Applied Soft Computing 46 (2016) 17–25

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Uniforming the dimensionality of data with neural networks for materials informatics

Hiroshi Ohno

Toyota Central R&D Labs., Inc., 41-1 Yokomichi, Nagakute, Aichi 480-1192, Japan

ARTICLE INFO

Article history:
Received 19 May 2015
Received in revised form 17 February 2016
Accepted 17 April 2016
Available online 27 April 2016

Keywords:
Denoising autoencoders
Neural networks
Materials informatics
Noise injection
Regularization
Regression

ABSTRACT

Materials informatics is a growing field in materials science. Materials scientists have begun to use soft computing techniques to discover novel materials. In order to apply these techniques, the descriptors (referred to as features in computer science) of a material must be selected, thereby deciding the resulting performance. As a way of describing a material, the properties of each element in the material are used directly as the features of the input variable. Depending on the number of elements in the material, the dimensionality of the input may differ. Hence, it is not possible to apply the same model to materials with different numbers of elements for tasks such as regression or discrimination. In the present paper, we present a novel method of uniforming the dimensionality of the input that allows regression or discriminative tasks to be performed using soft computing techniques. The main contribution of the proposed method is to provide a solution for uniforming the dimensionality among input vectors of different size. The proposed method is a variant of the denoising autoencoder Vincent et al. (2008) [1] using neural networks and gives a latent representation with uniformed dimensionality of the input. In the experiments of the present study, we consider compounds with ionic conductivity and hydrogen storage materials. The results of the experiments indicate that the regression tasks can be performed using the uniformed latent data learned by the proposed method. Moreover, in the clustering task using these latent data, we observed distance preservation in data space, which is also the case for the denoising autoencoder. This result may enable the proposed method to be used in a broad range of applications.

1. Introduction

The development of materials informatics has resulted in significant progress in the modeling and prediction of material properties, thereby reducing the costs of real-world experiments, and is becoming a promising research field for soft computing (for example, [2–4]). By using soft computing techniques such as neural networks, evolutionary and genetic algorithms, and fuzzy modeling, materials scientists can more effectively search for novel materials. These techniques are used alone and in combination with quantum calculations for materials design. For example, a method combining density functional theory and an evolutionary algorithm was used to predict the crystal structure of $LiBeH_3$ (Hu et al. [5]).

By organizing the data into a material database, researchers can determine the relationships between material properties (for example, conductivity, the critical temperature of superconductors, and melting temperature) and the properties (for example, atomic number, atomic mass, and electron negativity) of the elements in the material. The properties of elements or their combinations are referred to as descriptors (or "features" in computer science). Once the relevant features are obtained, the predictions of properties and the modeling of materials becomes easier. The literature [6] describes five descriptor categories: constitutional, topological, physicochemical, structural, and quantum-chemical. For instance, Seko et al. [7] adopted the sum and product of the element properties, such as atomic number, atomic mass, and number of valence electrons as features involved in the prediction of the melting temperature of single and binary compounds. These are constitutional descriptors. In addition, the use of sum and product operations is based on the domain knowledge of the compounds, and is also found in [8].

From the viewpoint of domain knowledge, we introduce two categories of materials feature representation: expert and naive. Expert representation is preferable for a material property that has a well-known mechanism or theoretical model. As such, many features (descriptors) based on the underlying theory would be expert representations (for example, Table 1 in [3]). In naive representation, we generate features based on the properties of elements in a compound, which are represented by a vector that consists of the

E-mail address: oono-h@mosk.tytlabs.co.jp

http://dx.doi.org/10.1016/j.asoc.2016.04.017
1568-4946/© 2016 Elsevier B.V. All rights reserved.

Table 1
Comparison of generalization results for the test datasets on the linear regression task.

Method	RMSE	Correlation
Proposed method	0.0872 ± 0.0045	0.862 ± 0.0142
Multi-layer autoencoder	0.168 ± 0.0128	0.251 ± 0.146
Denoising autoencoder	0.162 ± 0.0124	0.344 ± 0.120
Kernel PCA	0.148 ± 0.0241	0.476 ± 0.173

properties of elements. The advantage of the naive representation is that it is applicable to material properties with poorly understood mechanisms or theoretical models. In addition, it is simple and easy to interpret. As such, we herein adopt the naive representation. While the naive representation has good characteristics, the length of the vectors (namely, the dimensionality of the data) differs depending on the number of elements in the compound. Therefore, we cannot, for example, use the same model for compounds with different numbers of elements in modeling and prediction tasks. Thus, we propose a method of uniforming the dimensionality of input data that allows us to perform tasks using regression and discrimination methods.

Uniforming dimensionality is related to dimensionality reduction methods. Recently, a number of non-linear dimensionality reduction methods have been proposed [9,10]. These methods address the limitations of linear (traditional) methods, such as principle component analysis (PCA) and multidimensional scaling. Kernel PCA [11] and the multi-layer autoencoder [12] are well-known examples. These linear and non-linear methods cannot, however, be adopted as methods of uniforming dimensionality because when these methods are applied to datasets of different dimensionality, the resultant dimensionality of each dataset, even though they may be the same, has a different meaning. In addition, these methods focus primarily on the dimensionality reduction of data. As far as we know, there has been no previous study on non-linear uniforming of the dimensionality of data. Therefore, the present study may be the first attempt to make uniform the dimensionality of data while simultaneously considering both the expansion and reduction of the dimensionality of data. Moreover, if the data size is insufficient for learning, combining this data with data of a different dimensionality will allow the overall data to be learned.

In the neural network literature, the training algorithms of Deep Belief Networks (Hinton et al. [13], Bengio [14]) and stacked autoencoders (Vincent et al. [15]) have brought about great progress. An autoencoder consists of an encoding function, which maps the input data into a latent space, and a decoding function, which reconstructs the input data from the latent space. In the non-linear case, neural networks are often used as the encoding and decoding functions. As a regularized autoencoder, Vincent et al. [1,15,16] have proposed the denoising autoencoder, in which the input data are corrupted by Gaussian noise, whereas the target data used in learning are the original (clean) input data. Noisy inputs are used in a learning neural network to enhance generalization performance (An [17]).

For uniforming the dimensionality of input data, we propose a variant of the denoising autoencoder, in which the input data are corrupted, and an extended part added to make the dimensionality of input uniform is also injected by Gaussian noise. In the latent space formed by the encoding function, we obtain a uniformed representation with inputs of different dimensionality. Thus, we can apply the regression or discriminative tasks to the uniformed input data.

In the experiment, we first compare the proposed method with the multi-layer autoencoder, the denoising autoencoder, and the kernel PCA for synthetic data. Next, we evaluate the proposed method using compounds of four to six elements in ion-conducting bulk materials and hydrogen storage materials composed of two to five elements. We then show that regression can be performed using the uniformed input data, as well as the robustness with respect to data size and number of elements. Moreover, in a clustering task using these data and the k-nearest neighbors (k-nn) method, we find distance preservation, i.e., consistency of class assignment, in the data space, which also holds for the case using the denoising autoencoder. We evaluate the distance preservation using the difference in class assignments between the latent data in the latent representation and the original data in the input space.

The remainder of the present paper is organized as follows. In Section 2, we present background information and define the problem formulation. In Section 3, we describe the learning algorithm of the proposed method in detail. In Section 4, we conduct experiments involving a regression task on synthetic data and for the modeling of ion conductivity and hydrogen storage, and, using the uniformed input data, compare the distance preservation of the proposed method and the denoising autoencoder. In Section 5, we discuss the experimental results, related research, and future studies. Finally, Section 6 concludes the study.

2. Background and problem formulation

Descriptors (features) in materials sciences are crucial for computational materials design. In the case of the underlying theory and empirically known mechanism of material properties, the features are easily identifiable. However, it is necessary to generate the features derived from the properties of elements (for example, electron negativity, atomic number, and atomic mass). With regard to the representation of features, we refer to the former as an expert representation and the latter as a naive representation. The advantage of the naive representation is that it is applicable to the case of material properties with poorly known mechanisms or theoretical models. It is necessary to incorporate the (molecule or crystal) structural features in the representation if two materials with the same composition have different properties. In the case of isomers, the melting temperatures of C_4H_6 are $-125.7\,^\circ$C for 1-butyne and $-32\,^\circ$C for 2-butyne, respectively.

In the naive representation, for example, as the features of compound AB, which is composed of elements A and B, the corresponding vector v is composed from the three properties of elements A and B as follows:

$$\boldsymbol{v} = (v_{11} \quad v_{12} \quad v_{21} \quad v_{22} \quad v_{31} \quad v_{32})^T = (v_{ij}), \quad i = 1, 2, 3, j = 1, 2,$$

where the index i denotes the property of element, and T denotes transpose.

Index j corresponds to atom A or B. The length of the vector is the product of the number of elements in the compound and the properties of the elements. Therefore, the length of the vector differs depending on the number of elements in the compound. Thus, for compounds with different numbers of elements, we cannot use the input variables vector as a feature of the compound to perform regression or discriminative tasks. Moreover, as shown in the experiments described below, for the data on compounds having various numbers of elements, regression cannot be conducted because of a lack of data. The overall data need to be used for the task. As such, when using the overall data including all number of elements, the input variables as the features of the compounds have to be composed for the task. Therefore, it is necessary for the length of the vector to be made uniform. Note that the physical meaning of the vector changes according to the element (atomic) permutations in the vector. Thus, we sort the elements of the vector by atomic number. For example, if the atomic number A (j = 1) is larger