Predicting Strengths of Protein-protein Interactions Through Online Regression Algorithms

M. Hayashida¹, M. Kamada², and H. Koyano³
¹Department of Electrical Engineering and Computer Science, National Institute of Technology, Matsue College, Matsue, Shimane, Japan
²Graduate School of Medicine, Kyoto University, Kyoto, Kyoto, Japan
³Quantitative Biology Center, Riken, Kobe, Hyogo, Japan

Abstract—Protein-protein interactions take various important roles in living cells. Interaction strengths provide useful knowledge to understand complicated cellular networks, and several prediction methods have been developed. In our previous study, we proposed new feature space mappings based on protein domains, and employed support vector regression and relevance vector machine. The combination of the mapping and the supervised regression method outperformed the existing methods.

In this study, we examine online learning algorithms, the regression passive-aggressive (PA) and adaptive regularization of weights for regression with covariance reset (ARCOR) algorithms. Furthermore, we introduce nonlinear transformation to a linear regression formula and examine ensemble learning. For evaluation, we performed three-fold cross-validation computational experiments. The root mean square error (RMSE) by our proposed method was smaller than those by the existing methods. It implies that our method combining online regression algorithms with nonlinear transformation and sequences of domain regions is useful.

Keywords: protein-protein interaction strength, regression PA, ARCOR

1. Introduction

For understanding dynamic cellular systems, it is necessary to analyze strengths of protein-protein interactions (PPIs). If a protein strongly interacts with another protein, the proteins form a stable complex, and the complex maintains its function. In contrast, if a protein weakly interacts, the protein temporarily changes the state of its complex. In addition, weak interactions are known to be involved with enzyme regulation and signal transduction [1], [2].

The strength of a PPI is often represented by the dissociation constant that is the ratio of the rate constant of the dissociation reaction to that of association reaction. Physicochemical methods for measuring strengths of PPIs have been developed by utilizing solution nuclear magnetic resonance (NMR) [3], [4]. These methods, however, take long time to exhaustively measure whole pairs of proteins in an organism. Hence, several computational methods for predicting strengths of PPIs have been developed. Deng et al. proposed a probabilistic model representing protein-protein interactions using domain-domain interactions [5]. LPNM was developed as a linear programming-based method based on the probabilistic model, and minimizes the sum of errors between predicted and actual strengths [6]. ASNM is a faster method developed by modifying the association method [7], the prediction accuracy was comparable to that of LPNM [8]. Chen et al. proposed association probabilistic method (APM) by improving ASNM in consideration of the probabilistic model [9]. In our previous study [10], we proposed new feature space mappings from protein pairs using domain information such as amino acid sequences and domain compositions in a protein, and combined them with machine learning methods, support vector regression (SVR) [11] and relevance vector machine (RVM) [12]. The results of the cross-validation experiments showed that the root mean square error (RMSE) by our previous method was smaller than those by the existing methods, APM, ASNM, and LPNM.

Online linear classifiers are fast, reduce memory usage, and have been applied to analyses of big data. In online learning, a learning algorithm takes instances in a sequential manner, and outputs a new model after each observation. Crammer et al. proposed a passive-aggressive (PA) algorithm and two alternative modifications for coping with noise [13]. The PA algorithms update weights as little as possible such that the current training instance is correctly classified. Adaptive regularization of weights (AROW) is another online algorithm, can deal with non-separable data, and achieves state-of-the-art performance [14]. These classification algorithms can be extended to linear regression problems. The ARCOR algorithm is a modification of AROW for regression [15]. We examine the regression PA and ARCOR algorithms for improving prediction accuracy of strengths of PPIs. Furthermore, we introduce nonlinear transformation to a linear regression formula and examine ensemble learning. For evaluation, we performed three-fold cross-validation computational experiments. The RMSE by our proposed method was smaller than those by the existing methods.
2. Methods

In this section, we briefly review our previously proposed feature space mapping, called SPD [10], and online learning algorithms, the regression passive-aggressive (PA) algorithm [13] and adaptive regularization of weights for regression with covariance reset (ARCOR) [15]. We explain the proposed method using nonlinear transformation and ensemble learning.

2.1 Feature space mapping by restriction of spectrum kernel to domain regions (SPD)

The feature space mapping of SPD is obtained by restricting the application of the spectrum kernel [16] to domain regions (see Fig. 1).

![Fig. 1: Illustration on the feature space mapping of restriction of spectrum kernel to domain regions (SPD). A protein sequence contains two domains D1 and D2. Spectrum kernel is applied to the concatenated amino acid string in domain regions.](image)

Let $A$ and $A^k$ be an alphabet and the set of all strings with length $k$ consisting of letters in $A$, respectively. Let $\phi_s(P_i)$ be the number of occurrences of string $s$ in the sequence restricted to the domain regions in protein $P_i$. The feature space mapping $f_{ij}$ of SPD for a pair of proteins $P_i$ and $P_j$ is defined by

$$f_{ij}^{(n)} = \phi_{s_n}(P_i), \quad f_{ij}^{(|A|^k+n)} = \phi_{s_n}(P_j)$$

for all $s_n \in A^k$, where $|A|$ indicates the number of elements in $A$.

2.2 Passive-aggressive (PA) algorithm

Let $(x_t, y_t)$ be $t$-th example, where $x_t$ and $y_t$ mean a feature vector obtained from a pair of proteins $P_i$ and $P_j$ and the strength that $P_i$ and $P_j$ interact with each other for our purpose. Then, online linear learners find a weight vector $w$ such that $w \cdot x_t$ is close to $y_t$. Let $w_t$ be the weight vector on round $t$. $w_t$ is initialized to $(0, \ldots, 0)$. In the regression passive-aggressive algorithm, the new weight $w_{t+1}$ is determined depending on the $t$-th example $(x_t, y_t)$ such that it minimizes $||w_{t+1} - w_t||^2$ under the constraint $l(w_{t+1}; (x_t, y_t)) = \max \{0, y_t - w_{t+1} \cdot x_t - \epsilon\} = 0$ for some $\epsilon > 0$. If $l(w_t; (x_t, y_t)) = l_t = 0$, then $w_{t+1}$ is determined to be $w_t$ to minimize $||w_{t+1} - w_t||^2$. Otherwise, $w_{t+1}$ is aggressively determined to satisfy $l(w_{t+1}; (x_t, y_t)) = 0$. The Lagrange function is defined as $L(w_{t+1}, \tau) = ||w_{t+1} - w_t||^2 + \tau(y_t - w_{t+1} \cdot x_t - \epsilon)$ using a Lagrange multiplier $\tau > 0$. By solving $\partial L(w_{t+1}; \tau)/\partial w_{t+1} = 0$ and $\partial L(w_{t+1}; \tau)/\partial \tau = 0$, we have the update formula,

$$w_{t+1} = w_t + \text{sign}(y_t - w_t \cdot x_t)\tau_t x_t,$$

(2)

where $\tau_t = \frac{l_t}{||x_t||^2}$, and $\text{sign}(z) = -1$ if $z < 0$, $0$ if $z = 0$, $1$ if $z > 0$.

By allowing $l(w_{t+1}; (x_t, y_t)) \neq 0$ and introducing a constant $C > 0$, called aggressiveness parameter, two variants were proposed. One is to minimize $||w_{t+1} - w_t||^2 + C\xi^2$ under the constraints $l(w_{t+1}; (x_t, y_t))) \leq \xi$ and $\xi \geq 0$. Another is to minimize $||w_{t+1} - w_t||^2 + C\xi^2$ under the constraint $l(w_{t+1}; (x_t, y_t))) \leq \xi$. Then, the corresponding update formulas are given by replacing $\tau_t$ in Eq. (2) with $\min\{C, \frac{l_t}{||x_t||^2}\}$ for PA-I, and with $\frac{l_t}{||x_t||^2}$ for PA-II, respectively.

2.3 Adaptive regularization of weights for regression with covariance reset (ARCOR)

In the ARCOR algorithm, a Gaussian distribution $N(w_t, \Sigma_t)$ with a mean vector $w_t$ and a covariance matrix $\Sigma_t$ is maintained. First, $w_t$ and $\Sigma_t$ are initialized as $(0, \ldots, 0)$ and the identity matrix, respectively. Given the $t$-th example $(x_t, y_t)$, the new weight $w_{t+1}$ and covariance matrix $\Sigma_{t+1}$ are determined such that they minimize $D_KL(N(w_{t+1}, \Sigma_{t+1})||N(w_t, \Sigma_t)) + \frac{1}{2}||y_t - w_{t+1} \cdot x_t||^2 + \frac{r}{2}x_t^\top \Sigma_t^{-1}x_t$, where $D_KL$ indicates the Kullback-Leibler divergence, $r$ is a positive constant, and $x^\top$ indicates the transpose of $x$. The first term of the objective function requires not much to change the parameters $w_{t+1}$ and $\Sigma_{t+1}$ from $w_t$ and $\Sigma_t$. The second term requires to minimize the squared error for the current example, and the last term requires to reduce the variance for the parameter $w_{t+1}$. Then, the update formula is derived from the minimization problem as

$$w_{t+1} = w_t + \frac{y_t - w_t \cdot x_t}{x_t^\top \Sigma_t x_t + r} x_t \Sigma_t x_t^{-1},$$

and

$$\Sigma_{t+1} = \Sigma_t^{-1} + \frac{1}{r}x_t x_t^\top.$$ 

2.4 Nonlinear transformation and ensemble learning

In linear regression algorithms including PA-I, PA-II, and ARCOR, the weight $w$ of $y = w \cdot x$ is often estimated such that the value of some cost function for all examples $(x_t, y_t)$ is close to zero. However, actual values $y_t$ do not always follow such a linear formula, $w \cdot x$. Especially, for our purpose, in a feature vector $f_{ij}$ for a pair of proteins $P_i$ and $P_j$, a part of the feature vector involved with $P_i$ can be related to those with $P_j$. Hence, it is reasonable that quadratic or more terms of $x_i x_j$ and $x_i x_j x_k$ are included in the predictive function as well as linear terms. Thus, we introduce a nonlinear transformation function $g$ as $y = g(w \cdot x)$.

In addition, we examine a simple ensemble learning using the weights $w(m)$ $(m = 1, \ldots, M)$ obtained by $M$
regression learners, that is, we predict the interaction strength between proteins $P_i$ and $P_j$ as $\frac{1}{M} \sum_{m=1}^{M} g(w^{(m)} \cdot f_{ij})$.

3. Results

In the previous studies [6], [9], interaction sequence tags (ISTs) obtained by high-throughput yeast two-hybrid (Y2H) experiments [17] were used as strengths of protein-protein interactions. It, however, is known that Y2H includes a high false-positive rate mainly caused by non-specific interactions [18]. Hence, we used more reliable WI-PHI protein-protein interaction dataset [19] with 50000 protein pairs, in which PPIs are weighted by some reliability calculated in a statistical manner from several biological experiments, as interaction strengths under the difficulty of measuring strengths for many protein pairs. In our preliminary study [20], it was shown also for the IST dataset that our previous method 'SVR+SPD' outperformed the best existing method PPI2. We used the value dividing the weight of PPI by the maximum weight as the strength. We calculated the SPD feature vector using amino acid sequences and domain compositions of proteins stored in UniProt database [21]. Among 50000 protein pairs, we used 1387 pairs that contain complete domain sequences, which include 758 proteins and 327 domains. We added 100 randomly selected protein pairs as PPIs with strength 0. Totally 1487 protein pairs are the same as those in the previous experiment. The alphabet $A$ consists of 20 amino acids and ambiguous one. For evaluating prediction accuracy, we performed threefold cross-validation experiments, and calculated RMSE defined by $\sqrt{\frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2}$ for $N$ actual values $y_t$ and predicted values $\hat{y}_t$. We used the regression PA and ARCOR algorithms implemented in Hivemall with default parameters [22], which run under Hadoop scalable and distributed computing environment [23].

Table 1 shows the results of RMSE for test data by our proposed method using SPD with $k = 4$, our previous methods, and APM. 'SPD+PA-I' indicates the combination of SPD and PA-I. 'ARCOR-H' indicates the algorithm obtained by replacing the loss function of ARCOR with the hinge loss function as $l(w; (x_t, y_t)) = \max(0, |y_t - w \cdot x_t| - \epsilon)$. 'ARCOR-Hσ', 'PA-1σ' and 'PA-IIσ' indicate the algorithms obtained by replacing $\epsilon$ of the hinge loss function with $\sigma_t$, respectively, where $\sigma_t$ is the standard deviation of $y_1, \ldots, y_t$.

Table 1 shows the results of RMSE for test data by our proposed method using SPD with $k = 4$, our previous methods, and APM. 'SPD+PA-I' indicates the combination of SPD and PA-I. 'ARCOR-H' indicates the algorithm obtained by replacing the loss function of ARCOR with the hinge loss function as $l(w; (x_t, y_t)) = \max(0, |y_t - w \cdot x_t| - \epsilon)$. 'ARCOR-Hσ', 'PA-1σ' and 'PA-IIσ' indicate the algorithms obtained by replacing $\epsilon$ of the hinge loss function with $\sigma_t$, respectively, where $\sigma_t$ is the standard deviation of $y_1, \ldots, y_t$.

3. Results

In the previous studies [6], [9], interaction sequence tags (ISTs) obtained by high-throughput yeast two-hybrid (Y2H) experiments [17] were used as strengths of protein-protein interactions. It, however, is known that Y2H includes a high false-positive rate mainly caused by non-specific interactions [18]. Hence, we used more reliable WI-PHI protein-protein interaction dataset [19] with 50000 protein pairs, in which PPIs are weighted by some reliability calculated in a statistical manner from several biological experiments, as interaction strengths under the difficulty of measuring strengths for many protein pairs. In our preliminary study [20], it was shown also for the IST dataset that our previous method 'SVR+SPD' outperformed the best existing method PPI2. We used the value dividing the weight of PPI by the maximum weight as the strength. We calculated the SPD feature vector using amino acid sequences and domain compositions of proteins stored in UniProt database [21]. Among 50000 protein pairs, we used 1387 pairs that contain complete domain sequences, which include 758 proteins and 327 domains. We added 100 randomly selected protein pairs as PPIs with strength 0. Totally 1487 protein pairs are the same as those in the previous experiment. The alphabet $A$ consists of 20 amino acids and ambiguous one. For evaluating prediction accuracy, we performed threefold cross-validation experiments, and calculated RMSE defined by $\sqrt{\frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2}$ for $N$ actual values $y_t$ and predicted values $\hat{y}_t$. We used the regression PA and ARCOR algorithms implemented in Hivemall with default parameters [22], which run under Hadoop scalable and distributed computing environment [23].

Table 1 shows the results of RMSE for test data by our proposed method using SPD with $k = 4$, our previous methods, and APM. 'SPD+PA-I' indicates the combination of SPD and PA-I. 'ARCOR-H' indicates the algorithm obtained by replacing the loss function of ARCOR with the hinge loss function as $l(w; (x_t, y_t)) = \max(0, |y_t - w \cdot x_t| - \epsilon)$. 'ARCOR-Hσ', 'PA-1σ' and 'PA-IIσ' indicate the algorithms obtained by replacing $\epsilon$ of the hinge loss function with $\sigma_t$, respectively, where $\sigma_t$ is the standard deviation of $y_1, \ldots, y_t$.

For the case of $g(y) = y$, the RMSE by our proposed method was larger than that by 'RVM+SPD'. For the case of $g(y) = y^2$, the RMSE by the ensemble method with a bias term was the smallest in the experiments. It implies that the nonlinear transformation is useful for predicting the strength of PPIs.

4. Conclusion

We proposed a simple ensemble learning with nonlinear transformation for predicting strengths of protein-protein interactions. For evaluation of our proposed method, we performed threefold cross-validation experiments. The RMSE by our proposed method was smaller than those by the existing methods, APM, and our previously developed methods. It implies that our method combining online regression algorithms with nonlinear transformation and sequences of domain regions is useful although further evaluation for more data directly related to the dissociation constant is required. In this study, we used online learners, the regression PA and ARCOR algorithms for linear regression. These algorithms, however, can also handle kernel functions. For instance, the PA-I algorithm was combined with the polynomial kernel and applied to dependency parsing and hyponymy-relation extraction [24]. For further improvement of prediction accuracy, we would like to employ kernel functions combined with the regression PA and ARCOR algorithms.

Acknowledgements

This work was partially supported by Grants-in-Aid #16K00392, and #16KT0020 from MEXT, Japan.

References

Table 1: Result of root mean square error (RMSE) for test data by our proposed method using SPD with \( k = 4 \), and related methods.

<table>
<thead>
<tr>
<th>method</th>
<th>( g(y) = y ) without bias</th>
<th>( g(y) = y ) with bias</th>
<th>( y^{\frac{1}{2}} ) without bias</th>
<th>( y^{\frac{1}{2}} ) with bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPD+PA-I</td>
<td>0.1539</td>
<td>0.1451</td>
<td>0.1288</td>
<td>0.1304</td>
</tr>
<tr>
<td>SPD+PA-I( \sigma )</td>
<td>0.1539</td>
<td>0.1451</td>
<td>0.1288</td>
<td>0.1304</td>
</tr>
<tr>
<td>SPD+PA-II</td>
<td>0.1547</td>
<td>0.1457</td>
<td>0.1274</td>
<td>0.1215</td>
</tr>
<tr>
<td>SPD+PA-II( \sigma )</td>
<td>0.1548</td>
<td>0.1458</td>
<td>0.1275</td>
<td>0.1215</td>
</tr>
<tr>
<td>SPD+ARCOR</td>
<td>0.1558</td>
<td>0.1476</td>
<td>0.1280</td>
<td>0.1221</td>
</tr>
<tr>
<td>SPD+ARCOR-H( \sigma )</td>
<td>0.1538</td>
<td>0.1454</td>
<td>0.1288</td>
<td>0.1303</td>
</tr>
<tr>
<td>Ensemble</td>
<td>0.1546</td>
<td>0.1462</td>
<td>0.1274</td>
<td>0.1217</td>
</tr>
<tr>
<td>RVM+SPD(k=2)[10]</td>
<td></td>
<td></td>
<td></td>
<td>0.12470</td>
</tr>
<tr>
<td>SVR+SPD(k=2)[10]</td>
<td></td>
<td></td>
<td></td>
<td>0.12654</td>
</tr>
<tr>
<td>RVM+DN[10]</td>
<td></td>
<td></td>
<td></td>
<td>0.12873</td>
</tr>
<tr>
<td>SVR+DN[10]</td>
<td></td>
<td></td>
<td></td>
<td>0.12573</td>
</tr>
<tr>
<td>RVM+APM[10]</td>
<td></td>
<td></td>
<td></td>
<td>0.13556</td>
</tr>
<tr>
<td>SVR+APM[10]</td>
<td></td>
<td></td>
<td></td>
<td>0.13112</td>
</tr>
<tr>
<td>APM[9]</td>
<td></td>
<td></td>
<td></td>
<td>0.13517</td>
</tr>
</tbody>
</table>


