Information Modelling and Knowledge Bases XXXVI
Y. Kiyoki et al. (Eds.)

doi:10.3233/FAIA241584

Knowledge Modeling and Processing Based on Space Mapping: Concepts, Methods, and Applications

Xing Chen[1]
[1] *Department of Information & Computer Sciences*
Kanagawa Institute of Technology, Japan

Abstract. This paper proposes an innovative method for knowledge construction and application based on the concept of space mapping. Space mapping is a mathematical operation that maps a vector to a space, expressing the relationship between input and output information. By representing data as vectors and matrices, the method establishes a mathematical relationship between input and output using a "knowledge vector". The "dark-matter matrix" transforms the input vector into the knowledge vector, mapping the input data onto the output space. The approach is extended to "parallel spaces", allowing for independent knowledge vectors in each space.

The paper defines concepts related to space mapping, such as space-time mapping, time-space mapping, chain mapping, and parallel spaces. It presents schemes and methods for knowledge construction and application based on space mapping processing, including vector construction techniques and knowledge modeling methods like chain mapping and parallel spaces. These techniques enable handling complex data and implementing efficient, scalable knowledge construction and application schemes.

The method is illustrated with two application examples: robot navigation in a maze and image recognition using chain mapping. The results demonstrate the method's ability to handle temporal, high-dimensional, and heterogeneous data aspects, create non-redundant knowledge vectors, and generate different output information based on application requirements.

Keywords. Knowledge, knowledge presentation, knowledge generation, machine learning, semantic space, spatiotemporal space

1. Introduction

One of the challenges of knowledge modeling and processing is how to deal with the temporal aspect of data. Data can change over time, and the meaning and relevance of data can also vary depending on the context and the goal. For example, in stock trading, past trends and patterns of stock prices may not be applicable to the current situation, and the optimal decision may depend on market conditions and the trader's preferences. Therefore, it is necessary to develop a method that can capture the temporal dynamics of data and adjust knowledge accordingly.

[1] Xing Chen, 1030 Simo-Ogino, Atsugi-shi, Kanagawa 243-0292, Japan; chen@ic.kanagawa-it.ac.jp

Another challenge of knowledge modeling and processing is how to deal with the high-dimensional and heterogeneous aspects of data. Data can come from different sources and modalities, such as text, images, sound, etc., and each type of data may have different features and dimensions. For example, in image recognition, the input data is a high-dimensional image, and the output data is a low-dimensional label or category. Therefore, it is necessary to develop a method that can map different types of data to a common space and extract the relevant information.

In this paper, a method for knowledge construction and application based on the concept of space mapping is proposed. Space mapping is a mathematical operation that maps a vector to a space. The relationship between input and output information, which can be used for prediction, judgment, or action control, is expressed by space mapping. The mapping process is represented by vector calculation processing. The vectors used to map input data to a space are called knowledge vectors, and the vector that represents the input information is called the input vector. Some concepts and terms related to space mapping, such as space-time mapping, time-space mapping, chain mapping, parallel space, etc., are also defined.

Our method is inspired by previous research on semantic computing models [2, 3, 4, 5], which use matrices to represent the meaning of data and perform semantic calculations. The concept of "dark-matter" [1], a term used to describe time-related data that is hidden from our observation, is also a basis for our method. A matrix called the dark-matter matrix is used to transform the input information into the knowledge vector. Some methods for constructing the required vectors, such as input vectors, mapping vectors, and chain mapping construction, are also presented.

Based on this previous research, this paper introduces an innovative method for constructing and applying knowledge based on the concept of space mapping. By representing input and output data as vectors and matrices, the method establishes a mathematical relationship between them using a "knowledge vector". The "dark-matter matrix" is employed to transform the input data vector into the knowledge vector, which maps the input data onto the output data space. The approach is further extended to "parallel spaces", allowing for independent knowledge vectors in each space.

The paper also presents several theories and processing methods for knowledge modeling, including space-time mapping, chain mapping, knowledge construction, knowledge application, and parallel spaces. These techniques enable the handling of complex data and the implementation of knowledge construction and application schemes.

Our method has the following potential advantages and contributions:

- It can handle the temporal aspect of data by using the dark-matter matrix and time mapping.
- It can handle the high-dimensional and heterogeneous aspects of data by using space mapping and chain mapping.
- It can implement knowledge construction and application using vector calculation processing, which is efficient and scalable.
- It can create knowledge vectors that have no redundancy by ensuring that the similarity value of any two output vectors is above a given threshold.
- It can generate different output information according to different application requirements, such as prediction, judgment, or action control.

The structure of this paper is as follows:

- Section 2 reviews the concept of "dark-matter" and its relation to experience and knowledge.
- Section 3 describes the concept of space mapping and its types, such as space-time mapping, time-space mapping, and chain mapping.
- Section 4 presents the scheme for knowledge construction and application based on space mapping processing, and the methods for constructing the required vectors and matrices during the processing.
- Section 5 illustrates and explains our method with two application examples: a robot with four sensors navigating a maze, and image recognition using chain mapping.
- Section 6 concludes the paper and discusses future work.

2. Review of the concept of "dark-matter" and the "dark-matter learning model"

In this section, the concept of "dark-matter" and its relation to experience and knowledge are reviewed. The machine learning model created based on the concept of "dark-matter", which is called the "dark-matter learning model", is also reviewed. An extension of the "dark-matter learning model", which is called the "parallel spaces model", is also introduced. Mathematical equations are shown to explain how the model works.

The concept of "dark-matter" is based on the research works of semantic computing models [2, 3, 4, 5, 6, 7, 8, 9, 10], which use matrices to represent the meaning of data and perform semantic calculations. In the semantic computing models, data, which can be numbers, words, sentences, paragraphs, articles, or multimedia data [11, 12, 13, 14], are mapped by mapping functions into semantic spaces and become points in semantic spaces. Data with different meanings are mapped to different locations in the same space. The distances between the points of data are calculated. In this way, the semantic calculation is transformed into calculating Euclidean distances of those points. The closer the points are, the more similar the meanings of the data are in the semantic spaces.

The concept of "dark-matter" [1, 22] is used to describe the time-related data that is hidden from our observation. The time-related data can change over time and affect the meaning and relevance of the data. For example, in stock trading, the past trends and patterns of stock prices may not be applicable to the current situation, and the optimal decision may depend on the market conditions and the trader's preferences. Therefore, the temporal dynamics of data need to be captured and the knowledge need to be adjusted accordingly.

The "dark-matter learning model" [1, 20, 21] is a machine learning model that builds and uses knowledge based on the concept of "dark-matter". In this model, data can be either time-related or non-time-related, and they are represented by matrices. The input data are transformed into the output data by a special matrix called the dark-matter matrix. The dark-matter matrix has two components: the matter component and the dark-matter component. The matter component is a vector that reflects the observable data, such as sensor data. The dark-matter component is a matrix that contains random numbers, and it captures the hidden and chaotic data. The process of this model can be described by the following mathematical formulas, where $\mathbf{X}$ is the dark-matter matrix, $\mathbf{E}$ is the output data matrix, and $\mathbf{X}^{-1}$ is the inverse of $\mathbf{X}$:

$$
\begin{aligned}
\mathbf{C} &= \mathbf{X}^{-1}\mathbf{E} \\
\mathbf{E} &= \mathbf{X}\mathbf{C}
\end{aligned}
$$

If the matter component is a vector with n dimensions, then the dark-matter matrix $\mathbf{X}$ will be a square matrix with n rows and columns, which consists of the vector and another matrix, the dark-matter component of which has $\mathrm{n}-1$ rows and n columns.

The "parallel spaces model" [23] is an extension of the "dark-matter learning model" that allows multiple semantic spaces to exist simultaneously. This model is inspired by the research on deep neural networks [15, 16, 17, 18, 19], which use hierarchical structures and multiple layers to extract features and learn representations. In the parallel spaces model, each semantic space has its own dark-matter matrix $\mathbf{X}$ and output data matrix $\mathbf{E}$. When there are n parallel spaces, there can be up to n dark-matter matrices, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\mathrm{n}$, and n output data matrices, $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_n$. The knowledge learning and input data mapping processes are independent for each semantic space:

$$
\begin{aligned}
\mathbf{C}_\mathrm{i} &= \mathbf{X}_\mathrm{i}^{-1}\mathbf{E}_\mathrm{i} \\
\mathbf{E}_\mathrm{i} &= \mathbf{X}_\mathrm{i}\mathbf{C}_\mathrm{i}
\end{aligned}
$$

The parallel spaces model can handle more complex and diverse data and knowledge than the single-space model, as it can capture the different aspects and perspectives of the data and knowledge in different semantic spaces. It also allows for more flexibility and adaptability in constructing and applying knowledge, as the knowledge vectors and the mapping processes can be adjusted and optimized for each space separately or jointly.

3. Definitions and concepts related to space mapping and knowledge

In this section, some definitions and concepts related to mapping and knowledge in vector spaces are introduced. Examples are used to illustrate them.

3.1 Symbols

The following symbols are used in this section:

- $\mathbf{V}$: It is an n-dimensional row vector, which is composed of input data.
- $\mathbf{E}$: It is the output data vector corresponding to the input data vector $\mathbf{V}$.
- $\mathbf{C}$: It is an n-dimensional column vector, which can make the equation $\mathbf{E} = \mathbf{V} \times \mathbf{C}$ hold. In this paper, $\mathbf{C}$ is called the knowledge vector.

3.2 Mapping

The multiplication operation of two vectors is defined as the mapping one vector onto a space by another vector in this paper. According to this definition, the equation $\mathbf{E} = \mathbf{V} \times \mathbf{C}$ is defined as mapping the input data onto the output data space by the knowledge vector. Similarly, the equation $\mathbf{C} = \mathbf{X}^{-1} \times \mathbf{E}$ is defined as mapping the output data by the corresponding input data onto the knowledge space.

For example, when a 3-row and 4-column matrix, which is called the dark-matter, is combined with a given input data vector $\mathbf{V} = [1.02 \quad 0.02 \quad 0.02 \quad 1.02]$, a square matrix **X** with 4 rows and columns is constructed.

$$\mathbf{X} = \begin{pmatrix} 0.98 & 0.39 & 0.40 & 0.76 \\ 0.05 & 0.14 & 0.68 & 0.07 \\ 0.87 & 0.96 & 0.13 & 0.87 \\ 1.02 & 0.02 & 0.02 & 1.02 \end{pmatrix}$$

As the following, a 4-dimensional output vector **E** is mapped onto the knowledge space:

$$\mathbf{C} = \mathbf{X}^{-1} \times \mathbf{E} = \begin{pmatrix} 4.41 & -2.28 & -1.42 & -1.93 \\ -0.03 & -0.17 & 1.09 & -0.90 \\ 0.10 & 1.46 & -0.25 & 0.04 \\ -4.42 & 2.26 & 1.40 & 2.92 \end{pmatrix} \times \begin{bmatrix} 0.23 \\ 0.64 \\ 0.45 \\ 0.12 \end{bmatrix} = \begin{bmatrix} -1.33 \\ 0.26 \\ 0.85 \\ 1.43 \end{bmatrix}.$$

3.3 Time and Space mapping

When a column vector or a matrix is converted into a row vector, this conversion is called the mapping from space to time in this paper. For example, the transpose operation of the column vector $\mathbf{V}$, $\mathbf{V}^{\mathrm{T}}$, is defined as the mapping of the column vector $\mathbf{V}$ onto time in this paper. When a row vector is converted into a column vector or a matrix, this conversion is called the mapping from time to space in this paper.

3.4 Chain mapping

The chain mapping is a process of mapping the input data onto different spaces. In this process, the input data is first mapped onto a space S_1 by a knowledge vector, and a mapping vector is obtained. Then, this mapping vector is mapped from space to time and mapped onto another space S_2 by another knowledge vector, and a mapping vector on the space S_2 is obtained. This process is repeated until the input data is mapped onto the final space S_k by the last knowledge vector.

The chain mapping can be expressed by the following formula:

$$\mathrm{E} = (((\mathrm{V} \times \mathrm{C}_1)^{\mathrm{T}} \times \mathrm{C}_2)^{\mathrm{T}} \times \dots)^{\mathrm{T}} \times \mathrm{C}_k \,,$$

where V is the input data vector, E is the output data vector on the spaceS_k, and $C_1, C_2, \dots, C_k$ are the knowledge vectors for each space.

The chain mapping can be used to perform various tasks, such as data compression, dimensionality reduction, feature extraction, and so on. The advantages and applications of the chain projection will be discussed in the following sections.

3.5 Knowledge Construction

The process of finding the knowledge vector **C** is called the knowledge construction process in this paper. In this process, a knowledge vector is required for each parallel space. The "dark-matter learning model" is used to implement the process of solving the knowledge vector in this paper. In this process, the "dark-matter" matrix

X is first constructed by combining the input data vector with the "dark-matter", and then the output data vector corresponding to the input data is mapped onto the knowledge space by the inverse matrix of **X**, and finally the knowledge vector **C** is obtained,

$$\mathbf{C} = \mathbf{X}^{-1} \times \mathbf{E} .$$

3.6 Knowledge Application

The process of mapping the input data onto the output data space by using the knowledge vector is called the knowledge application process in this paper,

$$\mathbf{E} = \mathbf{V} \times \mathbf{C} ,$$

where **E** is the output data vector, **V** is the input data vector, and **C** is the knowledge vector.

4. Knowledge construction and knowledge application processing

This chapter presents the scheme for knowledge construction and knowledge application processing. First, an example is given, and then the processing steps of the scheme are introduced.

4.1 An example of knowledge construction and knowledge application processing

In the example, the input data consists of four column vectors, as shown in Figure 1. In the method introduced in this paper, each column vector needs to be transposed into a row vector, which is called the mapping to time in this paper.

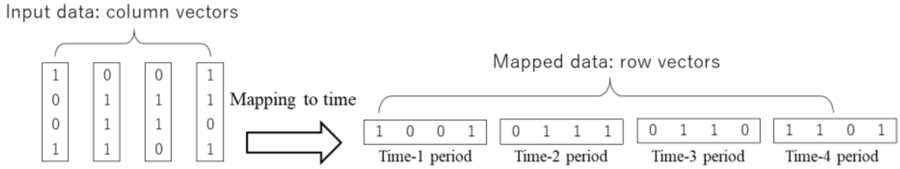

Figure 1. An example of input data and the its mapping to time processing

When using the algorithm in this paper, the values of the vector elements cannot be zero, and they cannot exceed a certain value range. Therefore, the vector needs to be normalized. The normalized vector is called the input data vector **V**. In this example, the normalization process is to add a small value of 0.02 to each element of the vector, as shown in Figure 2.

In Figure 2, the processing on creating the knowledge vector is shown.

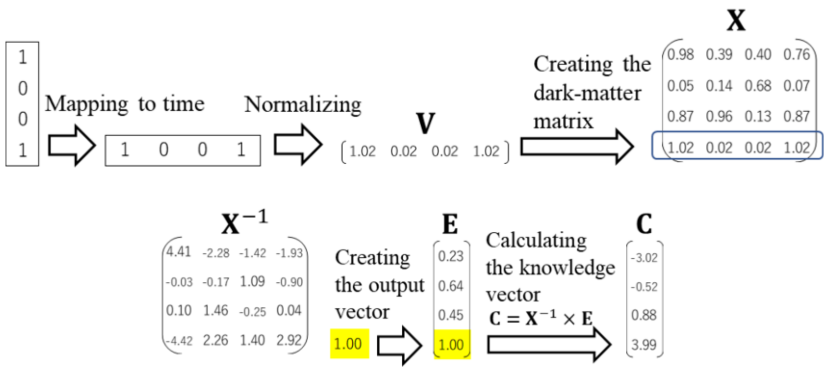

Figure 2. The processing on creating the knowledge vector

In the figure, the dark-matter matrix **X** is created as a 4×4 matrix, whose last row is the vector **V**, and the rest of the elements are random values.

The output vector **E** is created by adding the output value e to the last row of the column of vector **E**. In the example shown in Figure 2, the output data value is 2.00, $e = 2$. The column vector **E** has the same number of rows as the matrix **X**, except that the element of the last row is e, and the values of the rest of the elements are random numbers. In the example in Figure 2, **E** is a column vector with four elements.

The knowledge matrix **C** is obtained by multiplying the inverse matrix $\mathbf{X}^{-1}$ of **X** and the vector **E**: $\mathbf{C} = \mathbf{X}^{-1} \times \mathbf{E}$.

As shown in Figure 1, the input data is mapped onto four time periods, so a knowledge vector needs to be obtained for each time period, and a total of four knowledge vectors need to be obtained. Figure 2 shows the process of obtaining the first knowledge vector.

Figure 3 shows the process of creating the second knowledge vector.

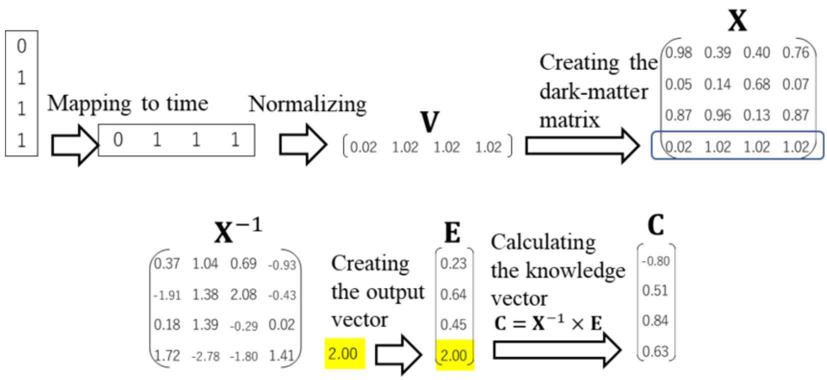

Figure 3. The processing on creating the second knowledge vector

In the method proposed in this paper, when total input data is not covered, chain mapping processing is required. In this example, because each time period does not cover all the input data, chain mapping processing is required.

In the first stage of the mapping process, the input data is mapped onto the space S_1 according to the time period, forming four mapping values, 1.00, 2.00, 3.00 and 4.00, as shown in Figure 4. According to the mapping values of these four time periods, four input vectors **V** composed of mapping values corresponding to each time period can be formed respectively.

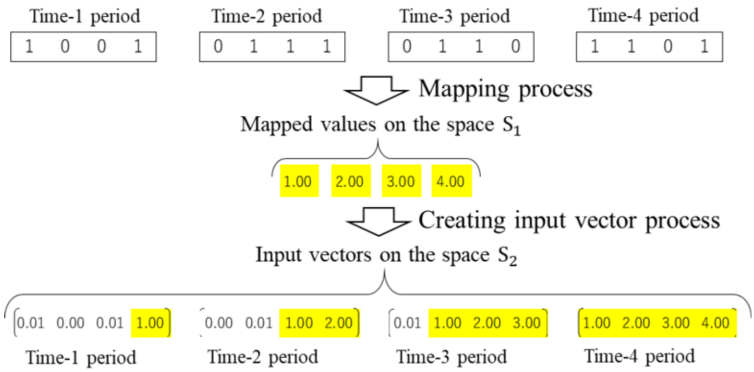

Figure 4. The processing on creating the input vectors on the space S_1

As shown in Figure 4, the input vector of mapping values corresponding to the first time period has only one mapping value, 1.00, and the values of the remaining three elements are three random values. The input vector of mapping values corresponding to the second time period has two mapping values, 1.00 and 2.00, and the remaining two elements are random values. The input vector corresponding to the third time period has three projection values, 1.00, 2.00 and 3.00, and the remaining one element is a random value. The input vector corresponding to the fourth time period has four mapping values, 1.00, 2.00, 3.00 and 4.00.

Figure 5 shows the process of creating the knowledge vector corresponding to the first time period. This process is the same as that shown in Figure 2.

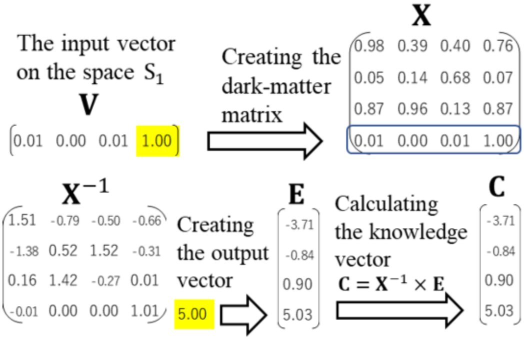

Figure 5. The process of creating the knowledge vector for the first time period on the space S_2

In the figure, the dark-matter matrix **X** is created as a 4×4 matrix, whose last row is the vector **V** on the spaceS_1, and the rest of the elements are random values.

The output vector **E** on the space S_1 is created with five elements by adding the output value e to the last row of the column of vector **E**. In the example shown in Figure 5, the output data value is 5.00, e = 5. The values of the rest of the elements of **E** are random numbers.

The knowledge matrix **C** is obtained by multiplying the inverse matrix $\mathbf{X}^{-1}$ of **X** and the vector **E**: $\mathbf{C} = \mathbf{X}^{-1} \times \mathbf{E}$.

When the knowledge vectors corresponding to the four time periods are obtained, the knowledge vectors covering all the input data are obtained.

Figure 6 shows the process of using the knowledge vectors. This is a chain mapping process. In the process, the input data is first mapped onto the space S_1, producing a mapping data of 1.00. Then, this mapping data is furtherly mapped onto the next space S_2, producing the final output data of 5.00.

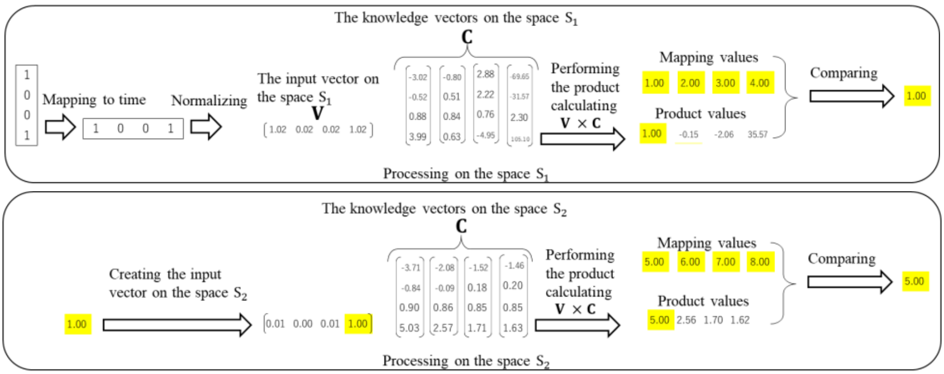

Figure 6. The process of using the knowledge vectors

The process shown in Figure 6 is performed as the following steps:

Step-1: The input data is mapped onto time, and a row vector [1001] is obtained. By normalizing this row vector, an input vector **V** on the space S_1 is constructed.

Step-2: The products of the input vector **V** and the four knowledge vectors on the space S_1 are calculated respectively, $\mathbf{V} \times \mathbf{C}$ and four product values are obtained, 1.00, -0.15, -2.06 and 35.37. These four product values are compared with the four mapping values on the space S_1, 1.00, 2.00, 3.00 and 4.00. Because the in the four calculated values, the value 1.00 has the smallest absolute value of the difference from the mapping value, the value 1.00 is taken as the output value on the space S_1.

Step-3: Using the output value on the space S_1, 1.00, an input vector on the space S_2 is constructed, and the products of it and the four knowledge vectors on the space S_2 are calculated respectively, and four values are obtained, 5.00, 2.56, 1.70 and 1.63. These four values are compared with the four mapping values on the space S_2, 5.00, 6.00, 7.00 and 8.00. Because the value 5.00 has the smallest absolute value of the difference from the mapping values, the value 5.00 is taken as the final output data.

4.2 Schemes of knowledge construction and knowledge application processing

The following are schemes of implementing knowledge construction and its application:

4.2.1 Parallel Spaces and Knowledge Vector Construction

1.1. According to the set time period, the input data is projected onto time, and an input row vector $\mathbf{V}$ is generated.
1.2. For the vector $\mathbf{V}$, a square dark-matter matrix $\mathbf{X}$ is constructed, whose bottom row is the vector $\mathbf{V}$, and the rest of the elements are random numbers. The inverse matrix $\mathbf{X}^{-1}$ of the matrix $\mathbf{X}$ is calculated.
1.3. For the output value e of the vector $\mathbf{V}$, a column matrix with the same number of rows as the dark-matter matrix $\mathbf{X}$, that is, the output matrix $\mathbf{E}$, is constructed. The value of the last row of this matrix is e, and the values of the rest of the elements are random numbers.
1.4. The knowledge vector $\mathbf{C}$ is obtained by the inverse matrix of $\mathbf{X}$ and the output matrix $\mathbf{E}$,

$$\mathbf{C} = \mathbf{X}^{-1} \times \mathbf{E}$$

1.5. A parallel space is constructed by the above steps 1.1 to 1.4. The above steps 1.1 to 1.4 are repeated repeatedly, and new parallel spaces are continuously constructed until no new spaces need to be constructed.
1.6. For the input vector $\mathbf{V}$ generated by step 1.1, if there is already a knowledge vector, then for any knowledge vector $\mathbf{C}$, the product of the input vector and the knowledge vector is calculated, $\mathbf{V} \times \mathbf{C}$. If the absolute value of the difference between the calculated value and each value of all the output values used in step 1.3 is less than a specified value, then there is no need to construct a new space, so steps 1.2, 1.3, and 1.4 can be skipped.

4.2.2 Chain Mapping Scheme

In the process of generating the input vector, if the generated input vector is only composed of a part of the input data, then a chain mapping structure needs to be constructed to cover all the input data. For the chain mapping space, the input data is first mapped onto the space S_1, and then the data on the space S_1 is mapped onto its following space S_2. This mapping process continues until the data is mapped onto the final output data space S_k.

The specific steps are as follows:

2.1. According to the set time period, the mapping value e of the input data mapped onto the space S_i , $S_i \in \{S_1, S_2, \dots, S_k\}$, is taken out, and e is further mapped onto time, generating a vector $\mathbf{V}$.
2.2. The knowledge vector is constructed for the generated vector $\mathbf{V}$. This construction process needs to skip the above step 1.1 and start processing from step 1.2 until the construction is completed.
2.3. If the knowledge vector construction process covering all the input data is completed, the process is ended, otherwise the following space needs to be constructed. If the space constructed this time is S_i, then the following space is

S_{i+1}. Repeat steps 2.1 and 2.2 until the knowledge vector of the last space S_k is constructed.

4.2.3 Knowledge Vector Application Scheme

After the knowledge vector construction is completed, the knowledge vector needs to be retained, and a knowledge vector set corresponding to the mapping space is constructed. In addition, all the mapping data on the mapping space needs to be retained, and a mapping data set is constructed for each mapping space.

3.1. For the given input data, an input row vector $\mathbf{V}$ is generated according to step 1.1.

3.2. A knowledge vector $\mathbf{C}$ is taken out from the knowledge vector set of the space S_i, $S_i \in \{S_1, S_2, \ldots, S_k\}$, and $e = \mathbf{V} \times \mathbf{C}$ is calculated. The value e is compared with all the data in the mapping data set on the space S_i, and a value with the smallest absolute value of the difference from e is selected as the output data. This step is repeated until all the knowledge vectors are processed and the value with the smallest absolute value of the difference from e is selected.

3.3. If the space S_i is not the final output data space, then a vector $\mathbf{V}$ is generated according to the above step 2.1, and then the processing of step 3.2 is performed until the final output space S_k is processed and the final output data is obtained.

5. Application examples

The method introduced in Section 4 is illustrated by using two application examples in this section. This section is divided into two parts. In the first part, the automatic moving of a robot is controlled by using the method proposed in this paper. The robot has four sensors. The control signal is obtained by multiplying the sensor data and the knowledge vector, which is used to control the robot to move automatically from the starting point to the destination point. The process of generating the knowledge vector is detailed. In the second part, image recognition is implemented by using the method proposed in this paper. The image recognition vale is obtained by multiplying the image information and the knowledge vector. The generation of knowledge vectors, making them suitable for image recognition processing, is detailed.

5.1 An example on the processing of the automatic moving of a robot

The sensor data and action control process of the robot are abstracted in the following introduction, omitting the specific details such as noise processing in the actual signal, position positioning, and moving control. This can highlight the main content that we want to describe.

The robot has four sensors, which can detect the forward routes in four directions. The output signal of the sensor is '1' when there is no obstacle in the forward route, and '0' when there is an obstacle in the forward route. The detection results of five different forward routes are shown in Figure 7. In the figure, the four detection directions are defined as 'up', 'down', 'left', and 'right'. The output signal of the sensor is a four-dimensional column vector, and the value on each dimension indicates whether there is an obstacle in that direction.

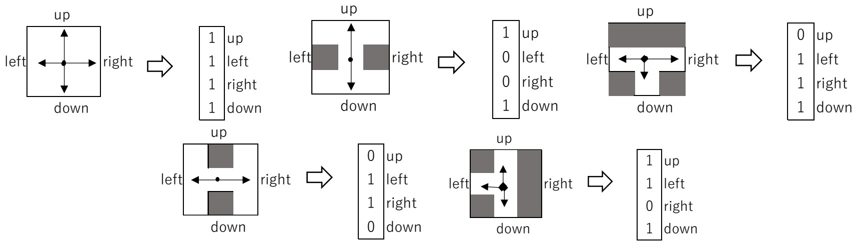

Figure 7. Detection results of five different forward routes

The moving route of the robot is shown in Figure 8. Four control signals are needed to control the robot to move in four directions, 'up', 'down', 'left', and 'right', in order to make the robot move from the starting point to the destination point. The control signals for moving 'up', 'right', and 'down', and 'left' are represented by three values 5, 6, 7. Control signals are sent based on the detection results of the sensors and the robot is controlled to move based on the control signals. The implementation details will be given next.

Figure 8. The robot's moving route and the required control signals

The output signals of the sensor at four time points, t_1, t_2, t_3, t_4, are shown in Figure 9. The robot needs to be provided with control signals at each time point. It is not feasible to provide control signals after receiving input signals from all four time points, therefore chain mapping processing is required.

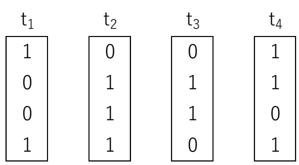

Figure 9. Output signals of the sensors at four time points

Two stages of chain mapping processing are used in this example. In the first-stage mapping processing, the mapping values of the sensor signals at four time points on the space S_1 are set to values 1, 2, 3, 4, as shown in Figure 10. The output signals of the sensors and the corresponding mapping output values, 1, 2, 3, 4, at each time point are also shown in this figure. The process of solving the knowledge vector after completing the mapping to time processing of the sensor signal, creating the output vector, is also shown in the figure. The four knowledge vectors solved are expressed by the 4×4 matrix **C** shown in the figure.

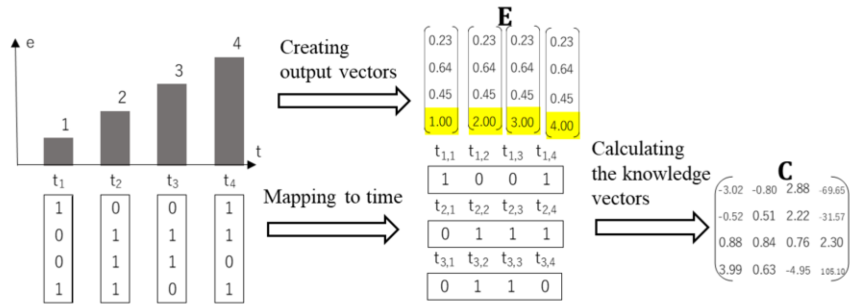

Figure 10. The first-level chain projection processing process of the control signal

The second-stage of the chain mapping processing is shown in Figure 11. In this processing process, the mapping values of the sensor signals at four time points on the space S_2 are set to values 5, 6, 6, 7. The output control signals at four time points, t_1, t_2, t_3, t_4, and the output vector **E** created according to the control signal, are shown in the figure. The input vector **V** after the normalization processing corresponding to each time point, and the four knowledge vectors expressed by the matrix **C** obtained, are also shown in the figure.

Figure 11 shows the process of producing a control signal. In the process, the input data is first mapped onto the space S_1, producing a mapping data of 1.00. Then, this mapping data is furtherly mapped onto the next space S_2, producing the control signal 5.00.

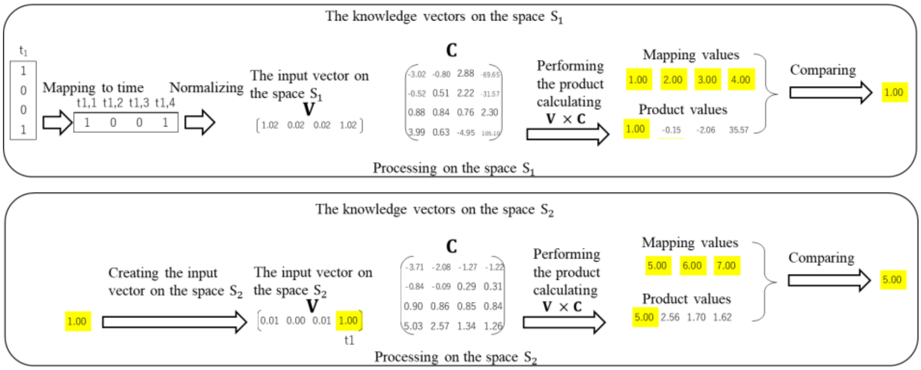

Figure 11. The process of producing a control signal

For implementing the movement control of the robot, the output signal of the sensor is first mapped to time and normalization processing is performed to obtain the input vector **V** on the space S_1. Then, the mapping value of the sensor signal on the space S_1 is calculated by multiplying the input vector and the knowledge matric, which is created based on the knowledge vectors, $\mathbf{V} \times \mathbf{C}$. After that, the input vector **V** on the space S_2 is created, according to the calculated mapping value, 1.00. The movement control signal of the robot is calculated by multiplying the input vector and the knowledge vector on the space S_2, $\mathbf{V} \times \mathbf{C}$, and the control signal 5.00 is obtained to control the robot moving up.

5.2 An example on the processing of the image recognition

As shown in Figure 12, there are three images. The images have the shapes of "X", "O", and "+". This example illustrates how the method proposed in this paper can be applied to recognize these three images. The recognition results of these three images are set as 1, 2, and 3, respectively.

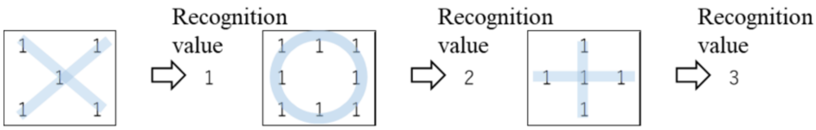

Figure 12. Three images and their recognition values

In this example, the three images are simplified. Each image is simplified into a 3×3 matrix. By using the simplified image matrices, the image recognition process can be simplified, and the method proposed in this paper can be explained more clearly.

A 2×2 sized window is used to scan the images. The images are scanned by using a "Z" shape scanning method. Four scanning results are shown in Figure 13. The scanning result of the window is a 2×2 matrix. After the time mapping process, the scanning result is transformed into a four-dimensional row vector. Since the window cannot cover the whole image, the chain mapping is needed. A two-stage chain mapping is used in this example, and the mapping spaces are S_1 andS_2.

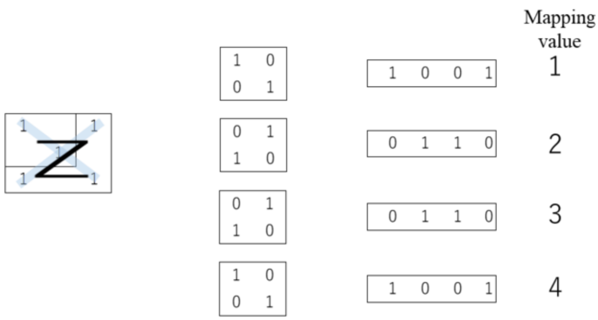

Figure 13. Four scanning results of an image and their mapping values on S_1

Each image has four scanning results, and the three images have eight scanning results. The four mapping values of the first image on the S_1 space are set as 1, 2, 3, and 4, as shown in Figure 13. The mapping values of the second and third images on the S_1 space are set as 5, 6, 7, 8, and 9, 10, 11, 12, respectively.

Figure 14 shows the process of solving a knowledge vector. This figure takes the scanning value of the first window of the first image as an example, and shows the process of solving the knowledge vector. In this process, the input data is normalized first, and then the dark-matter matrix $\mathbf{X}$ and the mapping vector $\mathbf{E}$ are constructed. The knowledge vector is obtained by the product of the inverse vector of **X** and the mapping vector **E**.

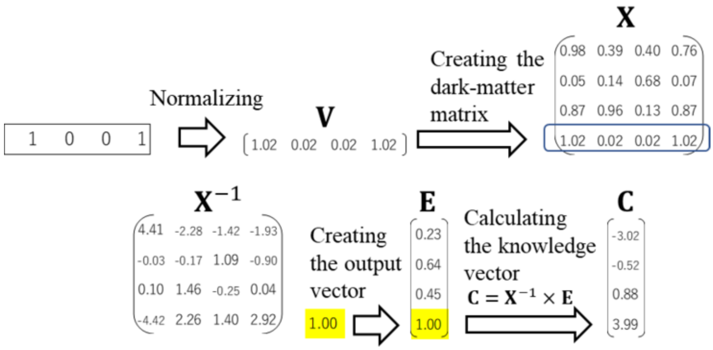

Figure 14. The process of solving a knowledge vector

Figure 15 shows the whole process of solving the knowledge vectors needed to recognize the three images. In this process, the input vectors are generated first, and then the knowledge vectors are calculated according to the mapping values. The figure shows that six knowledge vectors are obtained in the S_1 space. In the figure, these six knowledge vectors are expressed as a 4×6 matrix. Three knowledge vectors are obtained in the S_2 space, and they are expressed as a 4×3 matrix.

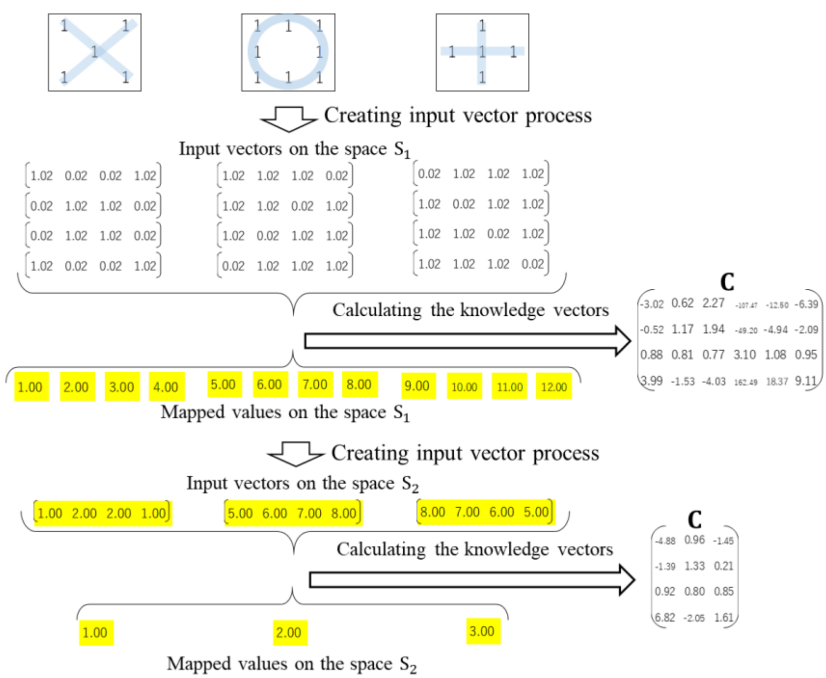

Figure 15. The process of solving the knowledge vectors needed to recognize the three images

Figure 16 shows the process of solving the recognition values of the three images. In this process, the images are converted into input vectors in the S_1 space first, and then the projection values are obtained by the multiplication operation of the input vectors and the knowledge matrix. Next, in the S_2 space, the projection values are converted into input vectors in the S_2 space, and then the projection values are obtained by the multiplication operation of the input vectors and the knowledge matrix in the S_2 space. The projection values in the S_2 space are the recognition values of the images.

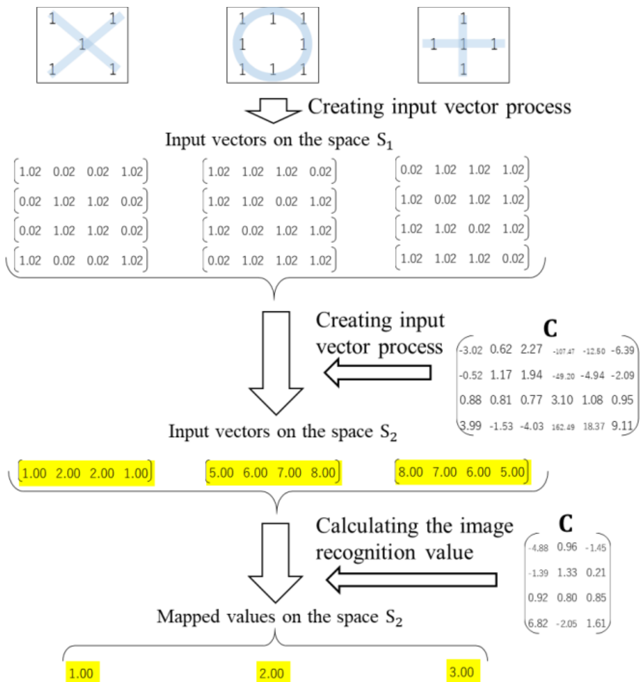

Figure 16. The process of solving the recognition values of the three images

6. Conclusion and future work

This paper proposes an innovative method for knowledge construction and application based on the concept of space mapping. Space mapping expresses the relationship between input and output information, which can be used for prediction, judgment, or action control. Vectors and matrices represent the mapping process, and concepts such as space-time mapping, time-space mapping, chain mapping, and parallel spaces are defined.

The method represents input and output data as vectors and matrices, establishing a mathematical relationship between them using a "knowledge vector". The "dark-matter matrix" transforms the input vector into the knowledge vector, mapping the input data onto the output space. The approach is extended to "parallel spaces", allowing for independent knowledge vectors in each space.

Several theory and processing methods for knowledge modeling are presented, including space-time mapping, chain mapping, knowledge construction, knowledge application, and parallel spaces. These techniques enable handling complex data and implementing knowledge construction and application schemes. The feasibility and effectiveness of the method are demonstrated through application examples, showcasing its potential for various domains involving knowledge modeling and processing.

In conclusion, this paper offers a fresh perspective on understanding and processing knowledge using space mapping, vectors, and matrices. It provides a new model for knowledge processing based on space mapping, and demonstrates its feasibility and effectiveness through theoretical discussions and practical examples, making it a promising contribution to the field of knowledge modeling.

Future work includes developing more application systems based on the method and evaluating their performance and usability. Exploring more types and methods of space mapping, extending the method to handle more complex and dynamic data and knowledge, and investigating the theoretical and practical implications of the method for knowledge representation and reasoning are also planned.

References

[1] Chen, X. and Kiyoki, Y., "*On Semantic Spatiotemporal Space and Knowledge with the Concept of "Dark-Matter"*," Information Modelling and Knowledge Bases XXXIII, IOS Press, pp.110-128, 2021.

[2] Chen, X. and Kiyoki, Y., "*A query-meaning recognition method with a learning mechanism for document information retrieval*," Information Modelling and Knowledge Bases XV, IOS Press, Vol. 105, pp.37-54, 2004.

[3] Chen, X. and Kiyoki, Y., "*A dynamic retrieval space creation method for semantic information retrieval*," Information Modelling and Knowledge Bases XVI, IOS Press, Vol. 121, pp.46-63, 2005.

[4] Kiyoki, Y. and Kitagawa, T., "*A semantic associative search method for knowledge acquisition*," Information Modelling and Knowledge Bases, IOS Press, Vol. VI, pp.121-130, 1995.

[5] Kitagawa, T. and Kiyoki, Y., "*A mathematical model of meaning and its application to multidatabase systems*," Proc. 3rd IEEE International Workshop on Research Issues on Data Engineering: Interoperability in Multidatabase Systems, pp.130-135, April 1993.

[6] Chen, X., Kiyoki, Y. and Kitagawa, T., "*A multi-language oriented intelligent information retrieval system utilizing a semantic associative search method,*" Proceedings of the 17th IASTED International Conference on Applied Informatics, pp.135-140, 1999.

[7] Chen, X., Kiyoki, Y. and Kitagawa, T., "*A semantic metadata-translation method for multilingual cross-language information retrieval,*" Information Modelling and Knowledge Bases XII, IOS Press, Vol. 67, pp.299-315, 2001.

[8] Kiyoki, Y., Kitagawa, T. and Hitomi, Y., "*A fundamental framework for realizing semantic interoperability in a multidatabase environment*, " International Journal of Integrated Computer-Aided Engineering, Vol.2, No.1(Special Issue on Multidatabase and Interoperable Systems), pp.3-20, John Wiley & Sons, Jan. 1995.

[9] Kiyoki, Y., Kitagawa, T. and Hayama, T., "*A metadatabase system for semantic image search by a mathematical model of meaning*, " ACM SIGMOD Record, Vol.23, No. 4, pp.34-41, Dec. 1994.

[10] Kiyoki, Y, Chen, X. and Kitagawa, T., "*A WWW Intelligent Information Retrieval System Utilizing a Semantic Associative Search Method*," APWeb'98, 1st Asia Pacific Web Conference on Web Technologies and Applications, pp. 93-102, 1998.

[11] Ijichi, A. and Kiyoki, Y.: "*A Kansei metadata generation method for music data dealing with dramatic interpretation*," Information Modelling and Knowledge Bases, Vol.XVI, IOS Press, pp. 170-182, May, 2005.

[12] Kiyoki, Y., Chen, X. and Ohashi, H.: "*A semantic spectrum analyzer for realizing semantic learning in a semantic associative search space*," Information Modelling and Knowledge Bases, Vol.XVII, IOS Press, pp.50-67, May 2006.

[13] Takano, K. and Kiyoki, Y.: "*A causality computation retrieval method with context dependent dynamics and causal-route search functions*," Information Modelling and Knowledge Bases, ISO Press, Vol.XVIII, pp.186-205, May 2007.

[14] Chen, X. and Kiyoki, Y.: "*A visual and semantic image retrieval method based on similarity computing with query-context recognition*," Information Modelling and Knowledge Bases, IOS Press, Vol.XVIII, pp.245-252, May 2007.

[15] Nitta T, "*Resolution of singularities introduced by hierarchical structure in deep neural networks*," IEEE Trans Neural Netw Learn Syst., Vol.28, No.10, pp.2282-2293Oct. 2017.

[16] Wiatowski, T. and Bölcskei, H., "*A Mathematical Theory of Deep Convolutional Neural Networks for Feature Extraction*," IEEE Transactions on Information Theory, PP(99) · Dec. 2015.

[17] Hochreiter, S., Bengio, Y., Frasconi, P. and Schmidhuber, J. "*Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*," In Kremer, S. C. and Kolen, J. F. (eds.), *A Field Guide to Dynamical Recurrent Neural Networks*, IEEE Press, 2001.

[18] Hochreiter, S. and Schmidhuber, J., "*Long short-term memory*,". Neural computation, Vol.9, No.8, pp.1735-1780, 1997.

[19] Kalchbrenner, N., Danihelka, I. and Graves, "*A. Grid long short-term memory*," CoRR, abs/1507.01526, 2015.

[20] Chen, X. and Kiyoki, Y., "*On Logic Calculation with Semantic Space and Machine Learning*," Information Modelling and Knowledge Bases XXXI, IOS Press, Vol. 321, pp.324-343, 2019.

[21] Chen, X. Prayongrat, M. and Kiyoki, Y., "*A Concept for Control and Program Based on the Semantic Space Model*," Information Modelling and Knowledge Bases XXXII, IOS Press, Vol. 333, pp. 26-44, 2020.

[22] Chen, X., "*An Exploratory Research on the Expression of Knowledge and Its Generation Process based on the Concept of "Dark-matter"*, Information Modelling and Knowledge Bases XXXIV, IOS Press, Vol. 364, pp. 110-124, 2023.

[23] Chen, X. and Kiyoki, Y., "*A Knowledge Model Based on "Dark-Matter" and Parallel Spaces*," Information Modelling and Knowledge Bases XXXV, IOS Press, Vol. 380, pp.271-286, 2024.