Representations for Large-Scale Sequence Data Mining: A Tale of Two Vector Space Models

Vijay V. Raghavan[1], Ryan G. Benton[1], Tom Johnsten[2], and Ying Xie[3]

[1] Center for Advanced Computer Studies,
University of Louisiana at Lafayette, Louisiana, USA
{vijay,rbenton}@cacs.louisiana.edu
[2] School of Computing, University of South Alabama, Alabama, USA
tjohnsten@southalabama.edu
[3] Department of Computer Science, Kennesaw State University, Georgia, USA
yxei2@kennesaw.edu

Abstract. Analyzing and classifying sequence data based on structural similarities and differences is a mathematical problem of escalating relevance. Indeed, a primary challenge in designing machine learning algorithms to analyzing sequence data is the extraction and representation of significant features. This paper introduces a generalized sequence feature extraction model, referred to as the Generalized Multi-Layered Vector Spaces (GMLVS) model. Unlike most models that represent sequence data based on subsequences frequency, the GMLVS model represents a given sequence as a collection of features, where each individual feature captures the spatial relationships between two subsequences and can be mapped into a feature vector. The utility of this approach is demonstrated via two special cases of the GMLVS model, namely, Lossless Decomposition (LD) and the Multi-Layered Vector Spaces (MLVS). Experimental evaluation show the GMLVS inspired models generated feature vectors that, combined with basic machine learning techniques, are able to achieve high classification performance.

Keywords: Sequence Data, Classification, Feature Representation.

1 Introduction

Analyzing and classifying sequence data based on structural similarities and differences, no matter how subtle, is a mathematical problem of escalating relevance and surging importance in many different disciplines, particularly those in biology and information sciences. Characterizing patterns of all topologies at various levels of sophistication is a colossal problem lurking in the backdrop. One of the primary challenges in designing machine learning algorithms for the purpose of analyzing sequence data is the extraction and representation of significant features.

Most feature extraction methods are designed to represent sequence data based on the frequency of subsequences. For example, computational methods designed to analyze protein sequences typically represent a sequence as a set of features

D. Ciucci et al. (Eds.): RSFDGrC 2013, LNAI 8170, pp. 15–25, 2013.

corresponding to the frequency of subsequences of amino acids. It is easy to realize that such a simplistic approach fails to capture the complex relationships – be it temporal, spatial, local or global – in collections of sequence data. In response, we propose a generalized sequence feature extraction model, referred to as the Generalized Multi-Layered Vector Spaces (GMLVS) model, along with two special cases of the model referred to as the Lossless Decomposition (LD) model [1] and the Multi-Layered Vector Spaces (MLVS) model [2]. The GMLVS model represents a given sequence as a collection of features in which each individual feature can be mapped to a corresponding feature vector. The GMLVS model has the flexibility to generate diverse types of feature vectors. However, the size of the set of all possible features that can be generated is huge. This fact led to the development of the LD and MLVS models, which are able to generate different types of feature vectors using a well-defined subset of features represented through the GMLVS model. We believe the resulting feature vectors have the potential of penetrating into the micro structures embedded in sequences to provide an infrastructure for various forms of analysis at the local level, while concurrently addressing global patterns over those sequences.

The rest of this paper is organized as follows. Section 2 proposes the Generalized Multi-Layered Vector Spaces Model (GMLVS) for representing sequence data. Section 3 formally defines the Lossless Decomposition (LD) model and describes its application to the problem of pair-wise sequence alignment. Section 4 formally defines the Multi-Layered Vector Spaces (MLVS) model for representing sequence data and describes its application to the classification of biological sequences. Finally, Section 5 provides a discussion and summary of the work.

2 Generalized Multi-Layered Vector Spaces (GMLVS)

The proposed GMLVS model has several significant properties that collectively have the potential to discover interesting and novel patterns from sequence data. These properties include the ability to 1) discover both local and global patterns embedded in a sequence, 2) discover patterns defined in terms of the alphabet defined over a target collection of sequences, 3) reconstruct a sequence from its model representation, and 4) facilitate both descriptive and predictive data mining tasks. We now formally present the Generalized Multi-Layered Vector Spaces model for representing sequence data.

2.1 Model Formulations

A sequence S of finite length |S| defined over a finite alphabet β is viewed as a collection of generated subsequences, β_t^*, of length t where t = 1,..., |S|-1. Let β^* denote the set of all possible subsequences.

$$\bigcup_{t=1}^{|S|-1} \beta_t^* \tag{1}$$

The set of all possible pairs of subsequences (i, j), where i and j are elements of β^* is $\beta^* \times \beta^*$. Hence, the number of possible subsequences for a given t is equal to| β|^t

and the number of possible pairs of subsequences (*i*, *j*) for all t ($1 \leq t \leq |S|$) is equal to $(\beta^*)^2 * (k+1)$)). A *feature* is defined as a pair of subsequences *f* = (*i*, *j*), where *i* and *j* $\in \beta_t^*$, along with a specified step value *m* where $0 \leq m \leq k$. The parameter *m* stands for the number of spaces between the elements of a given feature. If *m*=1, then *f* represents a consecutive subsequence and if *m* > 1 then *f* is a subsequence with a gap, where the gap is filled by an arbitrary sequence of (*m – 1*) symbols (i.e. don't care). In the latter case, subsequences *i* and *j* are called, respectively, as leading and trailing subsequence. When *m* = 0, the leading subsequence is an element from β_t^* and the trailing element is a null symbol, which takes no space (i.e. size of trailing subsequence is zero). The upper bound for parameter k is (|R|-1), where R is the maximum admissible value of *m*. For instance, R is equal to |S| - 1, if the feature space is represented by all pairs of symbols (i, j), where *i* and *j* $\in \beta_1^*$. It should be noted that in order for a feature *f=(i, j)* to be valid, the sum of the length of subsequences *i* and *j* plus the value of *m* must be less than |S|. As a result, the number of possible features is less than or equal to the number of possible subsequences. Allowing multiple spaces between the elements of a feature generates a multitude of m-step pairs (families) $P_{0,}$ P_1,P_2,...,P_i,…,P_k, creating a multi-layered *k-clustering* C_k made up of sets $P_{m|(i,j)}$ where m=0,1,2,...,k. In general, the size of a cluster C_k is $|\beta|^t * (k+1)$, where t is equal to the sum of the length of the subsequences *i* and *j*. Using this notation, a sequence *S* can be represented by a set of features, which, in turn can be converted into a set of feature vectors. A feature is mapped into a corresponding feature vector only if it appears at least once in one of the sequences in a given collection of sequences. This fact can significantly reduce the size of the feature space. Assume *S* is <g, c, t, g, g, g, c, t, c, a, g, c, t, a, a, t, g, a, g, c>, *t*=1, and *m*=1. The feature (g,c), where g is the leading symbol and c is the trailing symbol, is present in the locations {1, 6, 11, 19}; this can be represented as a vector <1,6,11,10>. The resulting vector can be used to compare different sequences, or utilized to generate new representations. How this is done will be shown in the next section, which will present two specialized versions of the GMLVS model. The first model is the Lossless Decomposition Model, which corresponds to m=0 and t≥1. The second model is the Multi-Layered Vector Spaces model which corresponds to the case where m≥1 and t=2.

3 Lossless Decomposition Model

The Lossless Decomposition (LD) model creates a set of feature vectors **G** from a set of extracted features of the form *f* = (*i*, *NULL*), where $i \in \beta_t^*$ in which *m* = 0 such that **G** = *{<f_p> |f_p* is the starting position of the p^{th} instance of feature *f* in S*}*. The resulting feature vectors **G** represent a lossless decomposition since S can be reconstructed directly from **G**. The maximum number of LD feature vectors that can be generated from a sequence S is

$$\sum_{t=1}^{|S|-1} |\beta|^t \tag{2}$$

Example-1: Given the alphabet β = {a,c,g,t}, with |β| = 4 and the sequence defined over β S=[g, c, t, g, g, g, c, t, c, a, g, c, t, a, a, t, g, a, g, c]. The following GMLVS extracted features *a*, *gc*, and *gct* have corresponding LD generated feature vectors <10,14,15,18>, <1,6,11,19>, and <1,6,11>, respectively.

3.1 Pairwise Sequence Similarity

Measuring the degree of similarity between two sequences is an important task in several different domains. The LD model has been designed, in part, to facilitate the pairwise similarity measurement of sequences. By decomposing two sequences into a set of LD feature vectors, we are able to calculate the pairwise similarity of the sequences using parallel processes without sacrificing accuracy. For illustration purpose, we assume the feature vectors are based on GMLVS extracted features corresponding to the set β_2^* (m=0). In other words, we assume the generated feature vectors represent all possible consecutive subsequences of length two. Formally, given two sequences S1 and S2, the corresponding sets of feature vectors **G1** and **G2** are defined as follows:

$$\mathbf{G1} = \{<f_p> | f_p \text{ is the starting position of the } p^{th} \text{ instance of feature } f \text{ in S1}\}$$
$$\mathbf{G2} = \{<f_p> | f_p \text{ is the starting position of the } p^{th} \text{ instance of feature } f \text{ in S2}\}$$

Let a feature vector v1∈ **G1** be represented as $f_1, f_2, \ldots, f_i, \ldots, f_m$ where f_i is the i^{th} starting position of feature v1 in S1. Likewise, let a feature vector v2∈ **G2** be represented as $g_1, g_2, \ldots, g_i, \ldots, g_n$ where g_i is the i^{th} starting position of feature v2 in S2. We now define the distance between v1 and v2, which is denoted as *dist*(v1 , v2), to be a minimal cumulative distance calculated based on an optimal warping path between the feature vectors. The optimal warping path can be computed by the dynamic programming process, where the minimal cumulative distance $\Upsilon(f_i, g_j)$ is recursively defined as:

$$\Upsilon(f_i, g_j) = \mathrm{d}(f_i, g_j) + min(\Upsilon(f_i-1, g_j-1), \Upsilon(f_i-1, g_j), \Upsilon(f_i, g_j-1)) \qquad (3)$$

For example, assume the feature vector v1 is <0, 5, 9, 121, 130>, and the feature vector v2 is <4, 11, 100>. Then, by dynamic programming, the optimal alignment of these two vectors is illustrated in Figure 1. Then the distance between v1 and v2 can be calculated according to the optimal alignment as (4−0)+(5−4)+(11−9)+(121−100)+(130−100) = 58.

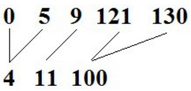


Fig. 1. Alignment between position sequences of two granules

Given the fact a feature vector represented in terms of the LD model is much shorter than the original sequence, the alignment between vectors by dynamic programming should be much more efficient than the alignment between the original sequences. The calculation of similarity between two sequences by pairwise alignment can be distributed across individual feature vectors. For this purpose, we define the distance between the sequence S1 and the sequence S2 as the aggregation of the distances between corresponding feature vectors. Let $v1_f$ and $v2_f$ represent the feature vector corresponding to feature $f = (i, NULL)$, where $i \in \beta_t^*$, in sequences S1 and S2, respectively:

$$\text{dist(S1, S2)} = \sum_{f = (i,NULL) \in \beta_t^*} dist(v1_f, v2_f) \tag{4}$$

From this definition, the calculation of the distance between two sequences can be distributed to $|\beta|^t$ calculations of distances between $|\beta|^t$ feature vectors.

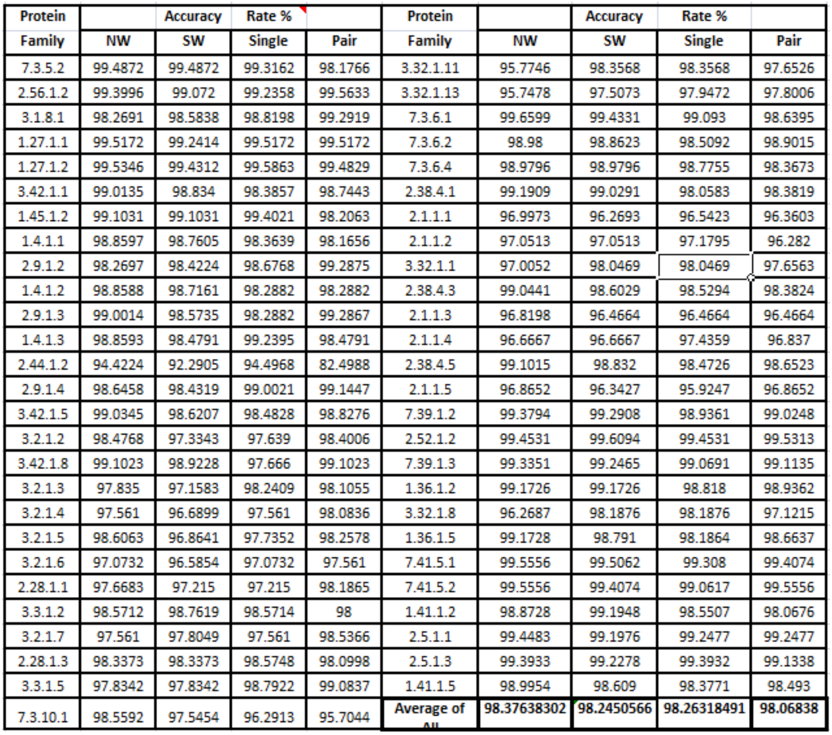
3.2 Experimental Investigation

We studied the performance of the proposed LD generated feature vectors in classifying 53 SCOP protein families. The data set of the 53 SCOP protein families can be downloaded from [11]. Each of the SCOP families contains a training data set and a testing data set as described in [3]. We simply used 1-nearest neighbor (1NN) approach to predict if a test sequence belongs to the given family or not. More specifically, for each test sequence, we evaluate its similarity with each training sequence, and then use the class label of the most similar training sequence as the label for this test sequence. The accuracy rate of the prediction for each family is reported.

We used the following approaches to evaluate similarity between two protein sequences: 1) the Needleman-Wunsch algorithm (NW) [4] ; 2) the Smith-Waterman algorithm (SW) [5]; 3) the proposed granular approach based on single amino acids (Single), and 4) the proposed granular approach based on pairs of amino acids (Pair). For NW and SW, we set the match reward to be 10 and mismatch penalty to be -8. No external scoring matrix is used for this preliminary experimental study. The classification results are summarized in Table-1.

As can be seen in Table-1, the proposed granular approach based on single amino acids reaches the same level of accuracy rate as the Needleman-Wunsch algorithm and the Smith-Waterman algorithm. In other words, the proposed granular approach is able to distribute the calculation of pairwise similarity to 20 parallel processes without sacrificing accuracy. The accuracy rate of the proposed granular approach based on pairs of amino acids is approximately 6% worse than the other three methods; however the calculation of similarity of two protein sequences under this setting can be distributed to 400 parallel processes, each of which deals with much smaller data. Therefore, this approach may be suitable for online analysis of very large scale protein sequence database, where the tradeoff between efficiency and accuracy is necessary.

Table 1. Preliminary experimental results



Protein		Accuracy	Rate %		Protein		Accuracy	Rate %	
Family	NW	SW	Single	Pair	Family	NW	SW	Single	Pair
7.3.5.2	99.4872	99.4872	99.3162	98.1766	3.32.1.11	95.7746	98.3568	98.3568	97.6526
2.56.1.2	99.3996	99.072	99.2358	99.5633	3.32.1.13	95.7478	97.5073	97.9472	97.8006
3.1.8.1	98.2691	98.5838	98.8198	99.2919	7.3.6.1	99.6599	99.4331	99.093	98.6395
1.27.1.1	99.5172	99.2414	99.5172	99.5172	7.3.6.2	98.98	98.8623	98.5092	98.9015
1.27.1.2	99.5346	99.4312	99.5863	99.4829	7.3.6.4	98.9796	98.9796	98.7755	98.3673
3.42.1.1	99.0135	98.834	98.3857	98.7443	2.38.4.1	99.1909	99.0291	98.0583	98.3819
1.45.1.2	99.1031	99.1031	99.4021	98.2063	2.1.1.1	96.9973	96.2693	96.5423	96.3603
1.4.1.1	98.8597	98.7605	98.3639	98.1656	2.1.1.2	97.0513	97.0513	97.1795	96.282
2.9.1.2	98.2697	98.4224	98.6768	99.2875	3.32.1.1	97.0052	98.0469	98.0469	97.6563
1.4.1.2	98.8588	98.7161	98.2882	98.2882	2.38.4.3	99.0441	98.6029	98.5294	98.3824
2.9.1.3	99.0014	98.5735	98.2882	99.2867	2.1.1.3	96.8198	96.4664	96.4664	96.4664
1.4.1.3	98.8593	98.4791	99.2395	98.4791	2.1.1.4	96.6667	96.6667	97.4359	96.837
2.44.1.2	94.4224	92.2905	94.4968	82.4988	2.38.4.5	99.1015	98.832	98.4726	98.6523
2.9.1.4	98.6458	98.4319	99.0021	99.1447	2.1.1.5	96.8652	96.3427	95.9247	96.8652
3.42.1.5	99.0345	98.6207	98.4828	98.8276	7.39.1.2	99.3794	99.2908	98.9361	99.0248
3.2.1.2	98.4768	97.3343	97.639	98.4006	2.52.1.2	99.4531	99.6094	99.4531	99.5313
3.42.1.8	99.1023	98.9228	97.666	99.1023	7.39.1.3	99.3351	99.2465	99.0691	99.1135
3.2.1.3	97.835	97.1583	98.2409	98.1055	1.36.1.2	99.1726	99.1726	98.818	98.9362
3.2.1.4	97.561	96.6899	97.561	98.0836	3.32.1.8	96.2687	98.1876	98.1876	97.1215
3.2.1.5	98.6063	96.8641	97.7352	98.2578	1.36.1.5	99.1728	98.791	98.1864	98.6637
3.2.1.6	97.0732	96.5854	97.0732	97.561	7.41.5.1	99.5556	99.5062	99.308	99.4074
2.28.1.1	97.6683	97.215	97.215	98.1865	7.41.5.2	99.5556	99.4074	99.0617	99.5556
3.3.1.2	98.5712	98.7619	98.5714	98	1.41.1.2	98.8728	99.1948	98.5507	98.0676
3.2.1.7	97.561	97.8049	97.561	98.5366	2.5.1.1	99.4483	99.1976	99.2477	99.2477
2.28.1.3	98.3373	98.3373	98.5748	98.0998	2.5.1.3	99.3933	99.2278	99.3932	99.1338
3.3.1.5	97.8342	97.8342	98.7922	99.0837	1.41.1.5	98.9954	98.609	98.3771	98.493
7.3.10.1	98.5592	97.5454	96.2913	95.7044	**Average of All**	**98.37638302**	**98.2450566**	**98.26318491**	**98.06838**

4 Multi-Layered Vector Spaces Model

The Multi-Layered Vector Spaces Model (MLVS) creates a set of feature vectors **G** based on GMLVS features of the form (i,j), where *i* and *j* ∈ β_1^*. The total number of feature vectors that can be generated from an alphabet β is $|\beta|^2$. In this specialized case, a sequence *S* is viewed to have a multi-layered structure made up of a set of *m*-step ordered pairs (features) (i,j), where *i* and *j* ∈ β_1^*, denoted by $P_{m|(i,j)}$, where $1 \leq m \leq$ k. Ordered pairs made up of consecutive elements of the sequence are said to form the family of 1-step (one-step) pairs, $P_{1|(i,j)}$. The concept of a multi-layered k-clustering C_k, as defined in the context of the GMLVS model, also applies to the MLVS model. Thus, the MLVS model views a sequence S as the as the union of all ordered pairs (i,j), where *i* and *j* ∈ β_1^* at k distinct layers. The following example demonstrates how the said structures are built.

Example-2: Given the alphabet β = {a,c,g,t}, with |β|=4, $|\beta|^2$ =16, and the sequence *S* = [g, c, t, g, g, g, c, t, c, a, g, c, t, a, a, t, g, a, g, c]. The following are sample *m*-step pairs (β_1^*): 1-step ordered pairs for (g,c) are located at step locations [1,2], [6,7], [11,12], and [19,20]; 1-step ordered pairs for (g,g) are located at step locations [4,5], and [5,6]; 2-step ordered pairs for (g,t) are located at step locations [1,3], [6,8], and [11,13]; 4-step ordered pairs for (c,g) are located at step locations [2,6], and [7,11].

4.1 Feature Vector Creation

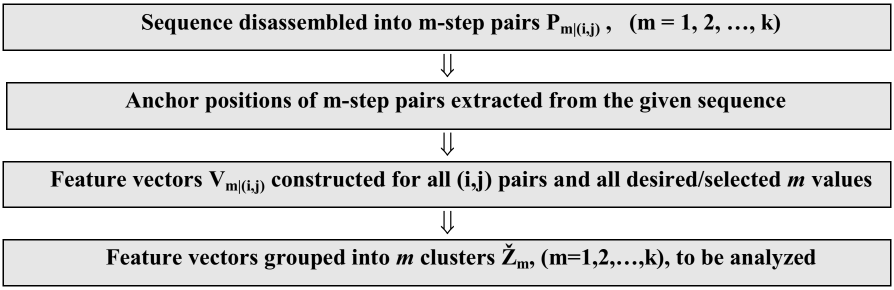
For a selected value of *m* and a given GMLVS extracted feature *f* = (i,j) (i, j∈β_1^*), the sequence of anchor positions is taken as forming the scalar components of an n-dimensional feature vector $\mathbf{V}_{\mathbf{m|(i,j)}}$ associated with the feature (i,j). The union of such vectors for all features (for a given *m*) forms a vector cluster $\check{\mathbf{Z}}_{\mathbf{m}}$ at step size *m*, providing a single-step representation for the sequence.

$$\check{Z}_m = \bigcup_{(i,j)} V_{m|(i,j)} \tag{5}$$

The union of vector clusters $\check{\mathbf{Z}}_{\mathbf{m}}$ provides a multi-layered feature vector space $\check{\mathbf{Z}}_{\mathbf{k}}$**,** one layer for each value of *m*, for the original sequence.

$$\check{Z}_k = \bigcup_m \bigcup_{(i,j)} V_{m|(i,j)} \tag{6}$$

Feature vectors for each *m*-step feature can be structured in at least two different ways. One approach is to simply record the step (spatial index) locations of anchor positions as Boolean values (1, 0). This approach is suitable for collections of equal length sequences. An alternative approach is to partition a sequence into *n* equal segments and record the number of anchor positions that fall into each segment. The number of segments *n* will determine the dimension of the vectors thus formed. The size of *n* can be adjusted to meet restrictions or expectations on resolution and accuracy. This approach has the advantage of mapping sequences of unequal length into fixed length feature vectors. For a given *m*, the construction scheme for $\mathbf{V}_{\mathbf{m|(i,j)}}$ can be implemented in two different ways: a vector can be constructed for each feature in the sequence to generate a vector cluster over the whole sequence, or feature vectors in the cluster are concatenated into a single vector to represent the entire sequence. The steps involved in the feature mapping process are illustrated in Fig. 2. As is the case with LD feature vectors, MLVS feature vectors can also be analyzed in a distributed manner. In particular, MLVS feature vectors can be processed in parallel based on either specific sets of ordered pairs (i, j) and / or range of step sizes (m).



Sequence disassembled into m-step pairs $\mathbf{P}_{\mathbf{m|(i,j)}}$,   (m = 1, 2, …, k)

⇓

Anchor positions of m-step pairs extracted from the given sequence

⇓

Feature vectors $\mathbf{V}_{\mathbf{m|(i,j)}}$ constructed for all (i,j) pairs and all desired/selected *m* values

⇓

Feature vectors grouped into *m* clusters $\check{\mathbf{Z}}_{\mathbf{m}}$, (m=1,2,…,k), to be analyzed

Fig. 2. Proposed feature mapping process

Example-2: Using the same alphabet and sequence as used in the previous examples, the following are sample feature vectors for a select group of m-step MLVS features:

Anchor positions of 1-step feature (g,c) are located at step (index) locations [1,6,11,19]; vector $\mathbf{V}_{1l(gc)}$, is represented by the Boolean feature vector <1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0> if step locations for the anchors are used directly as vector components. If we instead partition the sequence into 4 equal segments (n = 4), the vector $\mathbf{V}_{1l(gc)}$, is represented by the 4D feature vector <1,1,1,1> with vector components representing the number of anchor elements in each segment; anchor positions of the 1-step feature (g,g) are located at step (index) locations [4,5]; vector $\mathbf{V}_{1l(gg)}$ is represented by the Boolean feature vector <0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0> or by the 4D vector <2,0,0,0>; anchor positions of 2-step feature (g,t) are located at step (index) locations [1,6,11]; vector $\mathbf{V}_{2l(gt)}$ is represented by the Boolean vector <1,0,0,0,0,1 ,0,0,0,0,1,0,0,0,0,0,0, 0,0,0> or by the 4D vector <1,1,1,0>.

4.2 Experimental Investigation

Experiments were conducted to determine the potential usefulness of the MLVS generated feature vectors in classifying biological sequences. Specific objectives included: investigating the accuracy of classifiers constructed from various n-dimensional feature vectors $\mathbf{V}_{ml(i,j)}$; and, the accuracy of ensemble classifiers constructed from individual vector clusters $\check{\mathbf{Z}}_{\mathbf{m}}$. The results obtained from these classifiers were compared with results obtained from the (k,m)-mismatch kernel method [6,7].

The biological sequences utilized in the experiments corresponded to the classification of the 3PGK-DNA sequences, Eukaryota vs. Euglenozoa [8]. There were a total of forty-three instances belonging to the class Eukaryota and forty-four instances belonging to the class Euglenozoa. The alphabet β consisted of the elements {a,c,g,t}. Each instance was mapped into the following vector clusters $\check{\mathbf{Z}}_1$, $\check{\mathbf{Z}}_2$, $\check{\mathbf{Z}}_3$, and $\check{\mathbf{Z}}_{10}$. For the experiments, we set n=100; that is, we segmented each $\mathbf{V}_{ml(i,j)}$ into 100 equal segments. In addition, we arbitrarily selected the step sizes m=1,2,3, and 10. We utilized the decision tree classifier C4.5 [9] as implemented in the Weka data mining application [10]. The performance of the decision trees was evaluated using the hold-out method in which the feature vectors, $\mathbf{V}_{ml(i,j)}$, for a given GMLVS feature f = (i, j) (i, j$\in$ β_1^*), were randomly divided into five pairs of training and test sets. The reported performance is the average accuracy over five runs.

The results of the experiments are shown in Tables 2 and 3. Table-2 shows the accuracy of the decision trees constructed from the feature vectors for each ordered pair feature. For instance, the decision tree constructed from the feature vectors corresponding to the ordered pair (a,a) has an estimated predicted accuracy of 75%, 82%, 75%, and 69% with respect to step sizes 1, 2, 3, and10, respectively. The results show for the selected step sizes, the decision trees are performing better than random guessing but not at a desired level. A significant improvement in performance is obtained from the use of ensemble (multiple) classifiers constructed from decision trees belonging to a single vector cluster. Table-3 shows the accuracy values obtained by combining multiple decision trees at step sizes 1, 2, 3, and 10. The grouping of classifiers into ensembles was based on the accuracy of individual decision trees constructed from single ordered pairs. Specifically, for a given step size, the decision trees were selected based on accuracy and the *r* most accurate decision trees were combined to form an ensemble of size *r*. The decision trees of a given ensemble were combined using *un-weighted* majority voting. Several of the constructed ensemble classifiers shown in Table-3 have a high degree of accuracy, and in particular the ensemble classifier consisting of fifteen decision trees at step size m=1 (15:96) has a 96% level of accuracy.

Table 2. Decision tree accuracy values for selected feature vectors

$V_{m\|(i,j)}$	**m=1**	**m=2**	**m=3**	**m=10**
(a,a)	75	82	75	69
(a,c)	77	63	69	64
(a,g)	75	89	83	75
(a,t)	77	78	82	71
(c,a)	69	68	71	67
(c,c)	76	75	82	87
(c,g)	64	78	74	67
(c,t)	76	68	77	67
(g,a)	70	78	74	72
(g,c)	75	67	82	82
(g,g)	70	66	84	85
(g,t)	76	64	69	76
(t,a)	66	68	72	67
(t,c)	87	75	74	70
(t,g)	72	70	67	67
(t,t)	76	61	75	72
Average	74	72	76	72

Table 3. Ensemble decision tree accuracy values for selected vector clusters

m	**# Classifiers : Accuracy (%)**
1	3:90; 5:93; 7:93; 9:94; 11:92; 13:92; **15:96**
2	3:90; 5:87; 7:87; 9:90; 11:87; 13:83; 15:79
3	3:87; 5:92; 7:91; 9:92; 11:93; 13:94; 15:94
10	3:92; 5:90; 7:92; 9:92; 11:90; 13:89; 15:87

Table 4. (k,m)-mismatchmethod accuracy values

K	**m = 0 (%)**	**m=1 (%)**
4	90	89
5	93	88
6	93	90
7	91	93
8	91	93
9	90	91
10	86	90

To evaluate the results recorded in Tables-2 and -3, we repeated the experiments using the (k,m)-mismatch kernel method. Specifically, the five pairs of training and test sets were evaluated using the (k,m)-mismatch method as implemented by the authors of [6,7]. Table-4 shows the classification accuracy results, averaged over the five runs, for contiguous subsequences of length k = 4, 5, …, 10 and zero or one mismatches (m). The maximum achieved accuracy was 93%, which is less than the 96% accuracy value obtained through the use of the proposed multi-layer vector space model. In addition, the comprehensibility of a decision tree classifier is, in general, much greater as compared to SVM classifiers (i.e. (k,m)-mismatch method). This difference is significant if one wishes to obtain a deep characterization of a collection of biological sequences.

5 Discussion and Summary

It is anticipated that the transparent quality, simplicity and therefore the interpretation of the feature extraction models discussed in this paper will shed light into the inner workings of the system being studied. The Generalized Multi-Layered Vector Spaces (GMLVS) model allows an investigator to map a collection of sequences into a very large space of feature vectors for the purpose of analyzing and classifying data. The generated feature vectors can be logically partitioned along multiple dimensions based on sets of specific GMLVS features (i, j) (i and $j \in \beta_t^*$) and/or specific step values m ($0 \leq m \leq k$). We believe a large feature vector space whose vectors can be partitioned into semantically related groups will provide a user-friendly mathematical habitat in which an investigator can discover the intrinsic elements of the system being studied such as the plausibility of interactions among micro patterns and causal connections embedded in a sequence. More generally, an investigator has the opportunity to discover relationships among various groups of feature vectors and to discover characteristics of the feature space as the step values (m) are increased to their limit.

We have also developed two related sequential data models, referred to as the Lossless Decomposition (LD) model and the Multi-Layered Vector Spaces (MLVS) model. These two models are able to generate different types of feature vectors using a well-defined subset of features represented through the GMLVS model. Preliminary experimental results reported on in this paper indicate both the LD and MLVS models have the capability to identify important relationships within individual sequences.

In the future, we plan to explore the utility of GMLVS (and specialized cases) in a variety of ways. One area of study is to explore the applicability of the GMLVS for signal peptide prediction; that is, to identify sections of amino acids that used to direct nascent, or newly formed, proteins to their correct locations. Moreover, we believe the GLMVS format (or a derivative) could be used to create human-interpretable rules; this is something currently lacking in of the current signal peptide detection techniques. Second, we also wish to explore the applicability of the MLVS model, combined with association mining, to detect potential mutations and frequent co-occurrences of mutations within cancer cells. Third, we are interested in exploring how to incorporate external scoring matrices into the LD model, along with developing adaptive search methods to exploit the LD representation. Finally, the MLVS and LD methods represent only two special cases of the GMLVS model; developing complementary special case models may yield additional advantages and insights.

References

1. Xie, Y., Fisher, J., Raghavan, V.V., Johnsten, T., Akkoc, C.: Granular approach for protein sequence analysis. In: Yao, J., Yang, Y., Słowiński, R., Greco, S., Li, H., Mitra, S., Polkowski, L. (eds.) RSCTC 2012. LNCS, vol. 7413, pp. 414–421. Springer, Heidelberg (2012)
2. Akkoç, C., Johnsten, T., Benton, R.: Multi-layered vector spaces for classifying and analyzing biological sequences. In: BICoB, pp. 160–166 (2011)
3. Liao, L., Noble, S.: Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. Journal of Computational Biology, 857–868 (2003)
4. Needleman, B., Wunsch, D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology, 443–453 (1970)
5. Smith, F., Waterman, S.: Identification of common molecular subsequences. Journal of Molecular Biology, 195–197 (1981)
6. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: A string kernel for SVM protein classification. In: Pacific Symposium on Biocomputing, pp. 564–575 (2002)
7. Leslie, C., Eskin, E., Weston, J., Noble, W.S.: Mismatch string kernels for SVM protein classification. In: Neural Information Processing Systems, pp. 1441–1448 (2003)
8. Sonego, P., Pacurar, M., Dhir, S., Kertesz-Farkas, A., Kocsor, A., Gaspari, Z., Leunissen, J., Pongor, S.: A protein classification benchmark collection for machine learning, D232-D236 (2007)
9. Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufmann (1993)
10. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDDD Explorations, 10–18 (2009)
11. Supplementary data (from paper [3]), `http://noble.gs.washington.edu/proj/svm-pairwise/`