Combination of Multiple Distance Measures for Protein Fold Classification

Chendra Hadi Suryanto, Hideitsu Hino, Kazuhiro Fukui
Graduate School of Systems and Information Engineering
University of Tsukuba, Japan
Email: chendra@cvlab.cs.tsukuba.ac.jp, hinohide@cs.tsukuba.ac.jp, kfukui@cs.tsukuba.ac.jp

***Abstract*—In structural biology, measuring the similarity between two protein structures is an essential task. The most common approach is to find the best alignment between two protein backbone structures and use the root mean square deviation (RMSD) of the superimposed alpha-carbon atom coordinates as the distance measurement. Other approaches extract features of the protein structures and the similarity measure is based on the extracted features. However, there is no single best approach, as each has its own advantages and limitations. One intuitive idea is that a better result can be obtained by combining complementary approaches. In this paper, we propose a new approach to protein fold classification, by introducing the concept of large margin nearest neighbor for combining multiple measures of distance between protein structures. We combine the Euclidean distance matrices of 12 features extracted from the amino acid sequence of the protein, the RMSD obtained from the geometrical alignment using Combinatorial Extension, and the canonical angles between the subspaces generated from the synthesized multi-view protein structure images. We demonstrate the effectiveness of the proposed method by classifying 27 fold classes of proteins in the Ding Dubchak dataset.**

***Keywords*-metric learning, large margin nearest neighbor, protein fold classification**

I. Introduction

Proteins are one of the most important substances in a living organism since proteins perform various crucial functions in the biological processes of cells, such as catalyst and transport. Protein structure is built from a sequence of amino acids, which fold into a 3D structure due to the interactions between the atoms and chemical bonding in the chain of amino acid molecules. It is known that proteins which have similar 3D folding share the same functionality, although occasionally this correlation does not exist for the poorly defined protein functions [1]. To support the analysis of proteins, such as the categorization of proteins based on their 3D folding, it is essential to have a reliable measure of the similarity between pairs of 3D structures. However, there is as yet no standard method for the automatic computation of the similarity between protein structures.

The most common approach for comparing two protein structures is to apply geometrical alignment to the 3D structures of the two proteins and compute the root mean square deviation (RMSD) of the pairs of superimposed alpha-carbon atom coordinates. The performance of this similarity measures relies mostly on the alignment technique, which is computationally expensive and hard, especially when the protein structures are very different. Well-known methods which use this approach are that of Dali [2] and Combinatorial Extension (CE) [3][4]. Other approaches extract features from the 3D protein structures, and the similarity is measured by comparing the extracted features. For examples, in [5] the protein structure is considered as an oriented open curve from which a compact 31-dimensional Gauss integral vector, called Gauss Integral Tuning (GIT), is generated. Then, the dissimilarity is defined as the Euclidean distance between the two GIT vectors. In [6] the similarity of 3D protein structures is measured by comparing the protein structure visualizations. First, 2D multiple-view images of a 3D protein structure are synthesized using 3D molecular graphics software. Next, the set of multiple-view protein images is represented as a low-dimensional subspace. Finally, the similarity is defined by the canonical angles between the two corresponding subspaces.

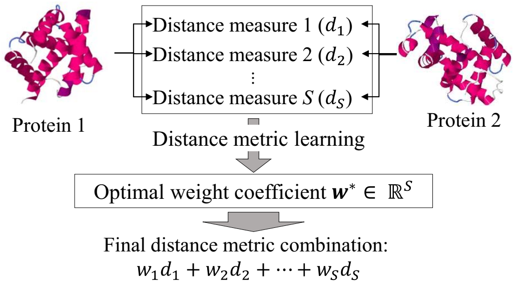
Protein 1
Distance measure 1 (d_1)
Distance measure 2 (d_2)
⋮
Distance measure S (d_S)
Protein 2
Distance metric learning
Optimal weight coefficient $\boldsymbol{w}^* \in \mathbb{R}^S$
Final distance metric combination:
$w_1 d_1 + w_2 d_2 + \cdots + w_S d_S$

Figure 1. Overview of the proposed method. The elements of vector $\boldsymbol{w}^* \in \mathbb{R}^S$ are the optimal coefficients for combining the distance measures.

Out of the similarity measures that have been proposed, there is no overall best since a particular approach may be superior in some cases but inferior in others. This leads to the expectation that a more reliable similarity measure can be obtained by combining multiple approaches. However, combining different similarity measures is not a trivial task because they have different metrics. In this paper, we propose a new approach for protein fold classification, that learns to combine multiple distance metrics using the formulation of large margin nearest neighbor (LMNN) [7]. An overview of our idea is shown in Figure 1. First, multiple distance measures are computed for each pair of protein

978-1-4799-2190-4/13 $26.00 © 2013 IEEE
DOI 10.1109/ACPR.2013.139

features in training data. Then these distance measures are fed into LMNN to obtain the optimal weight coefficients, $\boldsymbol{w}^*$, which are used for combining the measures. The distance metric learning algorithm is based on LMNN, because of the algorithm's effectiveness and simplicity. LMNN was originally proposed as an algorithm to learn a Mahalanobis distance metric for k-nearest neighbor (k-NN) problems. It ensures that the points belonging to the same class are moved closer to one another while at the same time the margin between different classes is enlarged. In this paper, we use the LMNN formulation to learn the optimal weight coefficients $\boldsymbol{w}^*$ for combining multiple distance metrics.

The validity of the proposed method is demonstrated through experiments on protein fold classification using the widely used Ding Dubchak protein dataset [8][9]. In the field of protein fold classification, there have been many attempts to predict protein fold categories using only the features extracted from protein sequences and without exploiting the 3D geometrical structure explicitly. For example, in [8] and [9], 12 types of features were extracted from protein sequences and their physico-chemical properties: amino acid composition, predicted secondary structure, hydrophobicity, van der Waals volume, polarity, polarizability, four types of pseudo-amino acid compositions, and two sequence alignments using Smith-Waterman scores. In our experiments, we combined the distance matrices of the 12 types of features extracted from protein sequences [9], the RMSD of the geometrical alignment of CE [3], the Euclidean distance of GIT feature vectors [5], and canonical angle similarity measures [6] in which we adopted the Constraint Mutual Subspace Method (CMSM) [10]. Since the dataset from [8][9] does not contain any geometrical features, such as alpha-carbon atom coordinates, we retrieved the geometrical features from the ASTRAL SCOP database [11][12].

The organization of this paper is as follows. First, we review LMNN in section II. Next, we formulate our proposed method for learning the optimal combination of multiple distance metrics using LMNN in section III. Then, we discuss the experimental results in section IV. Finally, conclusions and an indication of future work are provided in section V.

II. Large Margin Nearest Neighbor

In this section we briefly review the concept of LMNN [7]. Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ be a training set, where x_i is a data point and $y_i \in \{1, 2, \ldots, C\}$ is the class label of x_i. The objective of LMNN is to learn a distance measure such that the distances between data points in the same class are minimized and the margin between data points in different classes are maximized. Let the distance measure between two data points x_i and x_j be $d(x_i, x_j; \boldsymbol{w})$, where $\boldsymbol{w} \in \mathcal{W}$ specifies a concrete distance function. The goal of LMNN can be formalized as

$$\min_{\boldsymbol{w} \in \mathcal{W}} J(\boldsymbol{w}), \tag{1}$$

where $J(\boldsymbol{w})$ is

$$\sum_{i=1}^{n} \sum_{j:y_j=y_i} \left[d^2(x_i, x_j; \boldsymbol{w}) + \mu \sum_{h:y_h \neq y_i} [L(i,j,h;\boldsymbol{w})]_+ \right], \tag{2}$$

$$L(i,j,h;\boldsymbol{w}) = l + d^2(x_i, x_j; \boldsymbol{w}) - d^2(x_i, x_h; \boldsymbol{w}). \tag{3}$$

In this objective function, $\mu > 0$ is a balancing parameter, $l \geq 0$ is a margin parameter, and $[x]_+$ is the hinge loss defined as $[x]_+ = \max(0, x)$. In the original formulation of LMNN, the distance d is parameterized as a Mahalanobis distance, while in this work the distance is parameterized as a convex combination of multiple distance measures. This formulation of distance metric learning is similar to multiple kernel learning [13][14], which is intensively studied in the field of machine learning.

III. Distance Metric Learning using LMNN

A similarity and a distance measure are basically interchangeable; it is possible to convert a similarity measure to a distance measure and vice versa. In this paper, we combine multiple distance metrics with the similarity measure based on canonical angles. Each distance can be normalized to $0 \leq D_{ij} \leq 1$, by redefining $D_{ij} = D_{ij}/\max(\boldsymbol{D})$, where D_{ij} is the scalar distance between x_i and x_j, and the $n \times n$ matrix $\boldsymbol{D}$ is the distance matrix for n training samples. In the case of similarity measures using canonical angles, since $0 \leq \cos^2 \theta_{ij} \leq 1$, converting canonical angles to normalized distance measures can be simply done by setting $D_{ij} = 1 - \cos^2 \theta_{ij}$.

A. Convex Combination of Multiple Distance Measures

Distance measure between data points have to be nonnegative. Thus, a convex combination of S distance measures can be written as

$$d(x_i, x_j; \boldsymbol{w}) = w_1 d_1(x_i, x_j) + \cdots + w_S d_S(x_i, x_j), \tag{4}$$

where $d_s(x_i, x_j)$, $s = 1, \ldots, S$ are given distance measures, such as the RMSD of CE and canonical angles, between x_i and x_j. The weight of each distance measure $\boldsymbol{w} = (w_1, \ldots, w_S)^\intercal \in \mathbb{R}^S$ can be regarded as an element of the S-simplex which must satisfy

$$\sum_{s=1}^{S} w_s = 1, \quad w_s \geq 0, \ \forall s \in \{1, \ldots, S\}. \tag{5}$$

B. Formulation using the subgradient method

The distance combination from (4) is optimized by minimizing (2). Let $\boldsymbol{d}_{ij} = (d_1(x_i, x_j), \ldots, d_S(x_i, x_j))^\intercal \in \mathbb{R}^S$ and $d(x_i, x_j; \boldsymbol{w}) = \boldsymbol{w}^\intercal \boldsymbol{d}_{ij}$. The objective function $J(\boldsymbol{w})$ can be rewritten as

$$J(\boldsymbol{w}) = \boldsymbol{w}^\intercal \boldsymbol{M} \boldsymbol{w} + \mu \sum_{i=1}^{n} \sum_{j:y_j=y_i} \sum_{h:y_h \neq y_i} [L(i,j,h;\boldsymbol{w})]_+ \tag{6}$$

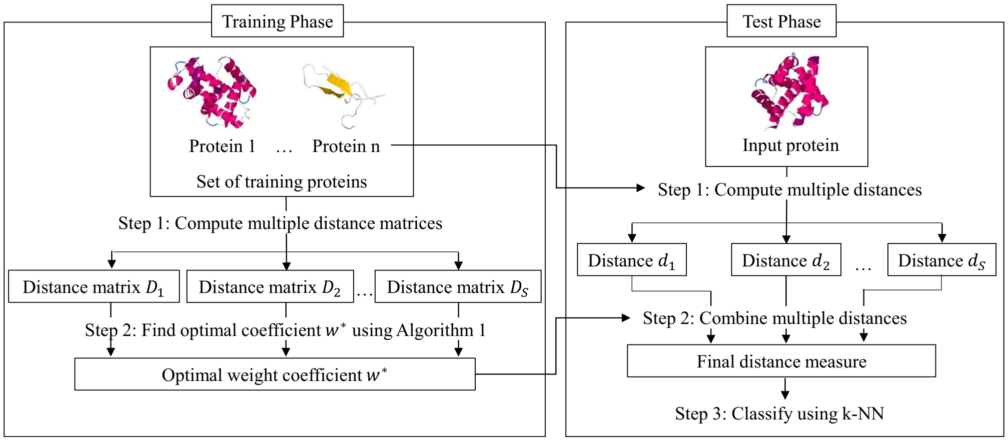
Training Phase
Protein 1 … Protein n
Set of training proteins
Step 1: Compute multiple distance matrices
Distance matrix D_1
Distance matrix D_2
…
Distance matrix D_S
Step 2: Find optimal coefficient w^* using Algorithm 1
Optimal weight coefficient w^*
Test Phase
Input protein
Step 1: Compute multiple distances
Distance d_1
Distance d_2
…
Distance d_S
Step 2: Combine multiple distances
Final distance measure
Step 3: Classify using k-NN

Figure 2. The flow of the classification using the proposed method.

$$L(i,j,h;\boldsymbol{w}) = l + \boldsymbol{w}^\mathsf{T}\boldsymbol{d}_{ij}\boldsymbol{d}_{ij}^\mathsf{T}\boldsymbol{w} - \boldsymbol{w}^\mathsf{T}\boldsymbol{d}_{ih}\boldsymbol{d}_{ih}^\mathsf{T}\boldsymbol{w}, \qquad (7)$$

$$\boldsymbol{M} = \sum_{i=1}^{n}\sum_{j:y_j=y_i} \boldsymbol{d}_{ij}\boldsymbol{d}_{ij}^\mathsf{T}. \qquad (8)$$

The objective function $J(\boldsymbol{w})$ contains the hinge loss $[x]_+$, which is not differentiable. However, it is possible to obtain a subgradient with respect to $\boldsymbol{w}$. Therefore, we can formulate an iterative method based on a subgradient to solve the convex minimization problem. In general, the subgradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ at $\boldsymbol{x}$ is a vector $\boldsymbol{z} \in \mathbb{R}^n$ such that $f(\boldsymbol{x}) + \boldsymbol{z}^\mathsf{T}(\boldsymbol{x}' - \boldsymbol{x}) \leq f(\boldsymbol{x}')$ for any $\boldsymbol{x}' \in \mathbb{R}^n$. For $J(\boldsymbol{w})$ the subgradient of $L(i,j,h;\boldsymbol{w})$ is given by

$$\boldsymbol{g}_{ijh} = \begin{cases} 2(\boldsymbol{d}_{ij}\boldsymbol{d}_{ij}^\mathsf{T} - \boldsymbol{d}_{ih}\boldsymbol{d}_{ih}^\mathsf{T})\boldsymbol{w} & L(i,j,h;\boldsymbol{w}) > 0 \quad (9) \\ 0 & \text{otherwise}, \quad (10) \end{cases}$$

where $\boldsymbol{g}_{ijh} \in \mathbb{R}^S$. After updating w using the subgradient $\boldsymbol{g}_{ijh}$, the next step is the projection to the S-simplex to ensure the constraint in (5). The complete iterative algorithm using the subgradient method is shown in Algorithm 1.

```
Algorithm 1 Subgradient algorithm to minimize $J(\boldsymbol{w})$
  Initialize: $\boldsymbol{w}_0 = (1/S, ..., 1/S)$, $\mu > 0$, $\epsilon > 0$, $l > 0$, $Threshold > 0$, and $\boldsymbol{M}$ as in (8).
  while $(||\boldsymbol{w} - \boldsymbol{w}_{\text{old}}||_2 > Threshold)$ do
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \epsilon \left\{ \boldsymbol{M}\boldsymbol{w} + \mu \sum_{i=1}^{n} \sum_{j:y_j=y_i} \sum_{h:y_h \neq y_i} \boldsymbol{g}_{ijh} \right\}$
    for $s = 1, \dots, S$ do: if $w_s < 0$ then $w_s \leftarrow 0$
    for $s = 1, \dots, S$ do: $w_s \leftarrow \frac{w_s}{\sum_{s=1}^{S} w_s}$
  end while
  Output: optimal coefficient $\boldsymbol{w}^*$
```

C. Flow of the classification

The flow of the classification is shown in Figure 2.

1) Training phase:

Step 1: Calculate S distance matrices for the training samples, $\{\boldsymbol{D}_1, \boldsymbol{D}_2, \dots, \boldsymbol{D}_S\}$.

Step 2: Find optimal coefficients $\boldsymbol{w}^*$ using Algorithm 1.

2) Test phase:

Step 1: Compute S distance measurements between an input protein and each training protein to obtain distances $\{d_1, d_2, \dots, d_S\}$.

Step 2: Combine $\{d_1, d_2, \dots, d_S\}$ and $\boldsymbol{w}^*$ as in (4) to obtain the final distance measure.

Step 3: Classify the input protein to the fold category with the smallest distance using k-NN.

IV. Experiments and Results

We used the Ding Dubchak dataset [8][9] to evaluate the proposed method. There are 12 features extracted from the protein sequences including the additional features from [9]: amino acid composition (C), predicted secondary structure (S), hydrophobicity (H), van der Waals volume (V), polarity (P), polarizability (Z), pseudo-amino acid compositions (λ_1, λ_4, λ_{14}, λ_{30}), and attributes from sequence alignment using Smith-Waterman scores (SW_1, SW_2). The dataset contains 27 fold categories, of proteins which are distributed nonuniformly among 313 training and 385 testing proteins. Due to the lack of a sequence record for some proteins [9] and because some GIT features could not be extracted using tools from [5], only 298 training proteins and 383 testing proteins were used. Examples of the proteins are shown in Figure 3. The protein with PDB code 2lhb belongs to fold *Globin-like*; 1ccr belongs to *Cytochrome c*; and 1enh belongs to *DNA-binding 3-helical bundle*. As mentioned previously, this database does not contain any geometrical features. Thus, to apply 3D structure based methods, we downloaded atom coordinate data for each protein from the ASTRAL SCOP database [12].

To evaluate the performance of the proposed method, we adopted the accuracy scheme of [8] in which accuracy is

Figure 3. Examples of protein structures in 3D visualization.

defined as the ratio of the number of correctly classified testing data to the total number of testing data. The overall accuracy is defined as a weighted accuracy with weights proportional to the number of the testing data in the fold category.

A. Distance matrix construction and parameter setting

We constructed distance matrices for each of the 12 features extracted from protein sequences by computing the Euclidean distance between the feature vectors of each of the training proteins. Since there were 298 training proteins, the size of each distance matrix was 298×298. GIT features were extracted using the tool from [5] and the distance matrix was constructed similarly. The tool from [4] was used to construct the CE distance matrix. For canonical angle based similarity, we followed the procedure described in [6]. First we synthesized the backbone, ribbons, and rockets visualizations for each protein structure using 3D molecular graphics software (Jmol [15]) by randomly rotating the 3D protein model 3000 times. After 3000 images at a size of 128×128 pixels had been collected for each visualization, HLAC [16] feature extraction was applied to each image and a linear subspace was generated for each image set. However, instead of directly computing the canonical angles between the two subspaces, we adopted the Constraint Mutual Subspace Method (CMSM) [10]. In CMSM, a constraint subspace $\mathcal{C}$ is generated by taking the M eigenvectors corresponding to the M lowest eigenvalues of the sum of the projection matrices of dictionary subspaces as the basis vectors. Here, each dictionary subspace was generated by applying PCA to the merged set of the training proteins belonging to the same fold category. In the computation of the similarity, the subspace corresponding to each protein was projected to the constraint subspace $\mathcal{C}$. The similarity is then defined by the canonical angles between the two projected subspaces. The distances and the canonical angle measures were finally normalized as described in section III.

In the implementation of Algorithm 1, the balancing parameter μ and the margin l were both set to values ranging from 0.01 to 1 in steps of 0.05. The iteration step size ϵ and the threshold used as the stopping criterion were both fixed to 10^{-5}. The parameter k in the k-NN procedure was set to values ranging from 1 to 20.

Table III
EXPERIMENTAL RESULTS OF THE PROPOSED METHOD (%)

Fold	DM-1	DM-2	DM-3	DM-4	DM-5	DM-6
1	83.33	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
2	77.78	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
3	40.00	90.00	90.00	90.00	90.00	**100.00**
4	62.50	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
5	77.78	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
6	22.22	**88.89**	**88.89**	**88.89**	**88.89**	77.78
7	47.73	90.91	90.91	**93.18**	90.91	**93.18**
8	25.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
9	53.85	84.62	84.62	92.31	92.31	**100.00**
10	16.67	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
11	37.50	87.50	87.50	87.50	87.50	**100.00**
12	26.32	**94.74**	89.47	89.47	89.47	**94.74**
13	75.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
14	25.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
15	42.86	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
16	54.17	97.92	97.92	97.92	97.92	**100.00**
17	66.67	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
18	38.46	76.92	76.92	**84.62**	76.92	76.92
19	62.96	**85.19**	**85.19**	**85.19**	**85.19**	81.48
20	41.67	**75.00**	**75.00**	**75.00**	**75.00**	66.67
21	25.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
22	75.00	**100.00**	**100.00**	**100.00**	**100.00**	66.67
23	57.14	85.71	85.71	**100.00**	**100.00**	**100.00**
24	25.00	75.00	75.00	75.00	75.00	**100.00**
25	25.00	75.00	75.00	**87.50**	**87.50**	75.00
26	37.04	85.19	85.19	**88.89**	**88.89**	70.37
27	92.59	**96.30**	81.48	85.19	88.89	92.59
Overall	50.91	91.91	90.60	**92.43**	92.17	91.12

B. Initial experiment

We conducted an initial experiment using conventional k-NN for each distance measurement. The best results for various values of k are shown in Table I. In structural biology, it is already known that protein fold category is determined by the geometry of the 3D structure. Therefore, the classification results using CE, GIT, and CMSM are much better than those using the 12 extracted features, as expected. These experimental results also confirm that one method might be superior for some cases and inferior for others. When focusing on the measures based on the 3D structure, for example, CE was inferior for recognizing fold 18 with only 46.15% accuracy, while GIT had 76.92% accuracy. On the other hand, both CE and GIT had an accuracy of 50% for fold 20, while CMSM achieved an accuracy of 58.33%.

C. Experiments using the proposed metric learning

To understand how the metric learning improves the classification performance, we used several combinations of distance matrices as shown in Table II. The experimental results are shown in Table III.

First, only the 12 distance matrices based on the extracted features were used (DM-1). From the results shown in Table III (DM-1), we can see that by learning the optimal combination of distances from the 12 extracted features, the overall accuracy improved from 45.43% to 50.91%.

Next, the CE distance matrix was added (DM-2). As shown in Table III (DM-2), the overall accuracy improved

Table I
INITIAL EXPERIMENTAL RESULTS IN TERMS OF ACCURACY FOR EACH FOLD (%)

Fold	(C)	(S)	(H)	(V)	(P)	(Z)	(λ_1)	(λ_4)	(λ_{14})	(λ_{30})	(SW$_1$)	(SW$_2$)	(CE)	(GIT)	(CMSM)
1	83.33	66.67	66.67	50.00	66.67	50.00	83.33	83.33	50.00	50.00	50.00	66.67	**100.00**	**100.00**	**100.00**
2	33.33	22.22	33.33	22.22	22.22	22.22	33.33	33.33	33.33	33.33	33.33	33.33	**100.00**	44.44	77.78
3	30.00	30.00	30.00	25.00	15.00	20.00	20.00	20.00	20.00	25.00	15.00	35.00	**100.00**	60.00	70.00
4	62.50	50.00	25.00	25.00	37.50	25.00	50.00	62.50	37.50	37.50	37.50	37.50	87.50	**100.00**	**100.00**
5	77.78	77.78	44.44	44.44	55.56	44.44	55.56	77.78	55.56	55.56	55.56	55.56	**100.00**	88.89	77.78
6	33.33	11.11	22.22	0.00	11.11	0.00	44.44	22.22	22.22	11.11	100.00	66.67	**88.89**	77.78	11.11
7	43.18	31.82	34.09	38.64	31.82	34.09	34.09	40.91	40.91	43.18	20.45	22.73	**95.45**	79.55	90.91
8	16.67	25.00	8.33	33.33	25.00	16.67	16.67	16.67	16.67	16.67	16.67	16.67	**100.00**	91.67	58.33
9	53.85	15.38	30.77	38.46	23.08	46.15	61.54	61.54	76.92	38.46	23.08	69.23	84.62	**92.31**	84.62
10	33.33	33.33	16.67	33.33	16.67	16.67	33.33	33.33	33.33	33.33	50.00	50.00	**100.00**	50.00	**100.00**
11	0.00	0.00	12.50	25.00	12.50	12.50	12.50	12.50	0.00	0.00	37.50	25.00	**75.00**	25.00	62.50
12	15.79	21.05	21.05	21.05	26.32	26.32	21.05	21.05	21.05	15.79	21.05	26.32	**94.74**	73.68	63.16
13	50.00	50.00	25.00	50.00	25.00	50.00	75.00	75.00	75.00	50.00	75.00	75.00	**100.00**	**100.00**	**100.00**
14	50.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	50.00	75.00	50.00	**100.00**	75.00	75.00
15	57.14	28.57	28.57	28.57	57.14	28.57	57.14	71.43	42.86	42.86	42.86	42.86	**100.00**	85.71	85.71
16	75.00	62.50	45.83	43.75	50.00	50.00	45.83	41.67	31.25	29.17	14.58	16.67	91.67	91.67	**100.00**
17	50.00	75.00	41.67	33.33	33.33	33.33	50.00	41.67	58.33	50.00	33.33	33.33	**100.00**	**100.00**	**100.00**
18	23.08	30.77	15.38	15.38	23.08	15.38	7.69	15.38	7.69	7.69	7.69	15.38	46.15	**76.92**	53.85
19	33.33	33.33	25.93	29.63	22.22	18.52	18.52	25.93	29.63	22.22	14.81	18.52	**92.59**	74.07	74.07
20	50.00	41.67	50.00	33.33	58.33	41.67	33.33	25.00	16.67	16.67	16.67	16.67	50.00	50.00	**58.33**
21	25.00	37.50	25.00	12.50	25.00	25.00	25.00	12.50	12.50	12.50	12.50	12.50	**100.00**	75.00	50.00
22	50.00	50.00	58.33	50.00	50.00	50.00	50.00	41.67	41.67	50.00	33.33	33.33	41.67	**91.67**	50.00
23	57.14	42.86	57.14	57.14	42.86	42.86	42.86	42.86	57.14	57.14	28.57	28.57	85.71	**100.00**	71.43
24	25.00	50.00	75.00	50.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	25.00	50.00	**100.00**	75.00
25	25.00	37.50	25.00	25.00	25.00	37.50	25.00	25.00	25.00	25.00	25.00	25.00	50.00	**87.50**	37.50
26	25.93	18.52	18.52	22.22	22.22	18.52	7.41	11.11	14.81	7.41	7.41	18.52	**85.19**	70.37	51.85
27	81.48	59.26	14.81	48.15	14.81	18.52	66.67	51.85	14.81	14.81	**88.89**	81.48	81.48	18.52	62.96
Overall	45.43	38.90	31.33	33.42	31.07	30.03	35.77	35.51	30.55	27.94	29.50	32.64	**86.68**	74.67	73.89

Table II
DISTANCE METRICS COMBINATIONS

	Amino acid sequence features												3D structure based		
Combination	(C)	(S)	(H)	(V)	(P)	(Z)	(λ_1)	(λ_4)	(λ_{14})	(λ_{30})	(SW$_1$)	(SW$_2$)	(CE)	(GIT)	(CMSM)
DM-1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
DM-2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
DM-3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
DM-4	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
DM-5	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DM-6													✓		✓

significantly to 91.91%. In DM-3, the distance measure based on GIT was added in the combination. The overall accuracy of DM-3 was better than the single distance measure, although it is not better than that of DM-2. In DM-4, the GIT distance matrix was replaced with the CMSM distance matrix, and we see that this combination could boost the accuracy to 92.43%. In DM-5 we combined all the distance metrics. The overall accuracy of this combination was 92.17%. Finally, we tried the combination of only the CE and CMSM distance matrices (DM-6). As can be seen by comparing the overall results for DM-4 and DM-6 in Table III, omitting the features extracted from the amino acid sequences made the performance is slightly worse. These experimental results suggest that the extracted features from protein sequences contain distinctive information for protein fold classification.

D. Discussions

In this part, we discuss the problem when using naive approaches for combining the multiple distance metrics.

Table IV
EXPERIMENTAL RESULTS USING SIMPLE METHODS (%)

Method	DM-1	DM-2	DM-3	DM-4	DM-5	DM-6
Averaging	46.21	61.10	61.10	68.67	68.41	90.86
Voting	48.56	54.83	62.66	62.40	71.54	NA

Table IV shows the recognition rates for two simple approaches for combining the multiple distance metrics. In the averaging method, the average of all the distances was used. Comparing the results in Table IV (DM-2, DM-3, DM-4, and DM-5) with those in Table III shows that the accuracy is significantly lowered if there are only few good measures available. The averaging method only performed better than a single distance measurement when only the CE and CMSM measures were used (DM-6). In the voting approach, the input protein is categorized to the fold category that has the most votes among the distance metrics. The voting approach has similar problems to the averaging method, although it can perform better than averaging when there are

Table V
OPTIMAL WEIGHT COEFFICIENT IN $\boldsymbol{w}^*$ (%)

Combination	Amino acid sequence		3D structure based		
	(C)	(S)	(CE)	(GIT)	(CMSM)
DM-4	0.45	18.32	62.55	0	18.68
DM-5	3.27	18.46	56.98	1.86	19.43

more measurements available for the voting. On the other hand, the voting approach is impossible if there are only two measurements available, such as in DM-6.

Unlike the naive approaches, distance metric learning using LMNN can produce a high recognition rate, although not all the distance measures perform well. However, the results of the proposed method also show that combining more distance measures did not always improve performance. For example, the result of combining all the 15 types of measurements (DM-5) was slightly worse than just combining 14 measurements (DM-4). To understand the contribution of each distance measure in the classification, the coefficients in $\boldsymbol{w}^*$ from DM-4 and DM-5 are shown in Table V (Coefficients that are zero for both DM-4 and DM-5 are not shown). We can see that the optimal coefficients are quite similar with the highest contribution coming from the distance matrix of the geometrical alignment (CE), followed by the appearance of the protein structure visualization (CMSM) and two features extracted from the protein sequences ((C) and (S)).

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new approach for protein fold classification that combines multiple distance measures using an algorithm based on large margin nearest neighbor (LMNN). We adapted the formulation of LMNN, which was originally proposed for Mahalanobis distance learning, for multiple distance metric learning, and solved the convex optimization problem using the subgradient method. The experimental results demonstrate the effectiveness of the proposed method: the accuracy was improved significantly by combining several multiple distance measures. However, a linear combination of the multiple distance measures may not be the best solution. So in the future, we will try to develop a non-linear version of the proposed method.

ACKNOWLEDGMENTS

We thank the reviewers for the constructive feedback. This work was supported by JSPS KAKENHI Grant Numbers 25540062 and 25870811.

REFERENCES

[1] J. C. Whisstock and A. M. Lesk, “Prediction of protein function from protein sequence and structure,” *Quarterly Reviews of Biophysics*, vol. 36, pp. 307–340, 2003.

[2] L. Holm and C. Sander, “Protein structure comparison by alignment of distance matrices,” *Mol. Biology*, vol. 233, pp. 123–138, 1993.

[3] I. N. Shindyalov and P. E. Bourne, “Protein structure alignment by incremental combinatorial extension (CE) of the optimal path,” *Protein Eng.*, vol. 11, pp. 739–747, 1998.

[4] A. Prlic, S. Bliven, P. W. Rose, W. F. Bluhm, C. Bizon, A.Godzik, and P. E. Bourne, “Precalculated protein structure alignments at the RCSB PDB website,” *Bioinformatics*, vol. 26, pp. 2983–2985, 2010.

[5] P. Røgen, “Evaluating protein structure descriptors and tuning gauss integral based descriptors,” *Physics Condensed Matter*, vol. 17, pp. 1523–1538, 2005.

[6] C. H. Suryanto, S. Jiang, and K. Fukui, “Protein structure similarity based on multi-view images generated from 3D molecular visualization,” *in ICPR*, pp. 3447–3451, 2012.

[7] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” *Machine Learning Research*, vol. 10, pp. 207–244, 2009.

[8] C. H. Q. Ding and I. Dubchak, “Multi-class protein fold recognition using support vector machines and neural networks,” *Bioinformatics*, vol. 17, pp. 349–358, 2001.

[9] T. Damoulas and M. A. Girolami, “Probabilistic multi-class multi-kernel learning: on protein fold recognition and remote homology detection,” *Bioinformatics*, vol. 10, pp. 1264–1270, 2008.

[10] K. Fukui and O. Yamaguchi, “Face recognition using multi-viewpoint patterns for robot vision,” *in 11th Int. Symposium of Robotics Research (ISRR’03)*, pp. 192–201, 2003.

[11] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, “SCOP: a structural classification of proteins database for the investigation of sequences and structures,” *Mol. Biology*, vol. 247, pp. 536–540, 1995.

[12] J.-M. M. Chandonia, G. Hon, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner, “The ASTRAL Compendium in 2004,” *Nucleic Acids Research*, vol. 32, pp. D189–D192, 2004.

[13] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, “Learning the kernel matrix with semidefinite programming,” *Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[14] H. Hino, N. Reyhani, and N. Murata, “Multiple kernel learning with gaussianity measures,” *Neural Computation*, vol. 24, no. 7, pp. 1853–1881, 2012.

[15] R. M. Hanson, “Jmol - a paradigm shift in crystallographic visualization,” *Journal of Applied Crystallography*, vol. 45, no. 5, pp. 1250–1260, 2010.

[16] N. Otsu and T. Kurita, “A new scheme for practical flexible and intelligent vision systems,” *in IAPR Workshop CV*, pp. 431–435, 1988.