original embedding space (connected length metric space), and let $X$ be a metric cone of $Z$ with a parameter $\beta > 0$. We assume that each data point $z_i \in Z$ ($i = 1, \ldots, n$) has its specific “height” $t_i \in [0, 1]$ in the metric cone $X$. Our proposed method embeds data points into a metric cone based on the estimated distances $d_{\beta}(x_i, x_j)$ ($i, j = 1, \ldots, n$) and tries to compute the heights $t_1, \ldots, t_n$ as a measure of the hierarchy level. However, it is not evident if these heights are identifiable only from the information of the original data points in $Z$ and the distances $d_{\beta}(x_i, x_j)$ ($i, j = 1, \ldots, n$) in the metric cone. The following theorem guarantees some identifiability.

Theorem 1  
(a) Let $n \geq 3$ and assume that $z_1, \ldots, z_n$ are not all aligned on a geodesic in $Z$. Then, the heights $t_1, \ldots, t_n$ are identifiable up to at most four candidates.
(b) Let $n \geq 4$ and assume $z_1, \ldots, z_n$ and $t_1, \ldots, t_n$ take “general” positions and heights, respectively. Then, the heights $t_1, \ldots, t_n$ are identifiable uniquely.
(c) If $d_{\beta}(z_i, z_j) \geq \frac{\pi}{2}$ for all $i, j = 1, \ldots, n, i \neq j$, then the heights $t_1, \ldots, t_n$ are identifiable uniquely.

A rigorous version of Theorem 1 including the precise meaning of “general” in (b), is explained in supplementary materials. Theorem 1(a) indicates that the candidates of heights are finite, and we can expect the algorithm to converge to one of them, except for a very special data distribution in the original space $Z$. Moreover, by (b), even the uniqueness can be proved under very mild conditions. The statement in (c) implies that the uniqueness holds for arbitrary data distributions when we set $\beta$ sufficiently small.

Remark that the assumption of “general” positions in Theorem 1(b) is satisfied easily for most data distributions. For example, if both $z_1, \ldots, z_n \in \mathbb{R}^d$ and $t_1, \ldots, t_n \in [0, 1]$ are i.i.d. from a probability distribution whose density function exists with respect to the Lebesgue measure, then it is
easy to see the assumption holds almost surely and therefore uniqueness of the solution is guaranteed. Note that for \( n = 3 \) under the same setting, there can be multiple solutions with a positive probability.

### 2.2.3 Using Pretrained Model for Computational Efficiency and adaptivity for adding hierarchical information

Consider a situation where we already have a trained graph embedding on a Euclidean space (e.g. LINE [Tang et al. 2015]), and we try to learn the embedding in a metric cone of the Euclidean space to extract information about the hierarchical structure. In this case, we can reduce the computational cost by fixing the coordinates corresponding to the original Euclidean space and learn only the one-dimensional parameters corresponding to heights in a metric cone added to the original space because the metric cone is one dimension larger than the original space. The distance between each embedding in the original space is calculated beforehand, since no updates are made by learning except for the 1D parameter to be added. By referring to the pre-computed distances in the original space when calculating the distances between each embedding on the metric cone (\( d_z(x, y) \) in Eq. (6)) during training, we can reduce the amount of computation for the parts other than preprocessing, regardless of the dimension. In addition, because the embedding in the original Euclidean space is preserved, it can be used as an input to the neural network (when the task considers information about the hierarchy, and the added one-dimensional parameters are also used as input) and can be easily applied to other tasks. However, other non-Euclidean embedding methods to extract hierarchical structures are not scalable because these methods cannot be applied directly to solve other tasks. For example, deep neural networks cannot use a non-Euclidean embedding as input because the sum of two vectors in the space and scalar product is not generally defined.

### 2.2.4 Comparison with hierarchical clustering

It has been shown by Sarkar [2011] that the embedding of tree-structured (undirected) graph data can be done naturally in hyperbolic space, but graph data with hierarchical structure does not necessarily have a tree structure in general. However, graph data with a hierarchical structure does not have a tree structure in general (e.g. there can be a cycle when a child node has two parents which have the same parent.), so the combination with hyperbolic embedding and hierarchical clustering may not be suitable in such cases. Cone embedding does not assume the tree structure and extracts the hierarchical structure by using the property that the closer to the origin \( O \), the shorter the distance of between data points, so that embedding can be learned even in this situation.

### 2.2.5 Variable Curvature

One of the essences of Poincaré embedding is that a negative curvature of the Poincaré sphere is suitable for embedding tree graphs. The curvature of a metric cone has a similar property, i.e. a metric cone has a more negative curvature than the original space and, furthermore, the curvature can be controlled by hyperparameter \( \beta \). We will verify these facts mathematically from two different aspects: (i) the scalar and the Ricci curvatures of a Riemannian manifold and (ii) the \( \text{CAT}(k) \) property of a length metric space.

First assume the original space \( M \) is a \( n \)-dimensional Riemannian manifold with a metric \( g \). Then the metric \( \tilde{g} \) of the corresponding metric cone with \( \beta \) can be defined except the apex and it becomes as (\( \emptyset \)). Let \( 1, \ldots, n \) be coordinate indices corresponding to \( x \in M \) and 0 be the index corresponding to \( s \in [0, 1] \). The Ricci curvatures \( \tilde{R}_{ij} \) and the scalar curvature \( \tilde{R} \) at \( (x, s) \) become

\[
\tilde{R}_{\alpha\gamma} = R_{\alpha\gamma} - \pi^2 (n - 1) \beta^{-2} \tilde{g}_{\alpha\gamma},
\]

\[
\tilde{R}_{\alpha 0} = \tilde{R}_{0\alpha} = \tilde{R}_{00} = 0,
\]

\[
\tilde{R} = \pi^2 R - n(n - 1) \beta^{-2} s^{-2}
\]

where \( \alpha, \gamma \) are coordinate indices in \( 1, \ldots, n \) and \( R_{ij} \) and \( R \) are the Ricci curvatures and the scalar curvature of \( M \), respectively. See the supplementary materials for the derivation of such curvatures. The scalar curvature and the Ricci curvatures \( \tilde{R}_{\alpha\gamma} \) becomes more negative than (a constant times of) the original curvature for \( \beta < \infty \) and \( n \geq 2 \). Moreover, the smaller value of \( \beta \) makes the curvature more negative thus it becomes possible to control the curvature by tuning \( \beta \). Note that the closer to the apex, i.e. the smaller the value of \( s \), the greater the change of the scalar curvature.

Second assume the original space \( M \) is a length metric space. This doesn’t require a differentiable structure and is more general than the Riemannian manifold. In this case, we cannot argue the curvatures using the Riemannian metric but the \( \text{CAT}(k) \) property can be used instead. In [Kobayashi and Wynn 2020], they proved the curvature of the metric cone is more negative or equal to the curvature of the original space and it can be controlled by \( \beta \) in the sense of the \( \text{CAT}(k) \) property.

### 3 EXPERIMENTS

The claim in this paper is that “a hierarchical structure can be captured by adding a one-dimensional parameter and embedding it in a metric cone.” Therefore, we evaluate the proposed method in two experiments:

- prediction of edge direction for artificial directed graphs
- estimation of hierarchical score by humans for WordNet.
As a comparison, we compare the proposed method with three other methods: Poincaré embedding [Nickel and Kiela 2017], and ordinary embedding in Euclidean space, which are known to capture the hierarchical structure of graphs. For the Euclidean embedding, we use the distance from the mean of embedded data points as the hierarchical score in (8).

### 3.1 Prediction of Edge Direction for Directed Graphs

In this experiment, we estimate the orientation of directed edges for some simple graphs. We use the following three patterns of graphs with a naturally set hierarchical structure:

- Graphs generated by a growing random network model called Barabási–Albert preferential attachment [Barabasi and Albert 1999] with $m = 2$, where $m$ is the number of edges to attach from a new node to existing nodes
- Complete $k$-ary tree
- Concatenated tree of two complete $k$-ary sub-trees

For each tree, node depth can be treated as its hierarchy. The concatenated tree is created by connecting the roots of two complete $k$-ary trees to a new node, which is then used as the new root. The concatenated tree is considered to study the effect of node degree on the cone embedding as will be explained below.

In this experiment, we learn the embedding of each undirected graph. For each undirected edge of the learned graph, we estimate the direction by the computed hierarchical scores $\text{score}(u, v)$:

$$\text{total score} = \sum_{(u, v) \in E} \frac{\text{score}(u, v)}{|E|} \quad (14)$$

$$\text{score}(\text{hypo}, \text{hype}) = \begin{cases} 
1 & \text{score}(\text{hypo}, \text{hype}) > 0 \\
0 & \text{otherwise} 
\end{cases} \quad (15)$$

where hypothesis: higher hierarchy node

hypo: lower hierarchy node

The experimental results are shown in Table[1] and the cone embedding shows overall good and stable estimation accuracy for hierarchies. Examples in Figure[2] depict how each graph is embedded in a metric cone. The reason for the instability in the accuracy of Poincaré embedding may be the lack of invariance with respect to the equidistance transformation, as we have explained. The main reason for the poor hierarchical estimation accuracy of Euclidean embedding is that the root or higher hierarchical nodes are embedded apart from the cluster of other nodes as in Figure[3]. As a result, the root becomes far from the mean of the embedded points.

For the Barabási–Albert model, the relationship between the (added 1D) coordinates of the cone hierarchy and the order is shown in Figure[3]. We can see that there is a strong relationship between the degree and the hierarchy. This raises the suspicion that the degree of the node alone determines the hierarchy of the embedding. However, the fact that cone embedding provides high estimation accuracy even for the concatenated trees with low root degree indicates that this is not true.

### 3.2 Embedding Taxonomies

Following an experiment in [Nickel and Kiela 2017] for the Poincaré embedding, we evaluate the embedding accuracy of the hierarchical structure using WordNet. To verify this, we embed the nouns in WordNet into a metric cone and use the score function (10). Note that hyperparameter $\alpha$ was set to 10. The output of this score function and the correlation coefficient of the HyperLex dataset’s score (evaluated manually whether a word is a subword of another word) are used to evaluate the ability to represent the hierarchical structure of the model. Also, we evaluate the effectiveness of proposed method by two setups, reconstruction and link prediction. In the reconstruction setup, all of the graph data are used for training, and results are evaluated according to the accuracy with which the graph is reconstructed from the learned embedding. Because the same data are used for training and evaluation, we evaluate the fittingness of the embedding method to the data (not generalization performance). In the link prediction setup, edges are randomly removed for training. In the evaluation, we use the same criteria as in the reconstruction setting to evaluate the accuracy of the removed edges. The accuracy is calculated as follows:

1. Fix one node and calculate the distance to all other nodes. Sort the nodes in order of proximity.
2. We sort all nodes except the reference node in order of distance proximity, and calculate the average value and average precision of the rankings of neighboring nodes.
3. Perform the above two operations on all the nodes and take the average.

since their experiments reported better accuracy when euclidean embedding is trained without normalization to the unit vector after updating embedding by Riemannian SGD, we believe that this is due to this effect. But since there is a hierarchy in the graph to be embedded, only the edges connected to nodes that are neither root nor leaf are split into test data in the link prediction setting. The embedding accuracy (mean rank (MR) and mean average precision (MAP)) and correlation coefficients are also shown in Table[2]. The table shows that our proposed model improves the score and captures the hierarchical structure better than other embedding methods. Also, the time taken to learn the embedding
Figure 2: Graphs used for training: (left) model trained by Barabási–Albert model, (middle) complete $k$-ary tree, (right) concatenated tree of two complete $k$-ary trees. The $x$- and $y$-axes represent embedding in Euclidean space, which is dimensionally reduced to two dimensions by principal component analysis, and the $z$-axis represents height in metric cone (coordinates representing hierarchy).

Table 1: Result for prediction of edge direction for directed graphs. (we list accuracy and standard deviation)

| Model                        | Barabási–Albert (100 nodes) | complete $k$-ary-tree $k=3$ (121 nodes) | complete $k$-ary-tree $k=5$ (781 nodes) | Concatenated tree of two $k$-ary trees $k=3$ (81 nodes) | Concatenated tree of two $k$-ary trees $k=5$ (313 nodes) |
|------------------------------|-----------------------------|----------------------------------------|----------------------------------------|-----------------------------------------------------|-----------------------------------------------------|
| Metric Cone                  | 0.936(se:0.005)             | 0.787(0.049)                          | 0.799(0.037)                          | 0.783(0.045)                                        | 0.744(0.056)                                        |
| Euclidean                    | 0.181(0.004)                | 0.074(0.088)                          | 0.127(0.020)                          | 0.190(0.136)                                        | 0.155(0.031)                                        |
| Poincaré                     | 0.957(0.012)                | 0.935(0.015)                          | 0.351(0.022)                          | 0.880(0.060)                                        | 0.606(0.043)                                        |

Figure 3: Plot of the hierarchy value of each node in a cone embedding (a newly added one-dimensional parameter) against their node degree.

is shown in the same table. The time units are in CPU time, and all the results are from the reconstruction settings and link prediction experiment. From the table, we can see that our method is efficient in learning and does not vary with the dimension of the embedding. This is due to the fact that the distance matrix is calculated beforehand, which avoids repeating useless calculations.

Furthermore, an example visualization of the hierarchical structure of the embedding vectors obtained by the training is shown in Figure 5. As the figure illustrates, the closer the coordinate corresponding to the height in the cone is to zero (closer to the top of the cone), the higher the noun in the hierarchy is located in the embedded representation.

Figure 4: An embedding of the complete $k$-ary tree ($k = 3$). Each point is plotted by the 3D Euclidean embedding, and the color represents length of shortest path from root node (the bluer the color, the higher the hierarchy).
Table 2: MAP, Mean Rank score, Hyperlex score (correlation efficient) and computation time for the Reconstruction and Link Prediction setting in WordNet. Since the link prediction settings is to estimate generalization performance, the computation time and correlation with human score are omitted.

| Model    | evaluation metric | Reconstruction | Link Prediction |
|----------|-------------------|----------------|-----------------|
|          |                   | 10  | 20  | 50  | 100 | 10  | 20  | 50  | 100 |
| Euclidean| MR                | 3.16| 3.16| 3.16| 3.16| 29.37| 8.18| 8.16| 8.16|
|          | MAP               | 0.999| 0.999| 0.999| 0.999| 0.745| 0.990| 0.996| 0.997|
|          | comp. time        | 0.259| 0.423| 0.464| 0.478| -    | -   | -   | -   |
|          | corr              | 9958.1| 12126.0| 12062.2| 12043.5| -    | -   | -   | -   |
| Poincaré | MR                | 27.77| 26.62| 25.74| 26.12| 415.34| 389.34| 382.65| 380.41|
|          | MAP               | 0.879| 0.881| 0.883| 0.882| 0.721| 0.727| 0.729| 0.729|
|          | comp. time        | 0.297| 0.299| 0.290| 0.285| -    | -   | -   | -   |
| Cone     | MR                | 12250.1| 12152.4| 12180.1| 20795.6| -    | -   | -   | -   |
| (Our Model)| MAP              | 3.16| 3.16| 3.16| 3.16| 27.85| 8.18| 8.16| 8.16|
|          | corr              | 0.999| 0.999| 0.999| 0.999| 0.745| 0.990| 0.996| 0.997|
|          | comp. time        | 0.379| 0.470| 0.483| 0.490| -    | -   | -   | -   |
|          |                  | 10901.2| 11289.6| 11431.0| 12164.5| -    | -   | -   | -   |

For visualization, the embedding vectors in Euclidean space were reduced to two dimensions by principal component analysis.

Figure 5: Visualization of WordNet Embedding using metric cone: each point is a word, and the two points connected by directed edges indicate that the word at the end of the arrow is a hyponym of the word at the start.

4 DISCUSSION AND FUTURE WORKS

In this study, we have demonstrated that a graph embedding in a metric cone that is a dimension larger than the existing embedding methods has the following advantages: 1) we naturally define an index (score function) as an indicator of hierarchy, 2) the proposed method has some adoptivity since it can introduce the hierarchy into various pretrained models by learning only newly added 1D parameters, and 3) thus, the optimization is computationally inexpensive. By optimizing the 1D parameters, we have shown that the proposed method also has the flexibility to optimize the curvature to enhance the accuracy as well as other methods. Future research topics include 1) efficient optimization of curvature, 2) development of an embedding method to update existing embeddings in learning, and 3) discovery of applications to other tasks. In this study, we measured the improvement of the accuracy of curvature optimization by the grid search. However, more efficient methods, such as gradient-based methods, can be used to optimize the embedding space. We also learned embedding in a metric cone under the constraint of not updating existing embeddings. This constraint can cause the optimization to fall into a local solution. Therefore, to optimize the entire embedding, an efficient method of optimizing functions on a metric cone (or Riemannian manifold) should be developed in future work.

References

Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alexander A Alemi. Watch your step: Learning node embeddings via graph attention. In Advances in Neural Information Processing Systems, pages 9180–9190, 2018.

Albert-Laszlo Barabasi and Rita Albert. Albert, r.: Emergence of scaling in random networks. Science (New York, N.Y.), 286:509–12, 11 1999. doi: 10.1126/science.286.5439.509.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In Advances in neural information processing systems, pages 585–591, 2002.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In Advances in neural information processing systems, pages 2787–2795, 2013.
Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. In *AAAI*, volume 16, pages 1145–1152, 2016.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/0415740eea4d9decbc8da001d3fd805f-Paper.pdf.

David Cox, John Little, and Donal OShea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer Science & Business Media, 2013.

Michel Marie Deza and Elena Deza. *Encyclopedia of distances*. In *Encyclopedia of distances*, pages 1–583. Springer, 2009.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. volume 80 of *Proceedings of Machine Learning Research*, pages 1646–1655, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/ganea18a.html.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the american Statistical association*, 97(460):1090–1098, 2002.

Svante Janson. Riemannian geometry: some examples, including map projections. *Notes*, 2015. URL http://www2.math.uu.se/~svante/papers/sjN15.pdf.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *NIPS Workshop on Bayesian Deep Learning*, 2016.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *In International Conference on Learning Representations (ICLR)*, 2017.

Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature Communications*, 11(1):1–9, 2020.

Kei Kobayashi and Henry P Wynn. Empirical geodesic graphs and cat (k) metrics for data analysis. *Statistics and Computing*, 30(1):1–18, 2020.

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

Brice Loustau. Hyperbolic geometry. *arXiv preprint arXiv:2003.11180*, 2020.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in neural information processing systems*, pages 6338–6347, 2017.

Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the Lorentz model of hyperbolic geometry. volume 80 of *Proceedings of Machine Learning Research*, pages 3779–3788, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/nickel18a.html.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

Rik Sarkar. Low distortion delaunay embedding of trees in hyperbolic plane. In *International Symposium on Graph Drawing*, pages 355–366. Springer, 2011.

Karl-Theodor Sturm. Probability measures on metric spaces of nonpositive curvature. *Heat Kernels and Analysis on Manifolds, Graphs, and Metric Spaces: Lecture Notes from a Quarter Program on Heat Kernels, Random Walks, and Analysis on Manifolds and Graphs: April 16-July 13, 2002, Emile Borel Centre of the Henri Poincaré Institute, Paris, France*, 338:357, 2003.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.
A DERIVATION OF THE METRIC TENSOR OF A METRIC CONE

Let $\mathcal{M}$ be an $n$-dimensional Riemannian manifold with a metric $g$. Then the metric $\tilde{g}$ of the corresponding metric cone $\tilde{\mathcal{M}} = \mathcal{M}_\beta$ can be defined except the apex. Denote the square of the infinitesimal distance in $\tilde{\mathcal{M}}$ as $|d\tilde{s}|^2$, then

$$|d\tilde{s}|^2 = \tilde{d}_\beta((x,r), (x + dx, r + dr))^2$$

$$= \beta^2(2r^2 + 2rdr + dr^2 - 2(r + dr))$$

$$\approx \beta^2(2r^2 + 2rdr + dr^2 - 2(r^2 + rdr)$$

$$= \frac{1}{2}(\pi d_M(x, x + dx)/\beta, 1))$$

$$\approx \beta^2dr^2 + \pi^2r^2 \sum_{i,j}g_{ij}dx_idx_j$$

$$+ \pi^2rdr \sum_{i,j}g_{ij}dx_idx_j$$

$$\approx \left(\begin{array}{c} dx \\ dr \end{array}\right)^\top \left(\begin{array}{cc} \pi^2r^2g_{ij} & 0 \\ 0 & \beta^2 \end{array}\right) \left(\begin{array}{c} dx \\ dr \end{array}\right). \quad (16)$$

Therefore, the metric tensor $\tilde{g}$ becomes

$$\tilde{g} = \left(\begin{array}{cc} r^2\pi^2g & 0 \\ 0 & \beta^2 \end{array}\right). \quad (17)$$

B DERIVATION OF THE RICCI AND THE SCALAR CURVATURES OF A METRIC CONE

We will derive the Ricci and scalar curvatures of metric cone $\tilde{\mathcal{M}}$ Let 0, 1, ..., $n$ be the coordinate indices of metric cone $\tilde{\mathcal{M}}$ where 0 corresponds to the radial coordinate $s \in (0, 1)$ and 1, ..., $n$ correspond to $x \in \mathcal{M}$.

Claim 2 The Ricci curvatures $\tilde{R}_{ij}$ and the scalar curvature $\tilde{R}$ become

$$\tilde{R}_{\alpha\gamma} = R_{\alpha\gamma} - \pi^2(n - 1){\beta^{-2}}g_{\gamma\gamma}, \tilde{R}_{\alpha0} = \tilde{R}_{0\alpha} = \tilde{R}_{00} = 0,$$

$$\tilde{R} = \pi^2R - n(n - 1){\beta^{-2}}s^{-2} \quad (18)$$

where $\alpha$ and $\gamma$ are coordinate indices in 1, ..., $n$ and $R_{ij}$ and $R$ are the Ricci curvatures and the scalar curvature of $\mathcal{M}$, respectively.

Proof. By Example 4.6 of [Janson 2015], if the metric of $\mathcal{M}$ is defined by the squared infinitesimal distance $|ds|^2$ in $\mathcal{M}$ and a $C^2$-class function $w$ on an open interval $J \subset \mathbb{R}$ as

$$|ds|^2 = \beta^2|dr|^2 + w(r)^2|ds|^2, \quad (19)$$

the Ricci curvature tensor becomes

$$\tilde{R}_{\alpha\gamma} = R_{\alpha\gamma} - \left(\begin{array}{c} (n - 1) \frac{w'}{w} \\ \frac{w''}{w} \end{array}\right) \tilde{g}_{\alpha\gamma},$$

$$\tilde{R}_{\alpha0} = \tilde{R}_{0\alpha} = \tilde{R}_{00} = 0, \quad \tilde{R}_{00} = -(n - 1) \frac{w''}{w} \quad (20)$$

and the scalar curvature becomes

$$\tilde{R} = w^{-2}(R - n(n - 1)(w')^2 - 2nw''w). \quad (22)$$

Since the metric of a metric cone $\tilde{\mathcal{M}}$ is given by

$$|d\tilde{s}|^2 = \beta^2|dr|^2 + \pi^2r^2|ds|^2, \quad (23)$$

by setting $\tilde{r} := \beta r$ and $w(\tilde{r}) := \pi \beta^{-1}\tilde{r}$, we obtain the following form similar to (19):

$$|d\tilde{s}|^2 = |d\tilde{r}|^2 + w(\tilde{r})^2|d\tilde{s}|^2. \quad (24)$$

By substituting $w(\tilde{r}) = \pi \beta^{-1}\tilde{r} = \pi \tilde{r}$, $w'(\tilde{r}) = \pi \beta^{-1}$ and $w''(\tilde{r}) = 0$, we obtain the Ricci and scalar curvatures in Claim 2. \qed

C IDENTIFIABILITY OF THE HEIGHTS IN THE CONE EMBEDDING

In this section, we will prove Theorem 1 of the main article. Let us begin by rewriting Theorem 1 as a longer but more theoretically rigorous form.

Theorem 3 (A rigorous restatement of Theorem 1) Let $Z$ be a length metric space and $X$ be a metric cone of $Z$ with a parameter $\beta > 0$. Let $n$ be an integer at least 3. Fix $z_i \in Z$ and $x_i := (z_i, t_i) \in X$ with $t_i \in [0, 1]$ for $i = 1, \ldots, n$. Denote a matrix $D := [d_\beta(x_i, x_j)]_{i,j=1}^n.$

(a) Assume $z_1, \ldots, z_n$ are not all aligned in a geodesic.

Given $z_1, \ldots, z_n$ and $D$, the number of possible values of $(t_1, \ldots, t_n)$ is at most four.
(b) Let $n \geq 4$ and assume $z_1, \ldots, z_n$ and $t_1, \ldots, t_n$ are in a "general" position. Here "general" position means that, besides the assumption in (a), given any distinct 4 points $z_i, z_j, z_k, z_l \in Z$ and corresponding heights $t_i, t_j, t_k \in [0, 1]$, but still $t_l$ can take infinitely many values. Then $t_1, \ldots, t_n$ are determined uniquely by $z_1, \ldots, z_n$ and $D$.

(c) If $d(z_i, z_j) \geq \beta/2$ for all $i, j = 1, \ldots, n, i \neq j$, then $t_1, \ldots, t_n$ are determined uniquely by $z_1, \ldots, z_n$ and $D$.

Before the proof, we will state some remarks.

If $n = 2$, the identifiability problem reduces to an elementary geometric question: given a circle sector as the right two subfigures of Figure 1 of the main paper and the length of the blue line segment(s) connecting $(x, s)$ and $(y, t)$, can $s$ and $t$ be determined uniquely? The answer is evidently no. But it is notable that there are two types of counterexamples. The first type is as Figure 6(A), one point moves "up" and the other moves "down". But it is maybe counter-intuitive: both moves "up" or "down". Note that the second case does not happen if the angle $\theta$ is larger than or equal to $\pi/2$.

![Figure 6: Two types of movement for a line segment of constant length](image)

If $n = 3$, the picture becomes a tetrahedron as in Figure 7. Here the angles and edge lengths are defined by

$$
\theta_1 := \pi \min(d_Z(z_2, z_3)/\beta, 1), \quad a_1 := \tilde{d}_\beta(x_2, x_3)
$$
$$
\theta_2 := \pi \min(d_Z(z_3, z_1)/\beta, 1), \quad a_2 := \tilde{d}_\beta(x_3, x_1)
$$
$$
\theta_3 := \pi \min(d_Z(z_1, z_2)/\beta, 1), \quad a_3 := \tilde{d}_\beta(x_1, x_2)
$$

and $\theta_1 + \theta_2 + \theta_3$ is assumed to be at most $2\pi$. Then the geometrical question becomes "when angles $\alpha, \beta, \gamma$ and edge lengths $a_1, a_2, a_3$ of triangle $\triangle x_1 x_2 x_3$ is given, can the position of the points $x_1, x_2$ and $x_3$ be determined uniquely?" If it is not unique and there are two different positions of $x_1, x_2$ and $x_3$, at least one edge should move as in Figure 6(B) since it is impossible to move all the three edges as in Figure 6(A). But if all of the angles are larger than or equal to $\pi/2$, this cannot happen. This gives actually a geometrical proof of Theorem 3(c).

Figure 7: Metric cone generated by three points $z_1, z_2, z_3 \in Z$

If $\theta_1 + \theta_2 + \theta_3$ is larger than $2\pi$, the geometric arguments become complicated. We do not need this kind of case analysis when we use algebraic arguments as in the following proof.

Now we will prove the theorem. In the proof, we use the Gröbner basis as a tool of computational algebra. See for example [Cox et al., 2013] about definition and application of the Gröbner basis.

**Proof.** (a) Since the maximum number of possible values of $(t_1, \ldots, t_n)$ does not increase with $n$, it is enough to prove for $n = 3$. We set $\theta_1, \theta_2, \theta_3 \in [0, \pi]$ and $a_1, a_2, a_3 \geq 0$ as (25). Then by the law of cosine,

$$
t_1^2 + t_2^2 - 2t_1t_2\cos\theta_1 = a_1^2,
$$
$$
t_2^2 + t_3^2 - 2t_2t_3\cos\theta_2 = a_2^2,
$$
$$
t_1^2 + t_3^2 - 2t_1t_3\cos\theta_3 = a_3^2.
$$

We consider this as a system of polynomial equations with variables $t_1, t_2, t_3$ and compute the Gröbner basis of the ideal generated by the corresponding polynomials by degree lexicographic monomial order (deglex) with $t_1 > t_2 > t_3$ by Mathematica. Then the output becomes as in Note 6 and the basis includes $-t_1^2 + (2\cos\theta_2)t_1t_3 - t_3^2 + a_2^2$, $-t_2^2 + (2\cos\theta_3)t_2t_3 - t_3^2 + a_3^2$ and $4v(\theta_1, \theta_2, \theta_3)t_3^3 + \text{terms of degree} \leq 2$ where

$$
v(\theta_1, \theta_2, \theta_3) := 1 + 2\cos\theta_1\cos\theta_2\cos\theta_3 - \cos_1^2 - \cos_2^2 - \cos_3^2.
$$

(27)

Note that when $\theta_1 + \theta_2 + \theta_3 \leq 2\pi$, $a_1, a_2, a_3$ of triangle $\triangle x_1 x_2 x_3$ is, therefore, it has a positive value unless the tetrahedron degenerates. By the assumption, $z_1, z_2, z_3$ are not aligned in a geodesic and therefore the tetrahedron does not degenerate and $v(\theta_1, \theta_2, \theta_3)$ must be nonzero. Note that this becomes negative when $\theta_1 + \theta_2 + \theta_3 > 2\pi$.

On the other hand, it is known that the system of polynomial equations with a Gröbner basis $G$ has a finite number of
(complex) solutions if and only if, for each variable \( x \), \( G \) contains a polynomial with a leading monomial that is a power of \( x \). Now all variables \( t_1, t_2 \) and \( t_3 \) satisfy such property, thus we conclude there are at most a finite number of solutions.

Then by Bézout’s theorem, the number of solutions is at most the product of the degree of three polynomial equations, i.e. \( 2 \times 2 \times 2 = 8 \). But if \((t_1, t_2, t_3)\) is a solution, \((-t_1, -t_2, -t_3)\) is also a solution, and only one of each pair can satisfy \(t_1, t_2, t_3 \leq 0\). Thus we conclude the number of possible values of \((t_1, t_2, t_3)\) is at most four.

(b) By the assumptions in (a), without loss of generality, we can assume \(z_1, z_2, z_3\) are not aligned in a geodesic. By the result of (a), given \(z_1, z_2, z_3\) and distances \(d_\beta(x_1, x_2), d_\beta(x_1, x_3), d_\beta(x_2, x_3)\), there are at most four variations of the values of \((t_1, t_2, t_3)\). Here we assume \(t_1\) can take multiple values including \(t_1\) and \(\bar{t}_1\).

Suppose, in addition to the above, the values of \(z_4\) and \(d_\beta(x_1, x_4) (= a_4)\) are given and let \(\theta_4 := \pi \min(d_Z(z_1, z_4)/\beta, 1)\). Then both \(t_1\) and \(\bar{t}_1\) satisfy \(t_1^4 + \bar{t}_1^4 - 2t_1\bar{t}_1\cos \theta_4 = a_4^2\) and therefore \(2t_1\cos \theta_4 = t_1 + \bar{t}_1\) must hold. Since \(t_1\) and \(\bar{t}_1\) are different non-negative values, \(t_1 + \bar{t}_1 > 0\) and, therefore, \(\cos \theta_4 \neq 0\). Hence we obtain \(t_4 = (t_1 + \bar{t}_1)/2\cos \theta\).

This means if \(t_4\) takes values except \((t_1 + \bar{t}_1)/2\cos \theta\), at most only one of \(t_1\) and \(\bar{t}_1\) can be a solution. We can reduce each pairwise ambiguity of the (at most) 4 possibilities of \((t_1, t_2, t_3)\) one by one similarly. Finally \((t_1, t_2, t_3)\) are determined uniquely for all except at most \(\binom{4}{3} = 6\) values of \(t_4\). But such finite values of \(t_4\) can be neglected thanks to the assumption of “general” position in the theorem. Since the same argument holds for any triplets, the statement has been proved.

(c) If \((t_1, \ldots, t_n)\) can take multiple values, without loss of generality we can assume \((t_1, t_2, t_3)\) takes multiple values. By the assumption in the theorem, \(\theta_1, \theta_2, \theta_3 \geq \pi/2\) and therefore all coefficients in each equation of (26) become positive. Thus, if \(t_i\) increases/decreases then \(t_j\) must decrease/increase for \((i, j) = (1, 2), (2, 3), (3, 1)\) but this cannot happen simultaneously. Hence \((t_1, t_2, t_3)\) cannot take multiple values.

Note that all of this proof works even when \(\theta_1 + \theta_2 + \theta_3\) is larger than \(2\pi\).

By an elemental geometrical argument, a new position of \(x_1, x_2, x_3\) and \(O'\) gives another solution of \(t_1, t_2, t_3\). Hence obviously there are infinite number of solutions.

Remark 5 The assumption of “general” positions of \(z_1, \ldots, z_n\) in Theorem 3(b) is satisfied easily for most data distributions. For example, if both \(z_1, \ldots, z_n \in \mathbb{R}^d\) and \(t_1, \ldots, t_n \in [0, 1]\) are i.i.d. from a probability distribution whose density function exists with respect to the Lebesgue measure, then it is easy to see the assumption holds almost surely and therefore uniqueness of the solution is guaranteed. Note that for \(n = 3\) under the same setting, there can be multiple solutions with a positive probability.

Note 6 Computation of the Gröbner basis by Mathematica:

For simplicity, we put \(x := t_1, y := t_2, z := t_3, a := 2\cos \theta_1, b := 2\cos \theta_2, c := 2\cos \theta_3, d := a_1^2, e := a_2^2\) and \(f := a_3^2\).

Note that the second, first and last polynomials in the output correspond to \(-t_1^4 + (2\cos \theta_1)t_1t_3 - t_3^4 + a_2^2, -t_2^4 + (2\cos \theta_2)t_2t_3 - t_3^4 + a_3^2\). Hence \((t_1, t_2, t_3)\) cannot take multiple values.

Remark 4 Assumption in Theorem 3(a) is necessary. If the assumption fails, the tetrahedron degenerates and \(x_1, x_2, x_3\) and the apex \(O\) are all in a plane. When \(O\) happens to be on a circle passing through \(x_1, x_2\) and \(x_3\), move \(O\) to another point \(O'\) on the same circle. Then the angles corresponding to \(\theta_1, \theta_2, \theta_3\) do not change by the inscribed angle theorem.
\textbf{In := GroebnerBasis}\{x^2 + y^2 - a*x*y - d, x^2 + z^2 - b*x*z - e, y^2 + z^2 - c*y*z - f\}, \{x, y, z\},
\textbf{MonomialOrder -> DegreeLexicographic}\}

\textbf{Out = \{f - y^2 + c y z - z^2, e - x^2 + b x z - z^2, d - x^2 + a x y - y^2, d x - e x + a e y - x y^2 - b d z + b y^2 z + x z^2 - a y z^2, -c d x + c e x - a c e y + b f y + c x y^2 - b y^3 + b c d z - a f z + a y^2 z - c x z^2 - b y z^2 + a z^3, a f x + d y - f y - x^2 y - c d z + c x^2 z - a x z^2 + y z^2, b f x - c e y + c x^2 y - b x y^2 + e z - f z - x^2 z + y^2 z, -c e x + a b f x + c x^3 + b d y - b f y - b x^2 y - b c d z + a e z - a x^2 z + c x z^2 + b y z^2 - a z^3, a c d x - a c e x + b f x + c d y - c e y + a^2 c e y - a b f y - b x y^2 + a b y^3 - c y^3 - a b c d z + e z - f z + a^2 f z - x^2 z + y^2 z - a^2 y^2 z + a c x z^2 + a b y z^2 - a^2 z^3, -a e f - d x y + e x y + f x y + c d x z - c e x z + b d y z - b f y z - b c d z^2 + a e z^2 + a f z^2 - 2 x y z^2 + c x z^3 + b y z^3 - a z^4, -c d e + c e^2 - a b e f + c d x^2 - c e x^2 - b d x y + b e x y + b f x y - a f x z - d y z + b^2 d y z + 2 e y z + f y z - b^2 f y z - x^2 y z + 2 c d z^2 - b^2 c d z^2 + a b e z^2 - 2 c e z^2 + a b f z^2 + a x z^3 - 3 y z^3 + b^2 y z^3 - a b z^4 + c z^4, -d x + a^2 d x + a b c d x + 2 e x - a^2 e x - a b c e x - a^2 f x + b^2 f x - x^3 + b c d y + a e y + a^3 e y - b c e y + a^2 b c e y + a f y - a b^2 f y + x y^2 - b^2 x y^2 - a y^3 + a b^2 y^3 - b c y^3 + b d z - a^2 b d z + a c d z - a b^2 c d z + b e z - a c e z - b f z + a^2 b f z - 2 x z^2 + 2 a^2 x z^2 - a^3 y z^2 + a b^2 y z^2 - a^2 b z^3 + a c z^3, -c d^2 + c d e + a b d f + c d x^2 - c e x^2 + b f x y - a b d y z - 2 c e y^2 + a^2 c e y^2 - a b f y^2 - b x y^3 + a b y^4 - c y^4 - a d x z + a e x z - a f x z - 2 d y z + e y z - a^2 e y z - f y z + a^2 f y z + x^2 y z + 3 y^3 z - a^2 y^3 z, d^2 - 2 d e + c^2 d e + e^2 - c^2 e^2 - 2 d f + b^2 d f + 2 e f - a^2 e f + a b c e f + f^2 - b^2 f^2 - c^2 d x^2 + c^2 e x^2 + b c d x y - b c e x y - b c f x y - b^2 d y^2 + b^2 f y^2 + a c d x z - a c e x z + a c f x z + a b d y z - a b e y z - a b f y z + a c d z^2 - 2 b c^2 d z^2 - a b c d z^2 - 2 c^2 d z^2 + b^2 c^2 d z^2 - 4 e z^2 + a^2 e z^2 - a b c e z^2 + 2 c^2 e z^2 - 4 f z^2 + a^2 f z^2 + 2 b^2 f z^2 - a b c f z^2 + 4 z^4 - a^2 z^4 - b^2 z^4 + a b c z^4 - c^2 z^4\}