Korean J. Math. **29** (2021), No. 4, pp. 649–664
http://dx.doi.org/10.11568/kjm.2021.29.4.649

ORGANIC RELATIONSHIP BETWEEN LAWS BASED ON JUDICIAL PRECEDENTS USING TOPOLOGICAL DATA ANALYSIS

Seonghun Kim and Jaeheon Jeong

Abstract. There have been numerous efforts to provide legal information to the general public easily. Most of the existing legal information services are based on keyword-oriented legal ontology. However, this keyword-oriented ontology construction has a sense of disparity from the relationship between the laws used together in actual cases. To solve this problem, it is necessary to study which laws are actually used together in various judicial precedents. However, this is difficult to implement with the existing methods used in computer science or law. In our study, we analyzed this by using topological data analysis, which has recently attracted attention very promisingly in the field of data analysis. In this paper, we applied the the Mapper algorithm, which is one of the topological data analysis techniques, to visualize the relationships that laws form organically in actual precedents.

1. Introduction

In recent decades, conflicts between many individuals or groups have increased exponentially as societies have diversified and lots of changes have occurred in the industrial structure. As a result, the demand for legal services has increased. Experts predict that the size of the legal market will continue to grow due to various reasons such as an increase in the number of lawyers and the expansion of the new legal service market [7]. However, in spite of these high demands, the law comes to non-professionals as a being that is difficult to deal with and esoteric. There have been many efforts at the level of government and individuals to solve this problem. In 2008, the government reclassified intricately entangled laws based on the lives of ordinary citizens and provide through the Practical Law Information Service(https://www.easylaw.go.kr/) so that non-experts can easily find and understand the contents of the esoteric laws. In addition, a number of studies have been conducted to make more convenient legal information search services, such as a research on how to search by matching life terms and legal terms [5]. However, despite these many efforts, most of the existing attempts are limited to forming keyword-oriented legal ontology based on natural language processing (NLP). Providing legal information grouped by keywords is very important

Received March 22, 2021. Revised October 14, 2021. Accepted October 16, 2021.
2010 Mathematics Subject Classification: 68P05.
Key words and phrases: Topological data analysis, the Mapper algorithm, Law, Organic relationship, Act, Article, judicial precedent.

and necessary in constructing legal ontology, but this alone does not establish a highly complete legal knowledge base. Because there are so many associated laws within a single keyword, it is very difficult to pinpoint the necessary laws within them. To make matters worse, laws that are grouped by the same keyword do not mean they are used together in actual cases. Users have to go through the entire list each time they search for each keyword corresponding to a case. To solve this inconvenience, we propose a new perspective of classifying laws by whether or not they actually work together.

In each precedent, laws may be referenced alone or in combination. They also relate to various laws in other precedents. However, this is not such a simple situation. We must also consider to what extent each law is related, because two laws referenced together multiple times have to be considered to be more *strongly* related than two laws referenced together only once. Figure 1 is an example of the relationship between laws.

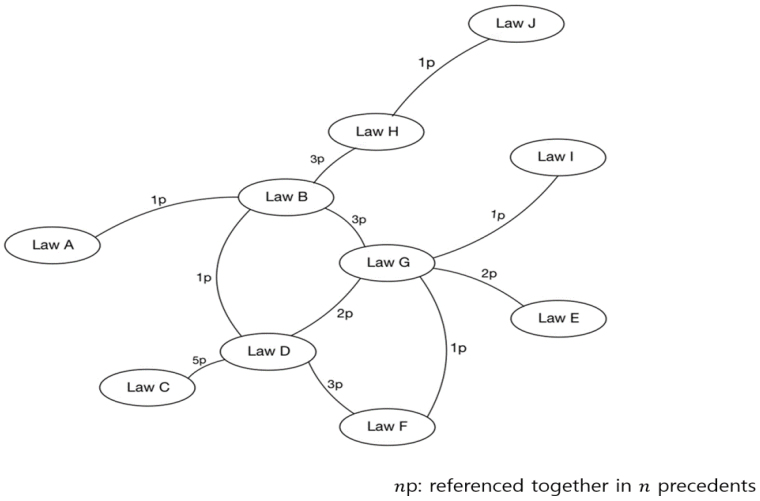

FIGURE 1. Relationship between laws

As such, each law is actually used together in various precedents such as Supreme Court precedents and Constitutional Court precedents to form a network of relationships. Therefore, it is important to understand the relationship organically formed by the law in actual precedents, away from the existing method of analyzing the law in the form of keywords or sentences. In addition, in order to provide easy and convenient legal services to non-professional users, it is necessary to present each law and the laws that are actually used together as related laws. Until now, there have been many studies to understand the structural properties of laws and to improve the legal system, but none of them have focused on which laws are used together actually. However, it is not easy to analyze these relationships with the usual methods used in law and computer science.

In this paper, we will use Topological data analysis (TDA), which is attracting great attention among a number of big data analysis techniques. Studies using TDA in Korea include a study that analyzed hotel review data based on emotional keywords [3], a study that analyzed mathematical anxiety [6], and a study that analyzed VLBI time series data [8]. Unlike other methods, topological data analysis has the

advantage of being able to simultaneously perform clustering and connectivity analysis. Our study of analyzing the real organic relationships of law is a very fresh and novel approach and will contribute to building a more advanced legal ontology by organizing legal knowledge more efficiently.

2. Topological data analysis

Topology is a field of mathematics that studies shapes and properties that do not change due to continuous deformations such as Hausdorff and compactness. In topology, we basically pay more attention to connectivity rather than the distance between two objects that changes easily in scaling, or twisting. As you can see in Figure 1, it is our primary concern to understand how each law relates to other laws in various precedents and what networks they form. We can extract geometric features of data set such as *loops* which are continuous circular segments and *flares* which are long linear segments by applying topology to data analysis. Topological data analysis is a field that has been studied for about two decades, and is emerging as an innovative method that can complement the limitations of existing data analysis methods such as Principal Component Analysis (PCA) and Multidimensional scaling (MDS).

Topological data analysis has several features [9]. First, TDA finds the topological properties of the data set. A topological property means a property preserved under homeomorphism. For example, no matter how much an object is bent or stretched, the number of holes or the connectivity between each component does not change. TDA extracts these properties of the data set. This also means that TDA has resistance to noise. Second, TDA has the advantage of being able to observe the data set at multiple resolutions. we can analyze from the global structure of the data set to the local parts of each data point to various resolution we set. Third, TDA shows complex data in a compressed form such as a simplicial complex or a network. A simplicial complex $\mathcal{K}$ is a finite set of simplices that satisfies these two conditions:

i) If $\sigma_1 \cap \sigma_2 \neq \phi$ for two simplices $\sigma_1, \sigma_2 \in \mathcal{K}$, then they meet along faces.

ii) If a face $\tau \in \sigma_1 \cap \sigma_2$, then $\tau \in \mathcal{K}$.

Here, n-simplex means the convex hull of $n+1$ affinely independent vertices $S = \{v^i\}$, $i \in [0, 1, ..., n]$ in R^d where $d \geq n$ [10]. Finally, TDA makes the output in *coordinate free* way. Each data may have different coordinate values depending on the equipment and measurement technology used. But, TDA always produces similar results that capture the geometric properties of the data regardless of the coordinate system.

There are two methods in TDA: **Persistent homology** and the **Mapper algorithm**. Persistent homology extracts topological properties that appear persistently at different spatial resolutions. It is mainly used to compare with other objects. Meanwhile, the the Mapper algorithm is used to visualize the overall structure of the data as a network in a plane or low dimension [3]. The purpose of this paper is presenting and visualizing the relationship network of laws by using the Mapper algorithm.

2.1. Basic principle. The Mapper algorithm is a method suggested by Singh et al in 2007. In this section, I will introduce the basic principle of the Mapper algorithm with an intuitive example. If you want more theoretical details, see [12]. the Mapper algorithm proceeds through filtering, partitioning, clustering, and visualization. Any specific function or clustering algorithm is not obligatory in the Mapper algorithm.

You can choose the appropriate technique according to the situation and your purpose.

The Mapper algorithm begins with giving a metric to a data set X with $|X| = N$. By giving a metric, the data space becomes a easy-to-handle topological space, and we can manipulate the data through several transformations. Here, for easy understanding, we assume that human-shaped data are given in a two-dimensional Euclidean space as shown in Figure 2-(a). The distance in the data space does not be restricted to Euclidean distance. You can choose the best one corresponding the features of the data set(cosine distance, chebyshev distance, correlation distance, and etc). If the metric should be defined directly due to the characteristic of the data set, even though it does not satisfy the triangle inequality, you can give an N by N square matrix calculating the distances between each data point as an input of the algorithm.

In the filtering step, define a real-valued function f from the data set X to a parameter space Z. The parameter space Z may be the real line $\mathbb{R}$ or the unit circle S^1 in the plane. This function is called the **filter function**. The filter function often depends on the distance function we previously set. It plays a very important role in the Mapper algorithm to reflect the geometric properties of the data and the result of the Mapper can be changed by the filter function. Since it is similar to what lens the camera sees through, filter function is also called a lens(lens is also used as a word indicating projected data in the actual programming process). As a filter function, you can use various functions such as a density estimator that can measure the density of data using Gaussian kernel, eccentricity that measures *Data Depth*, a concept representing how far each data is from the center, etc. You can also apply PCA or MDS at this stage. In our example, we will choose the orthogonal projection onto y-axis, the simplest one(b).

After each data is moved on y-axis by the lens, set a covering U that covers all the filter values of the data. If the parameter space is the real line $\mathbb{R}$, then the covering U will consist of some intervals. You can also set several filter functions at the same time. In this case, the parameter space will be $\mathbb{R}^n$ and the elements of the covering U will be n-dimensional hypercubes. In this step, it is important to ensure that adjacent elements overlap each other to cover the ranges of the filter function. There is no need to keep the overlapping ratio constant, but it is usually unified as one for convenience. Applying the Mapper algorithm, you can set several variables such as resolution or gain as input values. Resolution means the number of elements composing of the covering U, and gain refers to how much each adjacent element overlaps. So, gain is also called percent overlap.

Next, give each element $U_i \Phi (i = 1, 2, ..., n)$ of the covering U a different color, and find each inverse image $f^{-1}(U_i)$ under the filter function f. Since we set adjacent cubes that overlap by a certain amount in the covering U, each $f^{-1}(U_i)$ will also form regions of different colors that overlap adjacently in the data space(c).

Also, since the covering $\{U_i\}$ covers the entire range of f, $\{f^{-1}(U_i)\}$ also covers the entire data set. Thus, for any data point $x \in X$, there exists $i_0 \in \{1, 2, ..., n\}$ such that $x \in f^{-1}(U_{i_0}) \in \{f^{-1}(U_i)\}$. This allows us to partially cluster the data set X into several overlapping groups. You can apply clustering algorithm on the projection in the parameter space Z, but this may give you some projection loss. So, you'd better cluster on the original data set X to avoid it. In this step, You are not forced to use a certain specific clustering algorithm. You can try k-means, hierarchical clustering, spectral clustering and so on. However, there is one caveat here. Even data projected

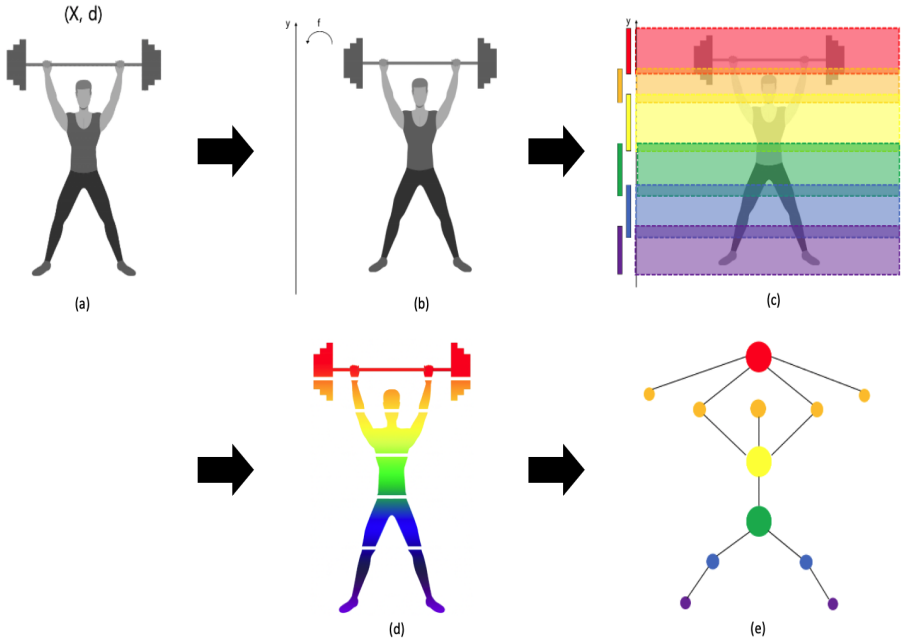

FIGURE 2. The procedure of the Mapper algorithm

into the same interval do not have to belong to the same cluster. When clustering in the data space, you should ensure that the distance between the inner data in each cluster is always less than the distance between the clusters. Under this condition, we can check that the left leg and the right leg can be separated in Figure 2-(d).

In the visualization step, we express each cluster as a node. The size of each node depends on the amount of data included. The more data each node contains, the larger the node, and the less data it contains, the smaller the node. Since we set adjacent clusters overlap each other, some data point can appear in multiple nodes. If two nodes have a non-empty intersection, then connect them with an edge. We can represent the data set in the form of a simple network in this way(e). At this time, the filter value of the data constituting the node can be roughly known through the color(a red node contains data with high filter values, and a blue node contains data with low values). Although the Mapper's output is compressed and has a much simpler form compared to the complex data prototype, it provides relatively accurate information about the distribution of data through the size and color of each node.

Through these series of procedures, the Mapper algorithm represents the relationship of high-dimensional complex data in the form of a network in a two-dimensional plane or a multidimensional simplicial complex. In the example above, we can confirm that the Mapper grasps the partial characteristics corresponding to each body part or weight, as well as the connectivity between each cluster. Topological data analysis extracts key topological properties such as loops and flares in the data space unlike existing methods where it is difficult to study the structural aspects or geometric properties of the data.

3. The application of the Mapper algorithm

Although TDA plays an active role in various fields around the world and have many possibilities, there are not many papers applying this method in Korea yet. Moreover, even that is difficult to check the code actually executed. It makes it hard for TDA novices who understand the theory to get to practical application. Thus, we want to help those who want to study this field by showing the actual application process of the Mapper algorithm. Until a few years ago, many programs had to be used in complex ways to apply this algorithm. But now, we can easily implement the Mapper through **kepler Mapper** that is a Python library adopting the scikit-learn API as much as possible. In this paper, we visualized the network of laws by applying the Mapper algorithm to a number of judicial precedents and analyzing the distance relationship between the laws referenced in each precedent.

3.1. Data collecting and preprocessing. From 2019, information on all court precedents across the country can be searched using the "Internet integrated browsing and search service for judgments". However, this information cannot be viewed in a lump, but only judgments corresponding to our search terms. Besides, not only is it too costly to browse them, they are not data in a form that is easy for us to handle. Korea's major precedent data can be obtained at the National Law Information Center(https://www.law.go.kr/). You can browse about 82,000 cases here, but the number of data that can be downloaded in Excel format is limited. We conducted the study with 10,677 data.

We used the `openpyxl` Python library in the data preprocessing. In the downloaded list, there are data including empty blanks due to system errors. We first removed all these corrupted data. After that, when a single article is classified in more detail, we treated them as the same (e.g. Article 360-24 of the Commercial Law → Article 360 of the Commercial Law). In our experiment, subdividing Articles further is meaningless. Lastly, we re-stated the Act in each Article when several Articles are listed in one Act (e.g. Civil Code — Article 416, Article 419, Article 421, Article 423 → Article 416 of the Civil Law, Article 419 of the Civil Law, Article 421 of the Civil Law, Article 423 of the Civil Law).

3.2. Metric. The Mapper is an algorithm for simplifying and visualizing the structure of high-dimensional data in low-dimension. You can give a metric using distance functions already built in the kepler Mapper library. Figure 4 is a list of distance functions built into the library. When using the Mapper algorithm, you should give the appropriate metric for the situation because the results can be completely different depending on which distance function is used. In our study, we have considered two laws that are referenced together in the same precedent as being related. When two laws are referenced together multiple times, they become increasingly exponentially associated in proportion to the number of times they are referenced together. We can express the association between two laws as a distance. Hence, the distance $d_{A,B}$ between two laws A and B can be calculated as follows:

$$d = \frac{d_0}{2^n}$$

where d_0 is the distance between two unrelated law and n is the number of times A and B are referenced together. Because this metric cannot be obtained with only

		공익사업을 위한 토지 등의 취득 및 보상에 관한 법률	제70조
		공익사업을 위한 토지 등의 취득 및 보상에 관한 법률 시행규칙	제26조
937	손실보상금청구	사도법	제2조, 제4조
		공익사업을 위한 토지 등의 취득 및 보상에 관한 법률	제77조
938	수용보상금증액	공익사업을 위한 토지 등의 취득 및 보상에 관한 법률 시행규칙	제46조
939	시정명령등취소청구	독점규제 및 공정거래에 관한 법률	제19조
940	시정명령등취소청구의소	전자상거래 등에서의 소비자보호에 관한 법률	제21조, 제32조
941	시정명령및과징금납부명령취소청구의소	독점규제 및 공정거래에 관한 법률	제19조
942	시정조치등취소청구의소		제19조
		국가유공자 등 예우 및 지원에 관한 법률 시행규칙	제8조의3, 제8조의4
943	장해등급미달처분취소	국가유공자 등 예우 및 지원에 관한 법률 시행령	제14조
		공익사업을 위한 토지 등의 취득 및 보상에 관한 법률 시행규칙	제24조
		국토의 계획 및 이용에 관한 법률 시행령	제51조
		농지법	제2조
		농지법 시행령	제2조
944	토지수용보상금	산림법	제8조, 제11조, 제90조
945	감사업무제한등처분취소	자본시장과 금융투자업에 관한 법률	제125조, 제429조, 제430조
		영유아보육법	제34조, 제34조의2, 제34조의3, 제35조, 제36조, 제40조, 제40조의2, 제45조, 제46조, 제54조
946	공공형어린이집선정취소처분취소	영유아보육법 시행규칙	제35조의3
947	과징금부과처분취소	약사법	제76조

		공익사업을 위한 토지 등의 취득 및 보상에 관한 법률 제70조 ,
		공익사업을 위한 토지 등의 취득 및 보상에 관한 법률 시행규칙 제26조 ,
937	손실보상금청구	사도법 제2조 ,사도법 제4조 ,
		공익사업을 위한 토지 등의 취득 및 보상에 관한 법률 제77조 ,
938	수용보상금증액	공익사업을 위한 토지 등의 취득 및 보상에 관한 법률 시행규칙 제46조 ,
939	시정명령등취소청구	독점규제 및 공정거래에 관한 법률 제19조 ,
940	시정명령등취소청구의소	전자상거래 등에서의 소비자보호에 관한 법률 제21조 ,전자상거래 등에서의 소비자보호에 관한 법률 제32조 ,
941	시정명령및과징금납부명령취소청구의소	독점규제 및 공정거래에 관한 법률 제19조 ,
		국가유공자 등 예우 및 지원에 관한 법률 시행규칙 제8조 ,국가유공자 등 예우 및 지원에 관한 법률 시행규칙 제8조 ,
943	장해등급미달처분취소	국가유공자 등 예우 및 지원에 관한 법률 시행령 제14조 ,
		공익사업을 위한 토지 등의 취득 및 보상에 관한 법률 시행규칙 제24조 ,
		국토의 계획 및 이용에 관한 법률 시행령 제51조 ,
		농지법 제2조 ,
		농지법 시행령 제2조 ,
944	토지수용보상금	산림법 제8조 ,산림법 제11조 ,산림법 제90조 ,
945	감사업무제한등처분취소	자본시장과 금융투자업에 관한 법률 제125조 ,자본시장과 금융투자업에 관한 법률 제429조 ,자본시장과 금융투자업에 관한 법률 제430조 ,
		영유아보육법 제34조 ,영유아보육법 제34조 ,영유아보육법 제34조 ,영유아보육법 제35조 ,영유아보육법 제36조 ,영유아보육법 제40조 ,영유아보육법 제40조 ,영유아보육
946	공공형어린이집선정취소처분취소	영유아보육법 시행규칙 제35조 ,
947	과징금부과처분취소	약사법 제76조 ,

FIGURE 3. Before and after data preprocessing

```
["braycurtis", "canberra", "chebyshev", "cityblock", "correlation", "cosine", "dice",
 "euclidean", "hamming", "jaccard", "kulsinski", "mahalanobis", "matching", "minkowski",
 "rogerstanimoto", "russellrao", "seuclidean", "sokalmichener", "sokalsneath",
 "sqeuclidean", "yule"].
```

FIGURE 4. The list of built-in distance functions

the built-in functions of the kepler Mapper library, we have to create our custom metric. However, how to do this is not well explained, and we couldn't find cases of TDA that made custom metrics. Basically, the `project` function of the kepler Mapper library performs projection in a way that returns the $N \times N$ distance matrix after automatically calculating the distances between each coordinated data using the metric already built in the library. However, since our legal data has a characteristic that is difficult to coordinate, the above process could not be applied as it is. Therefore, we modified the code of kmapper.py and used our custom metric directly for projection without going through the above process. We presented the code of process creating our custom metric in the appendix. You can check the entire process of our the Mapper algorithm on [4].

3.3. Filtering. The filter function is a function that determines which geometric characteristics of the data set are to be extracted and is a very important factor that influences the result of the Mapper. In our study, we used t-Distributed Stochastic Neighbor Embedding (t-SNE) as a filter function. Stochastic Neighbor Embedding (SNE), developed by Hinton and Roweis in 2002 [2], is a technique that preserves distance information between high-dimensional data and embeds it into a low-dimensional space. In SNE, the similarity of the data point x_j to x_i is expressed as a conditional probability $p_{j|i}$ using the distance between the two points. $p_{j|i}$ is proportional to the probability density values of x_j in a Gaussian distribution whose

center is x_i. We can compute $p_{j|i}$ as following:

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

The similarity between the two data points y_j and y_i in the lower dimension corresponding to x_j and x_i can be expressed in similar. Here, the variance σ_i is set to $\frac{1}{\sqrt{2}}$ and $q_{j|i}$ is calculated as follows;

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

In order to minimize the error between each $p_{j|i}$ and $q_{j|i}$, the gradient descent method is used, where the cost function is as follows;

$$\text{cost function} = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

However, there are some shortcomings to this SNE technique, and to compensate for this, Maaten and Hinton proposed t-SNE in 2008 [13]. In t-SNE, the cost function is changed to a symmetrized version to solve the problem of SNE, which was difficult to optimize. For this, p_{ji} and q_{ji} are used instead of $p_{j|i}$ and $q_{j|i}$, and p_{ji} is computed as $p_{ji} = \frac{(p_{j|i}+p_{i|j})}{2N}$. In addition, Student t-distribution which has much more heavier tails than Gaussian is used as a probability distribution in the low-dimension in order to solve the crowding problem that occurs when mapping high-dimensional data into a low-dimensional space. Accordingly, q_{ij} is expressed as follows;

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k\neq l}(1 + \|y_k - y_l\|^2)^{-1}}$$

In addition, the performance of the algorithm is further improved through some tricks such as "early compression" that adds $L2$-penalty to the cost function and "early exaggeration" that multiplies each conditional probability p_{ij} by a specific value appropriately to each situation at the beginning of optimization.

The Mapper algorithm creates a metric in the form of a square matrix by automatically calculating the distance between each data when we put the coordinate values of data and a distance function as an input. However, since we did not follow this usual procedure and created our custom metric directly, we cannot use filter functions that require the coordinate values of the data. By applying this t-SNE technique, we embedded high-dimensional law data into low-dimension while minimizing information loss.

3.4. Partitioning. In this step, we need to adjust two hyperparameters: resolution and gain. Resolution, denoted by `n_cubes`, means the number of hypercubes in the parameter space. If we set `n_cubes` high, the parameter space will be divided into more areas and more clusters will be formed in the original data set (to be precise, more colored clusters are created). Then single pieces of data form clusters and we can observe the locality of the data. Conversely, if we set `n_cubes` small, the original data is grouped into a small amount of clusters. Thus, we can see the global structure of the data set. Figure 5 shows the results that change with the increase of `n_cubes`.

On the other hand, gain, denoted by `perc_overlap`, means the degree of overlap between each hypercube. If we set `perc_overlap` high, the probability that each data

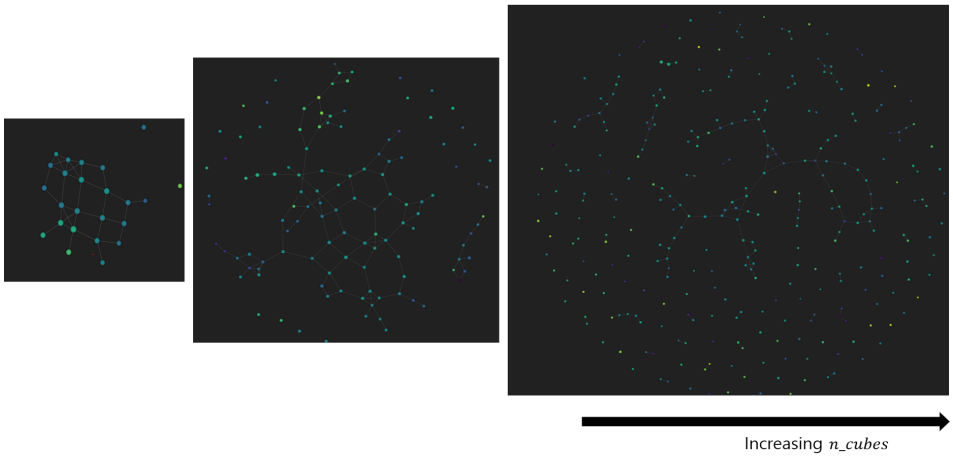

FIGURE 5. Results in various `n_cubes`

belongs to multiple hypercubes at the same time increases. Therefore, the connectivity between each cluster becomes complex and the number of edges increases. Conversely, if we set `perc_overlap` low, the probability that the intersection of each adjacent hypercube is empty is high. Then a simple network is formed between each cluster in the original data set. Figure 6 is the results in various `perc_overlap`.

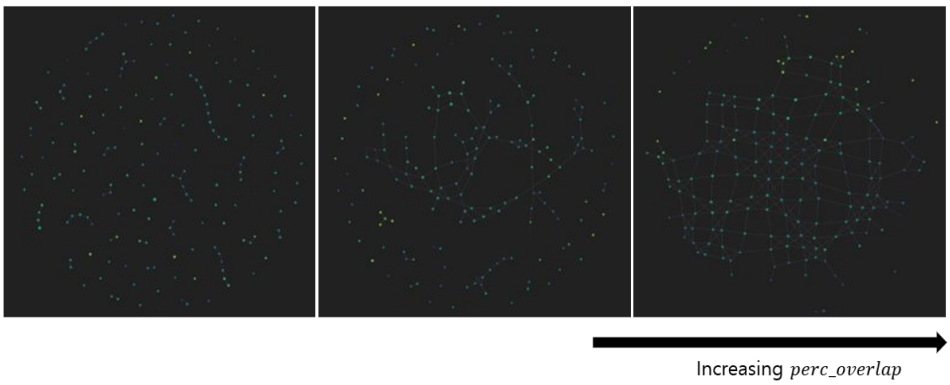

FIGURE 6. Results in various `perc_overlap`

3.5. Clustering and visualization. The Mapper algorithm does not restrict the use of any specific clustering algorithm. In our study, we used the Density Based Spatial Clustering of Applications with Noise (DBSCAN) technique, which is one of the most commonly used clustering algorithms. DBSCAN is a data clustering technique on density-based notion proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996. There are some preliminaries before exploring the principle of this algorithm. DBSCAN requires some hyperparameters as an input such as epsilon(ϵ) and minimum samples(m).

DEFINITION 3.1. A point a is a **core sample** if $\exists$ m other samples in $N_\epsilon(a)$. These other samples are **neighbors** of the core sample.

Here, $N_\epsilon(a)$ is a closed ball of points z such that $\{z \mid \mathrm{dist}(z, a) \leq \epsilon\}$

DEFINITION 3.2. A point a is **directly density-reachable** from a point b with respect to ϵ, m if $a \in N_\epsilon(b)$ and $|N_\epsilon(b)| \geq m$.

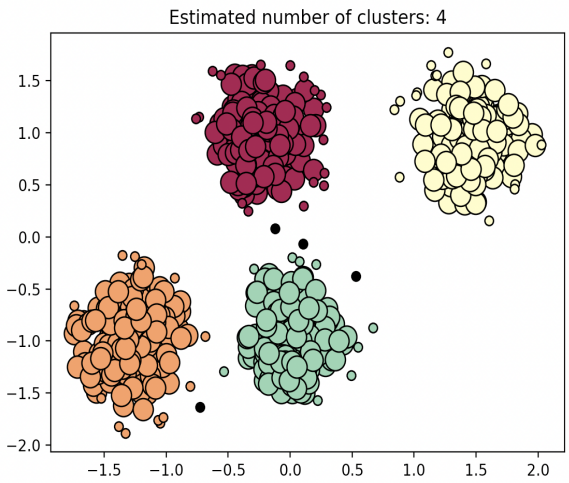

FIGURE 7. Clusters by DBSCAN

DEFINITION 3.3. A point a is **density-reachable** from a point b with respect to ϵ and m if $\exists$ points $p_1, ..., p_n$ such that $p_1 = b$, $p_n = a$, and p_{i+1} is directly density-reachable from p_i.

DEFINITION 3.4. A point a is **density-connected** to a point b with respect to ϵ and m if $\exists$ a point o such that both a and b are density-reachable from o with respect to ϵ and m.

DEFINITION 3.5. A **cluster** C with respect to ϵ and m is a non-empty subset of a database if

i) $\forall a \in C$, $\forall b$ density-reachable from a with respect to ϵ and m, $b \in C$

ii) $\forall a, b \in C$, a is density-connected to b with respect to ϵ and m.

It is trivial that density-connectivity is a symmetric relation. For each core sample, it is density-connected to itself, which means density-connectivity is reflexive. Suppose that a is density-connected to b, and b is density-connected to c for core samples a, b, and c. Then, there exists a core sample o_1 such that a and b are density-reachable from o_1, and there exists a core sample o_2 such that b and c are density-reachable from o_2 with respect to ϵ and m. Note that density-reachability is symmetric for core samples. Hence, both a and c are density-reachable from the core sample b. Thus, a is density-connected to c by definition. Therefore, density-connectivity is transitive.

In the procedure of forming clusters, DBSCAN begins with an arbitrary point a and collects the points density-reachable from a with respect to ϵ and m. The success of cluster formation depends on whether a is a core sample or not. If a is a core sample, DBSCAN will make a cluster containing a. If not, any cluster cannot be generated because no point is density-reachable from a. Next, DBSCAN continues the above process. Since density-connectivity is an equivalence relation for core samples, the members of core samples in each cluster do not change according to the data order while non-core samples can be assigned to the different clusters due to the order. DBSCAN is a deterministic algorithm in that it always creates the same clusters if data is given in the same order.

See figure 7. There are large and small circles in each cluster. The large circles represent core samples while the small circles represent neighbors of core samples.

The black circles represent noise data. The **noise** is the set of points

$$\{p \mid \forall i, p \text{ does not belong to any cluster } C_i\}.$$

DBSCAN can find out clusters of any shape unlike algorithms such as k-means, and is efficient for large spatial databases. DBSCAN is an appropriate algorithm that can well catch partial characteristics of the dataset for our purpose. Refer to [1], [11] for more information on DBSCAN.

4. Results

There are three important hyperparameters that influence the outcome of our experiment(`n_cubes`, `perc_overlap`, and ϵ). We have been able to find values that reflect meaningful results through numerous trials. This experiment shows completely different aspects of results around when $\epsilon = 1$. Even data belonging to the same cube in the parameter space can be grouped into different clusters according to the characteristics of the data. The assumption in our experiment was that the default value of the distance between each law was 1. Therefore, if we set the value of ϵ to 1 or more, it means that all data belonging to the same cube will be grouped into the same cluster. Therefore, it is desirable to set the value of ϵ less than 1. See Figure 8. Overall, most clusters exist independently without being related to each other. Observing the data constituting each cluster, you can find that the members of each cluster are all articles within the same Act. From this, it is confirmed that the Mapper algorithm performed its part well. However, unlike most clusters that exist in the form of singletons, there are several clusters that are connected to each other and form a group. Observing these groups, the data in the same group were all articles of the same Act even if they belonged to different clusters. For example, in Group A, Article 37 of the Criminal Procedure Act and Article 151 of the Criminal Procedure Act belonged to different clusters. This means that the articles of the same Act can have different degrees of association. Meanwhile, even within the same Act, there were clusters that formed different groups. In the figure 8, articles of the Civil Code form different groups (Group B and Group C). We can see from this that articles of the same Act can be related to each other in different groups.

We set $\epsilon = 1$ to also check the association between articles of different Acts. Then, we can find that the articles of different Acts can be included in one cluster. In Figure 9, Article 17 of the Public Officials Ethics Act and Article 2 of the Banking Act are actually somewhat related. Article 17 of the Public Officials Ethics Act is the law on the restriction of employment of retired public officials, and Article 2 of the Banking Act defines what is banking. However, this result is not entirely reliable. As mentioned before, this is because all data in the same cube are grouped into one cluster if the value of ϵ becomes 1. In fact, Article 49 of the Urban Parks and Green Spaces Act is also included in this cluster, which is not related to the previous two laws.

At the beginning of this experiment, we expected to discover relationships not only between articles of the same Act, but also between articles of different Acts. In reality, there are many articles of different Acts that are interrelated, but we were not able to check that part in this experiment. This is because we could obtain only a very small amount of precedent data. As a result, we could only identify relationships between

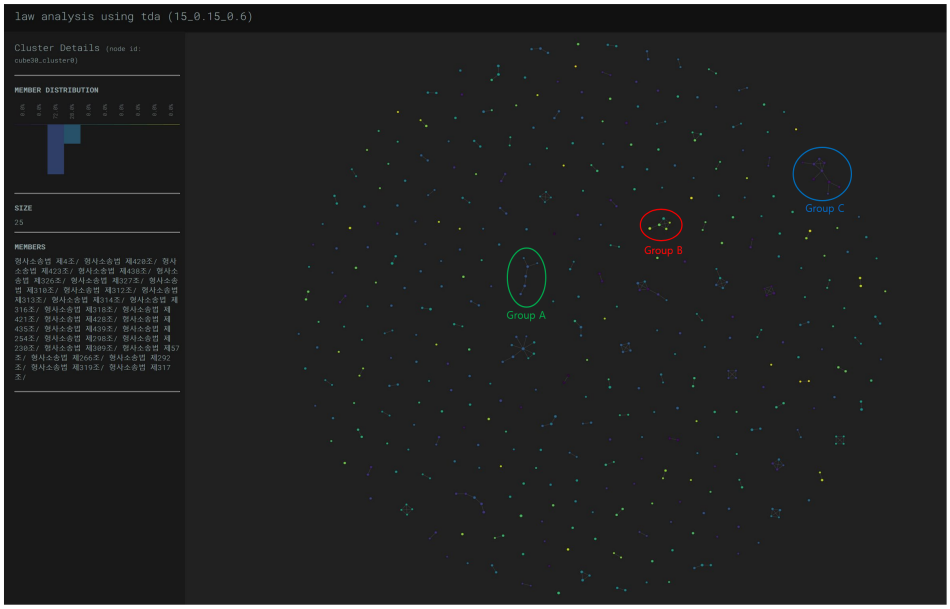

FIGURE 8. The case when `n_cubes`=15, `perc_overlap`= 0.15, ϵ=0.6

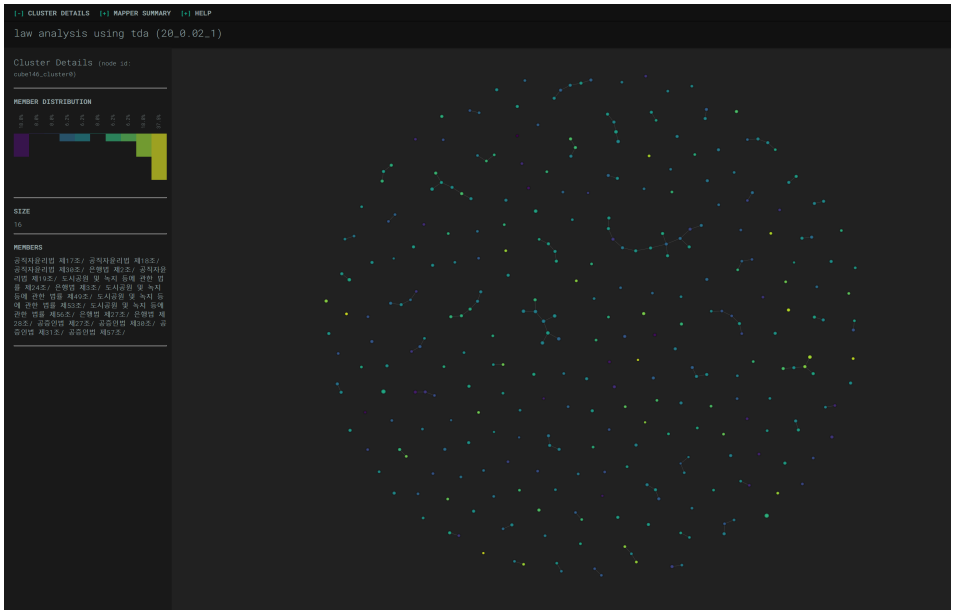

FIGURE 9. The case when `n_cubes`=20, `perc_overlap`=0.02, ϵ=1

laws that were *very strongly* related. If more precedent data are provided to the public in the future, this problem will be solved.

5. Conclusion

In this paper, we investigated the relationships that laws form organically in actual precedents. In the legal ontology that has been analyzed only by keyword until now, this approach will help to compose very diverse content. There is not much precedent data that we can secure yet, but if more and more data can be obtained later, then

we can discover a very complete and systematic relationship network of laws by using the code executed in this study. In future work, we will develop an application that provides associated laws in order of higher relevance(closer distance), and will study ways to provide easy and convenient legal information to the public.

Appendix

Lastly, we will present the process of creating a custom metric. You can check all the processes we performed the Mapper algorithm and the results directly in [4].

```
import operator
import numpy as np
from openpyxl import load_workbook

load_wb = load_workbook("./law_data/law_data.xlsx")
load_ws = load_wb['sheet1']

values = []
output = []
law_dict = {}
num_row = 17143 # number of rows of excel file

i=0
for row in load_ws.rows:
    i +=1
    if i == num_row :
        break

    row_value = []
    for cell in row:
        if cell.value != None :
            row_value.append(cell.value)
    values.append(row_value)

for j in range(len(values)) :
    if str(type(values[j][0])) == "<class 'int'>" :
        del values[j][0]
        output.append(values[j])
    else :
        output[-1].extend(values[j])

# removing error data (blank etc..)
idx = 0
while True :
    if idx == len(output) :
        break
    if len(output[idx]) % 2 == 1 :
        del output[idx]
        idx-=1
    idx+=1

```

```
# preprocessing data
for i in range(len(output)) :
    t_len = int(len(output[i])/2)

    for j in range(t_len) :
        pre = str(output[i][2*j])
        post = str(output[i][2*j+1]).split(', ')
        for k in range(len(post)) :
            if post[k][-2] == "의" :
                post[k] = post[k][0:-2]

            final = pre+' '+post[k]
            if final in law_dict :
                law_dict[final] += 1 # count+1
            else :
                law_dict[final] = 1 # new

law_list = list(law_dict.keys()) # list of whole laws used in all cases
law_num = len(law_list) # number of whole laws used in all cases


# make a numpy array for matching law and index
np_law_list=[]
for i in range(len(law_list)) :
    np_law_list.append(law_list[i] + "/")
np_law_list = np.array(np_law_list)
np.save('law_data/law_list',np_law_list)

distance_matrix = []

# initiate distance matrix uniformly (Set the distance between all laws to 1.)
for i in range(law_num) :
    tmp = []
    for j in range(law_num) :
        if i==j :
            tmp.append(0.0)
        else :
            tmp.append(1.0)
    distance_matrix.append(tmp)

distance_matrix = np.array(distance_matrix) # initiated distance matrix

# update distance matrix
for i in range(len(output)) :
    t_len = int(len(output[i])/2)

    for j in range(t_len) :
        case_law = [] # list for saving laws used in same case
```

```
90
91          pre = str(output[i][2*j])
92          post = str(output[i][2*j+1]).split(', ')
93          for k in range(len(post)) :
94              if post[k][-2] == "의" :
95                  post[k] = post[k][0:-2]
96              final = pre+' '+post[k]
97              case_law.append(final)
98
99          if len(case_law)==1 :
100             continue
101
102         ## shorten distance between laws used in same case (multiply
    0.5)
103         idx_list = []
104         for k in range(len(case_law)) :
105             idx_list.append(law_list.index(case_law[k]))
106
107         for n1 in idx_list :
108             for n2 in idx_list :
109                 if n1 == n2 :
110                     continue
111                 distance_matrix[n1][n2] = distance_matrix[n1][n2]/2
112
113 # save custom metric as binary file
114 np.save('law_data/custom_metric',distance_matrix)
```

References

[1] M Ester, HP Kriegel, J Sander, X Xu, *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*, Knowledge Discovery and Data Mining, **96** (34) (1996) Portland, OR, AAAI Press, 226–231.

[2] G Hinton and ST Roweis. *Stochastic Neighbor Embedding*, Neural Information Processing Systems, **15** (2002), 833–840.

[3] Y.S. Jeon, J.J. Kim, *Identication of sentiment keywords association-based hotel network of hotel review using mapper method in topological data analysis*, Korean J. Appl. Stat., **33** (1) (2020), 75–86.

[4] J.H. Jeong, *GitHub*, `https://github.com/zeebraa00/kmapper_law_analysis`.

[5] J.H. Kim J.S. Lee, M.J. Lee, W.J. Kim J.S. Hong, *Term Mapping Methodology between Everyday Words and Legal Terms for Law Information Search System*, J. Intell. Inf. Syst., **18** (3) (2012), 137–152.

[6] H.K. Ko, S.J. Park, *Mathematics Anxiety Analysis using Topological Data Analysis*, East Asian Math. J., **34** (2) (2018), 177–189.

[7] Korea Employment Information Service, *The future legal market seen by lawyers and law graduates*, `https://www.keis.or.kr/user/bbs/main/137/775/bbsDataView/48893.do`, (2021).

[8] D.J. Lee, J.H. Jung, *Time Series Analysis of VLBI Data with TDA and deep learnings*, J. Korean Soc. of Surveying, Geodesy, Photogrammetry and Cartography (2019), 257–262.

[9] P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson and G. Carlsson, *Extracting insights from the shape of complex data using topology*, Sci. Rep., **3** (1) (2013), 1–8.

[10] D Ristovska, P Sekuloski, *MAPPER ALGORITHM AND IT'S APPLICATIONS*, Int. Sci. J. Math. Model., **3** (3) (2019), 79–82.

[11] E Schubert, J Sander, M Ester, HP Kriegel, X Xu, *DBSCAN revisited, revisited: why and how you should (still) use DBSCAN*, ACM Trans. Database Syst., **42** (3) (2017), ACM New York, NY, USA, 1–21.
[12] G Singh, F Mémoli, GE Carlsson, *Topological methods for the analysis of high dimensional data sets and 3D object recognition*, Eurographics Symposium on Point-Based Graphics, **91** (2007).
[13] L Van der Maaten and G Hinton, *Visualizing data using t-SNE*, J. Mach. Learn. Res., **9** (2008), 2579–2605.

Seonghun Kim
Department of Mathematics, SungKyunKwan University Suwon, 16419, Korea
E-mail: jicsaw4816@g.skku.edu

Jaeheon Jeong
Department of Mathematics, SungKyunKwan University Suwon, 16419, Korea
E-mail: zeebraa00@g.skku.edu