Cross-temporal Link Prediction

Satoshi Oyama*, Kohei Hayashi† and Hisashi Kashima‡§
*Graduate School of Information Science and Technology, Hokkaido University
†Graduate School of Information Science, Nara Institute of Science and Technology
‡Graduate School of Information Science and Technology, The University of Tokyo
§Basic Research Programs PRESTO, Synthesis of Knowledge for Information Oriented Society

***Abstract*—The increasing interest in dynamically changing networks has led to growing interest in a more general link prediction problem called *temporal link prediction* in the data mining and machine learning communities. However, only links in identical time frames are considered in temporal link prediction. We propose a new link prediction problem called *cross-temporal link prediction* in which the links among nodes in different time frames are inferred. A typical example of cross-temporal link prediction is cross-temporal entity resolution to determine the identity of real entities represented by data objects observed in different time periods. In dynamic environments, the features of data change over time, making it difficult to identify cross-temporal links by directly comparing observed data. Other examples of cross-temporal links are asynchronous communications in social networks such as Facebook and Twitter, where a message is posted in reply to a previous message. We adopt a dimension reduction approach to cross-temporal link prediction; that is, data objects in different time frames are mapped into a common low-dimensional latent feature space, and the links are identified on the basis of the distance between the data objects. The proposed method uses different low-dimensional feature projections in different time frames, enabling it to adapt to changes in the latent features over time. Using multi-task learning, it jointly learns a set of feature projection matrices from the training data, given the assumption of temporal smoothness of the projections. The optimal solutions are obtained by solving a single generalized eigenvalue problem. Experiments using a real-world set of bibliographic data for cross-temporal entity resolution showed that introducing time-dependent feature projections improves the accuracy of link prediction.**

***Keywords*-link prediction; temporal data; entity resolution; social network analysis, dimension reduction**

I. Introduction

Link prediction is the task of inferring the existence or absence of certain relationships, such as identity, interaction, and collaboration, among data objects. In a link prediction problem, data objects and the relationships among them are considered nodes and edges in a graph. Link prediction is found in various applications in the fields of information integration, recommender systems, bioinformatics, and social network analysis. In information integration, entity resolution, which determines whether data objects refer to the same real-world entity, can be considered a link prediction problem by regarding the entity identity as a link. In recommender systems, a user product-purchase event can be considered a link between the user and the product, and predicting whether the user will buy the product can be considered a link prediction problem.

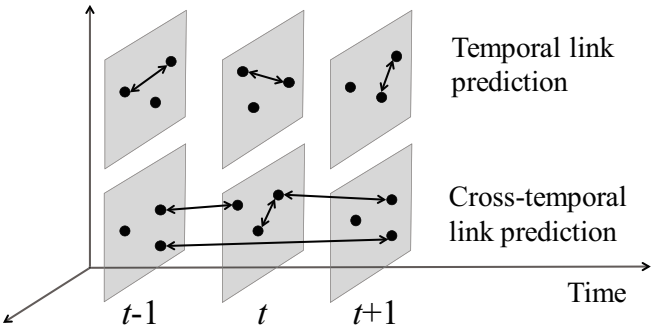

Figure 1. Link prediction problems for dynamically changing data.

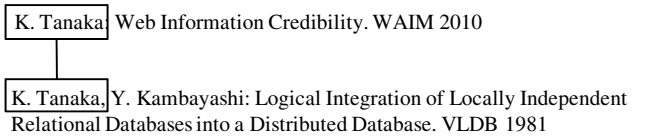

Figure 2. Example of cross-temporal entity resolution.

Link prediction is an important task in link mining [1]. Machine learning techniques proposed for predicting unknown links use the known links in a graph as training data. While conventional techniques are based on the assumption that the links are static, recent work has attempted to predict temporal links in dynamic and time-evolving networks [2]–[6]. In this *temporal link prediction*, only links among nodes within certain time frames are considered (upper part of Figure 1). In the work reported here, we dealt with cross-temporal links, i.e., links among nodes in different time frames (lower part of Figure 1). Previous work on link prediction in dynamic networks has not dealt with this kind of link, so we propose a new problem: *cross-temporal link prediction*.

Cross-temporal link prediction problems appear in various application domains. A typical example is entity resolution for time-evolving entities. Figure 2 illustrates the task of

identifying whether the same author name (in this example, "K. Tanaka," a common name in Japanese) in two bibliographic entries with greatly different publication dates are for the same person. The features that are useful for predicting links change over time, making cross-temporal link prediction more difficult than conventional link prediction. For example, the two instances of "K. Tanaka" in Figure 2 are for the same person, but it is difficult to determine this by simply comparing the features of the two papers, such as the keywords in the titles and the conference venues, since his research interests changed greatly during his long career.

The problem of inferring unobserved asynchronous communications in social networks can also be regarded as a cross-temporal link prediction problem. In asynchronous communication systems such as email, Facebook and Twitter, messages usually have time stamps, and many messages are generated by replying to or forwarding a previous message. If we regard people as nodes, a reply message can be considered a cross-temporal link from the replier at the time of reply to the sender of the original message at the time of the original message. Analysis of asynchronous communications is pervasive in studies using social network analysis. Since all communications among the target individuals are not necessarily available for analysis, considering unobserved communications as well is important for identifying the communication structure. Inferring unobserved asynchronous communications can be considered an extension of prediction of email messages formulated as temporal link prediction [3].

In various domains such as academic publishing, e-commerce, healthcare, and the Web and social media, data have been collected and archived over a long period of time. The longer the time range of such collected data, the more important cross-temporal link prediction is for discovering relationships among the temporally separated data.

In the work reported here, we adopted a dimension-reduction approach to predicting cross-temporal links. High-dimensional data are mapped to a low-dimensional latent feature space, and links are predicted on the basis of data proximity in the feature space. That is, the closer two data objects are to each other in the latent space, the greater the likelihood of a link between them. The projection to the low-dimensional space enables comparing data that are difficult to compare in the original high-dimensional space. For example, in Figure 2, if the terms "Web" and "Database" are mapped to the same latent feature space, they can be used to identify the authors in two bibliographic entries.

The proposed *cross-temporal locality preserving projection (CT-LPP)* method is an extension of the dimension-reduction-based link prediction method proposed by Vert and Yamanishi [9], meaning that it takes into account the time-variation of latent features. We assume that features useful for link prediction change over time and thus introduce different feature projections for different time frames. The data scarcity problem caused by splitting the data into multiple time frames is overcome by jointly learning feature projection matrices under the assumption that the projections are temporally smooth, derived from the idea of multi-task learning [10], [11]. The optimization problem results in a generalized eigenvalue problem, which gives the globally optimal solution.

As an example cross-temporal link prediction task, we used entity resolution in bibliographic data. We conducted experiments using a real-world data set obtained from the DBLP bibliography database. Comparison of the results of the CT-LPP method with those of the conventional method showed that the accuracy of link prediction was improved by introducing the time-dependent feature projections. We also conducted experiments using real-world data obtained from the Enron dataset, as an example asynchronous communications inference task. Due to space limitations, however, the results are not included in this paper.

II. Related Work

Link prediction methods can be roughly divided into two approaches: the topological-information-based approach and the node-information-based approach. The former uses only adjacent matrices of the graph, and the latter uses node information such as feature vectors of nodes or similarity values among nodes. In this paper, we discuss the latter type, i.e., link prediction based on node information.

A typical node-information-based method is the pair-wise support vector machine (pair-wise SVM), which combines node-wise kernel matrices to construct a pair-wise kernel matrix [12], [13]. There are also several supervised learning methods using node information as well as topological information [14], [15]. Several previous studies, e.g., [16], applied the statistical relational learning framework to link prediction.

Dimension-reduction-based link prediction is also used in various domains. Our cross-temporal link prediction method is an extension of one such method [9], which was originally applied to the task of metabolic network reconstruction using genomic data. Yamanishi proposed a method for predicting links between heterogeneous objects such as compounds and proteins by using two mappings of the heterogeneous objects to a common Euclidean space [17]. Khoshneshin and Street proposed a method for implementing collaborative filtering by mapping users and items in a common Euclidean space [18]. Sarkar and Moore described a method for predicting temporal links on the basis of the mapping of data objects to a latent Euclidean space and a system for analyzing changes in co-authorship over time [2].

Entity resolution, the determining of the correspondence between data objects in documents or databases and real-world entities, is an important step in information integration. It has long been an area of interest in both database research and linguistic research as "record linkage" or

"reference resolution." Entity resolution is accomplished by matching observed data objects on the basis of a measure of similarity between them, which is defined heuristically or obtained from training examples using machine learning [19]. The methods mentioned above use the features of only the two data objects for which a matching decision is considered. Collective entity resolution uses information related to other data objects as well and jointly performs resolution among more than two data objects. Previous studies, except for that of Oyama *et al.* [20], did not explicitly deal with the possibility of changes in entity features over time. They used a similarity measure reflecting the time interval between observations to match data objects observed in distant time periods. However, the similarity measure was formulated on the basis of domain knowledge, and no learning was used.

Multi-task learning [10] improves prediction accuracy by jointly learning models for multiple related tasks. A multi-task learning approach proposed by Micchelli and Pontil [11] is based on the assumption that learned model parameters for related tasks are similar. In their method, models for related tasks are made similar to each other by introducing the norm of the difference between model parameters as a penalty term in model estimation. In our method, the norm of the difference between projection matrices of adjacent time frames is introduced as a regularization term in optimization.

III. Link Prediction using Dimension Reduction

In this section, we review a link prediction method using dimension reduction. In many link prediction problems, while the feature vectors representing data objects are high-dimensional, the number of latent features actually effective for predicting links is assumed to be relatively small. Therefore, the accuracy of link prediction can be improved by identifying and working in a low-dimensional latent feature space. In supervised linear dimension-reduction methods, a linear projection $\mathbf{W}$ from the original D-dimensional feature space to a $d(< D)$-dimensional latent feature space is learned from training data consisting of data objects known to have or not to have links between them. The learning process seeks the linear projection $\mathbf{W}$ that makes the distance in the mapped space,

$$\|\mathbf{W}\mathbf{x} - \mathbf{W}\mathbf{y}\|,$$

as small as possible, where $\mathbf{x}$ and $\mathbf{y}$ are two nodes known to have a link between them. After the learning process is completed, two data objects with an unknown link status are mapped to the latent space by using $\mathbf{W}$. If the mapped images of the two data objects are sufficiently close to each other, they are considered to have a link between them.

Assume that we have N training data objects, $\mathbf{x}_1, \ldots, \mathbf{x}_N$, and that each data object $\mathbf{x}_i$ is represented in a D dimensional feature vector. The method proposed in [9] uses locality preserving projections [21] [1] to find the optimal linear projection matrix $\mathbf{W}^*$ by solving the following optimization problem:

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \sum_{i,j} A_{ij} \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|_2^2,$$

where $\|\cdot\|_2$ is the Euclidean norm (2-norm), and $\mathbf{A} = \{A_{ij}\}$ is the adjacency matrix defined by

$$A_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ have a link,} \\ 0 & \text{otherwise.} \end{cases}$$

The above optimization problem can be rewritten as

$$\begin{aligned} \mathbf{W}^* = & \arg\min_{\mathbf{W}} \mathbf{tr}\left(\mathbf{W}\mathbf{\Phi}^\mathsf{T}\mathbf{L}\mathbf{\Phi}\mathbf{W}^\mathsf{T}\right) \\ & \text{s. t. } \mathbf{W}\mathbf{\Phi}^\mathsf{T}\mathbf{D}\mathbf{\Phi}\mathbf{W}^\mathsf{T} = \mathbf{I}_d, \end{aligned} \tag{1}$$

where $\mathbf{\Phi}$ is the design matrix defined by $\mathbf{\Phi} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\mathsf{T}$, $\mathbf{D}$ is the diagonal degree matrix in which each element $D_{ii} = \sum_j A_{ij}$ represents the number of links node i has, and $\mathbf{L}$ is the Laplacian matrix defined by $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The constraint in (1) is introduced for avoiding the trivial solution ($\mathbf{W} = \mathbf{0}$) and ensuring the uniqueness of the solution, where $\mathbf{I}_d$ is the $d \times d$ identity matrix.

Solving the above constrained optimization problem is equivalent to solving the following generalized eigenvalue problem:

$$\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}^\mathsf{T}\mathbf{w} = \lambda\mathbf{\Phi}\mathbf{D}\mathbf{\Phi}^\mathsf{T}\mathbf{w}. \tag{2}$$

The optimal linear projection matrix $\mathbf{W}^*$ is obtained by finding d eigenvectors with the smallest positive eigenvalues for the generalized eigenvalue problem.

IV. Cross-temporal Link Prediction

In a dynamic and time-evolving environment, latent features useful for link prediction can change over time. Let the range of the time under consideration be segmented into T consecutive time frames. We use a different feature projection $\mathbf{W}^{(t)}$ for each time frame t, and a data object $\mathbf{x}^{(t)}$ in the time frame t is mapped using the corresponding projection as $\mathbf{W}^{(t)}\mathbf{x}^{(t)}$.

In the learning process, if two data objects, $\mathbf{x}^{(t)}$ and $\mathbf{x}^{(u)}$, belonging to different time frames are known to have a link, the corresponding linear projections, $\mathbf{W}^{(t)}$ and $\mathbf{W}^{(u)}$, are adjusted so that the distance between the two data objects in the mapped space,

$$\|\mathbf{W}^{(t)}\mathbf{x}^{(t)} - \mathbf{W}^{(u)}\mathbf{x}^{(u)}\|_2^2,$$

becomes small. In the link-prediction process, data objects in different time frames are mapped to the same latent feature space by using corresponding time-dependent linear projections, and link predictions are made on the basis of the distances in the latent space.

[1] Vert & Yamanishi [9] do not explicitly interpret their approach as an LPP method.

A. Extending LPP to Allow Temporal Variation

We extended the LPP method so that the set of time-dependent linear projections can be learned from the training data. By concatenating the projection matrices for time frames, we define the parameter matrix to be learned as follows:

$$\widetilde{\mathbf{W}} \equiv \left[\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \ldots, \mathbf{W}^{(T)}\right].$$

We define the design matrix for time frame t by arranging the data vectors in the time frame in rows:

$$\mathbf{\Phi}^{(t)} \equiv \left[\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \ldots, \mathbf{x}_{N(t)}^{(t)}\right]^{\mathsf{T}},$$

where $N(t)$ is the number of training data objects in time frame t, and $\sum_t N(t) = N$. Using the design matrices for time frames, we define the design matrix for all the data as an $N \times TD$ matrix:

$$\widetilde{\mathbf{\Phi}} \equiv \begin{bmatrix} \mathbf{\Phi}^{(1)} & & & \\ & \mathbf{\Phi}^{(2)} & & \\ & & \ddots & \\ & & & \mathbf{\Phi}^{(T)} \end{bmatrix}.$$

Note that $\widetilde{\mathbf{\Phi}}\widetilde{\mathbf{W}}^{\mathsf{T}}$ calculates the feature projection of every data vector by using the projection matrix for the time frame to which the data object belongs.

The learning of time-dependent feature projections is formulated as an optimization problem using the matrices defined above:

$$\widetilde{\mathbf{W}}^* = \arg\min_{\widetilde{\mathbf{W}}} \mathbf{tr}\left(\widetilde{\mathbf{W}}\widetilde{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{L}\widetilde{\mathbf{\Phi}}\widetilde{\mathbf{W}}^{\mathsf{T}}\right)$$
$$\text{s. t. } \widetilde{\mathbf{W}}\widetilde{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{D}\widetilde{\mathbf{\Phi}}\widetilde{\mathbf{W}}^{\mathsf{T}} = \mathbf{I}_d,$$

which is similar to the optimization problem (1) for the conventional LPP method. The degree matrix $\mathbf{D}$ and the Laplacian matrix $\mathbf{L}$ are the same as those defined for the time-independent problem in the previous section.

B. Imposing Temporal Regularization

In the extended method described above, the training data objects are divided among time frames, so the amount of training data available for each time frame can be limited, which increases the risk of overfitting. If there is no training data for a time frame, it is impossible to learn the projection matrix for the time frame. However, it is reasonable to assume that the latent features do not change much between successive time frames. Therefore, we impose an additional requirement for the projection matrices—the matrices for successive time frames must be similar. In multi-task learning, learning a projection matrix for each time frame is regarded as a single task, the learning tasks for successive time frame are regarded as related tasks, and the learned parameters (projection matrices) for related tasks are assumed to be similar.

To impose temporal smoothness on the projection matrices, we add a temporal-regularization term,

$$\sum_{t=1}^{T-1} \|\mathbf{W}^{(t)} - \mathbf{W}^{(t+1)}\|_F^2, \tag{3}$$

to the objective function, where $\|\cdot\|_F$ is the Frobenius norm (2-norm) of a matrix. We introduce matrix $\mathbf{\Lambda}$ with size $TD \times TD$ for temporal regularization:

$$\mathbf{\Lambda} \equiv \begin{bmatrix} \mathbf{I} & -\mathbf{I} & & & & \\ -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} & & & \\ & -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} & & \\ & & & \ddots & & \\ & & & -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} \\ & & & & -\mathbf{I} & \mathbf{I} \end{bmatrix},$$

where $\mathbf{I}$ is the $D \times D$ identity matrix. That the following equation holds is easily shown.

$$\widetilde{\mathbf{W}}\mathbf{\Lambda}\widetilde{\mathbf{W}}^{\mathsf{T}} = \sum_{t=1}^{T-1} \|\mathbf{W}^{(t)} - \mathbf{W}^{(t+1)}\|_F^2$$

Using this relationship, we formulate the optimization problem to learn time-dependent LPP with temporal regularization as

$$\widetilde{\mathbf{W}}^* = \arg\min_{\widetilde{\mathbf{W}}} \mathbf{tr}\left(\widetilde{\mathbf{W}}\left(\widetilde{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{L}\widetilde{\mathbf{\Phi}} + \sigma\mathbf{\Lambda}\right)\widetilde{\mathbf{W}}^{\mathsf{T}}\right)$$
$$\text{s. t. } \widetilde{\mathbf{W}}\widetilde{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{D}\widetilde{\mathbf{\Phi}}\widetilde{\mathbf{W}}^{\mathsf{T}} = \mathbf{I}_d, \tag{4}$$

where σ is a constant specifying the strength of temporal regularization.

This optimization problem can also be reduced to a generalized eigenvalue problem:

$$\left(\widetilde{\mathbf{\Phi}}\mathbf{L}\widetilde{\mathbf{\Phi}}^{\mathsf{T}} + \sigma\mathbf{\Lambda}\right)\mathbf{w} = \lambda\widetilde{\mathbf{\Phi}}\mathbf{D}\widetilde{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{w}. \tag{5}$$

Note that all projection matrices can be determined simultaneously by finding the eigenvectors of the above problem. We call this method *cross-temporal locality preserving projection (CT-LPP)*. The conventional LPP method, which uses a single time frame for the entire time range, is regarded as a special case of CT-LPP.

We implemented CT-LPP on the Matlab platform and used the built-in `eig` function to solve the generalized eigenvalue problem. Since finding the smallest eigenvalues of (5) is numerically problematic, we found the eigenvectors for the largest positive eigenvalues of the generalized eigenvalue problem:

$$\widetilde{\mathbf{\Phi}}\mathbf{D}\widetilde{\mathbf{\Phi}}^{\mathsf{T}}\mathbf{w} = \mu\left(\widetilde{\mathbf{\Phi}}\mathbf{L}\widetilde{\mathbf{\Phi}}^{\mathsf{T}} + \sigma\mathbf{\Lambda}\right)\mathbf{w}.$$

Note that finding the smallest positive eigenvalues for $\mathbf{A}\mathbf{w} = \lambda\mathbf{B}\mathbf{w}$ is mathematically equivalent to finding the largest positive eigenvalues for $\mathbf{B}\mathbf{w} = \mu\mathbf{A}\mathbf{w}$ by taking $\lambda = 1/\mu$. We set the dimension of the latent feature space (the number of eigenvectors used to form the projection matrix) to ten.

CT-LPP can be kernelized in a way similar to that of [21] and [9]. The size of the generalized eigenvalue problem for the kernelized version would become $TN \times TN$ while the size of the original problem would be $TD \times TD$. For datasets for which the number of training examples is smaller than the number of features, kernelization can save training computation time. Although we considered only undirected links in this paper, the proposed approach can be generalized for the directed link case such as the case described by Yamanishi [17].

In Equation 3, 2-norm is used for the temporal regularization. If 1-norm was used, the difference between two projection matrices in adjacent time frames would be a sparse matrix, and the projection matrix would selectively change its elements only when necessary. This is an interesting property from the viewpoint of change detection and worth further research although, with 1-norm, the problem, Equation (4), can no longer be reduced to an eigenvalue problem but requires iterative optimization.

V. Experiments on Entity Resolution

We experimentally evaluated the ability of our CT-LPP method to determine the identity of real entities represented by data objects observed in different time periods. We used data obtained from the DBLP database[2], which provides bibliographic information for major computer science journals and proceedings. We used the snapshot of the data for 2003 that is publicly available in XML format. We used both journal papers and conference papers. To automatically establish ground-truth labels for use in supervised learning, we assumed that authors with identical given and family names (i.e., surnames) are the same person. That is, if two instances of the same full name, e.g., "Katsumi Tanaka," are the same, the two data objects should be linked. If they have different full names, e.g., "Katsumi Tanaka and Ken Tanaka," they should not be linked. However, the author names in the database do not always include the full given name; some include only the initial. The task was to determine whether two instances of the same abbreviated author name, e.g., "K. Tanaka," in two bibliographic entries are for the same person.

We selected ten cases of first-initial-plus-surname names, which involve a collapsing of many distinct full names. We selected names like J. Smith rather than ones like J. Ullman to ensure a high level of collapsing. We then retrieved papers written by authors with the same surname and a given name starting with the same letter from the DBLP data. We abbreviated the given names to an initial and removed any middle names to mask the author identities. We split the set of bibliographic entries for each author into five disjoint subsets and performed five-fold cross validation. We use data *across* time frames in training, since we assumed batch tasks such as data integration of different data sources. Training and test data were generated by pairing papers with the same abbreviated author name. That is, training was done using links among 80% of the nodes, and testing was done by predicting the links among 20% of the nodes. We used words in titles and journal names and the names of coauthors as features. Since few words appear more than once in a bibliographic entry, we used binary features; that is, the value of the corresponding feature was set to one if a term appeared in the entry and to zero otherwise.

The evaluation metric was the accuracy of the pairwise link prediction. In practice, entity resolution results should satisfy transitivity; that is, if $\mathbf{x}_i$ and $\mathbf{x}_j$ refer to the same entity and $\mathbf{x}_j$ and $\mathbf{x}_k$ refer to the same entity as well, $\mathbf{x}_i$ and $\mathbf{x}_k$ must refer to the same entity. To satisfy this condition, clustering is usually performed as post-processing after pairwise prediction. Thus, the performance of entity resolution is affected not only by the pairwise predictions but also by the choice of post-processing method. Since our interest is the accuracy of link prediction, we neglected the post-processing and simply estimated the pairwise accuracy of the predictions.

All data objects in the test set were projected to a low-dimensional feature space by using the projection matrices learned from the training data. The number of dimensions was tuned manually and set to ten in the experiments. The pairs of test data were sorted in ascending order of the distance between the two data objects in each pair. A pair was considered to be linked if the distance was less than or equal to a threshold and considered not to be linked if the distance was greater. We plotted the ROC curves for various value of the threshold. An ROC curve shows the true positive rate and the false positive rate as the threshold is varied. As a summary performance measure for different threshold values, we used the AUC (area under the ROC curve) value.

The AUC values for different time frame lengths (unit time of one year) are shown in Table I. The constant σ for the strength of the temporal regularization in Equation (5) was set to 0.01. Each AUC value is the average of the AUC values obtained in the five cross-validation trials. The rightmost column is for the baseline method using conventional LPP, which does not consider time variations. The results were obtained by setting the time frame longer than the time range fo all the data so that CT-LPP would use a single time frame. For "A. Gupta," "H. Suzuki," and "H. Zhang," the time range was not more than 20 years, so the results for frame length 20 were the same as for the baseline method. In eight of the ten cases, the CT-LPP method outperformed the baseline method.

There was no single time frame length optimal for all example cases, suggesting that the proposed method is not particularly sensitive to the choice of the length. As shown in Table I, it outperformed the baseline method in seven of

[2]http://dblp.uni-trier.de/

Table I
AUCs FOR DIFFERENT TIME FRAME LENGTHS

Author name	Time frame length			Time independent
	5	10	20	
J. Anderson	0.676	0.691	**0.749**	0.644
A. Gupta	0.590	0.588	-	**0.592**
D. Johnson	0.611	0.631	0.623	**0.692**
J. Mitchell	0.589	**0.604**	0.590	0.601
M. Sato	0.607	**0.636**	0.615	0.522
J. Smith	0.653	0.643	**0.657**	0.593
H. Suzuki	**0.663**	0.563	-	0.552
K. Tanaka	**0.688**	0.652	0.679	0.604
Y. Wang	**0.592**	0.538	0.563	0.535
H. Zhang	0.584	**0.588**	-	0.526

the ten cases for three time frame lengths.

VI. Conclusion

Cross-temporal links appear in various domains such as bibliographic data and social networks. Predicting such cross-temporal links is a fundamental problem in entity resolution, automatic link generation, social network analysis, and recommendation. To the best of our knowledge, this is the first paper that gives a unified approach to this problem.

The contributions of this work are as follows: (1) Our proposed problem, "cross-temporal link prediction," is to predict links between data objects in distant time periods, something that has not been considered in previous work on link prediction in a dynamic environment. This problem is more important the longer the time range of the target data set. (2) Our proposed CT-LPP method, an extension of the conventional link prediction method, can learn time-dependent feature projections that map data in different time frames to the same latent feature space. Using multi-task learning, it jointly learns the set of projection matrices for different time frames by solving a single generalized eigenvalue problem. (3) Experimental evaluation using an example of cross-temporal link prediction for cross-temporal entity resolution showed that introducing time-dependent feature projections improves the accuracy of link prediction.

Acknowledgment

SO is supported by a Grant-in-Aid for Young Scientists (No. 21700106) and by a grant for Toyota Physical & Chemical Research Institute Scholars. KH is supported by a Grant-in-Aid for JSPS Fellows.

References

[1] L. Getoor and C. P. Diehl, "Link mining: a survey," *SIGKDD Explorations*, vol. 7, no. 2, pp. 3–12, 2005.

[2] P. Sarkar and A. W. Moore, "Dynamic social network analysis using latent space models," *SIGKDD Explorations*, vol. 7, no. 2, pp. 31–40, 2005.

[3] K. Hayashi, J. Hirayama, and S. Ishii, "Dynamic exponential family matrix factorization," in *PAKDD*, 2009, pp. 452–462.

[4] R. Raymond and H. Kashima, "Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs," in *ECML PKDD*, 2010, pp. 131–147.

[5] Z. Huang and D. K. J. Lin, "The time-series link prediction problem with applications in communication surveillance," *INFORMS J. on Computing*, vol. 21, pp. 286–303, 2009.

[6] D. M. Dunlavy, T. G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," *ACM TKDD*, vol. 5, no. 2, pp. 10:1–10:27, 2011.

[7] E. Adar, L. Zhang, L. A. Adamic, and R. M. Lukose, "Implicit structure and the dynamics of blogspace," in *WWW 2004 Workshop on the Weblogging Ecosystem*, 2004.

[8] S. J. Green, "Building hypertext links by computing semantic similarity," *IEEE TKDE*, vol. 11, pp. 713–730, 1999.

[9] J.-P. Vert and Y. Yamanishi, "Supervised graph inference," in *NIPS 17*, 2005, pp. 1433–1440.

[10] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[11] C. A. Micchelli and M. Pontil, "Kernels for multi-task learning," in *NIPS 17*, 2005, pp. 921–928.

[12] J. Basilico and T. Hofmann, "Unifying collaborative and content-based filtering," in *ICML*, 2004, pp. 9–16.

[13] S. Oyama and C. D. Manning, "Using feature conjunctions across examples for learning pairwise classifiers," in *ECML*, 2004, pp. 322–333.

[14] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *LinkKDD*, 2005.

[15] J. O'Madadhain, J. Hutchins, and P. Smyth, "Prediction and ranking algorithms for event-based network data," *SIGKDD Explorations*, vol. 7, no. 2, pp. 23–30, 2005.

[16] B. Taskar, M. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," in *NIPS 16*, 2004, pp. 659–666.

[17] Y. Yamanishi, "Supervised bipartite graph inference," in *NIPS 21*, 2009, pp. 1841–1848.

[18] M. Khoshneshin and W. N. Street, "Collaborative filtering via euclidean embedding," in *RecSys*, 2010, pp. 87–94.

[19] M. Bilenko, W. W. Cohen, S. Fienberg, R. J. Mooney, and P. Ravikumar, "Adaptive name-matching in information integration," *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16–23, 2003.

[20] S. Oyama, K. Shirasuna, and K. Tanaka, "Identification of time-varying objects on the Web," in *JCDL*, 2008, pp. 285–294.

[21] X. He and P. Niyogi, "Locality preserving projections," in *NIPS 16*, 2004, pp. 153–160.