IEICE TRANS. INF. & SYST., VOL.E95–D, NO.11 NOVEMBER 2012

PAPER

Link Prediction Across Time via Cross-Temporal Locality Preserving Projections

Satoshi OYAMA[†], ***Member***, **Kohei HAYASHI**[††], ***Nonmember***, ***and*** **Hisashi KASHIMA**[††,†††], ***Member***

SUMMARY Link prediction is the task of inferring the existence or absence of certain relationships among data objects such as identity, interaction, and collaboration. Link prediction is found in various applications in the fields of information integration, recommender systems, bioinformatics, and social network analysis. 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 among nodes at the same time point are considered in temporal link prediction. We propose a new link prediction problem called *cross-temporal link prediction* in which the links among nodes at different time points 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 and a real-world set of emails for unobserved asynchronous communication inference showed that introducing time-dependent feature projections improved the accuracy of link prediction.

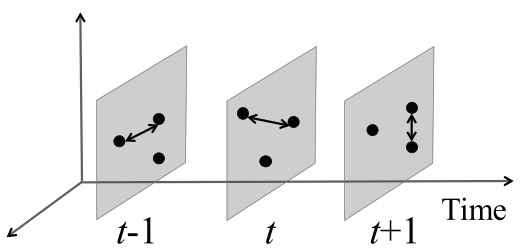
key words: *link prediction, temporal data, entity resolution, social network analysis, dimension reduction*

1. Introduction

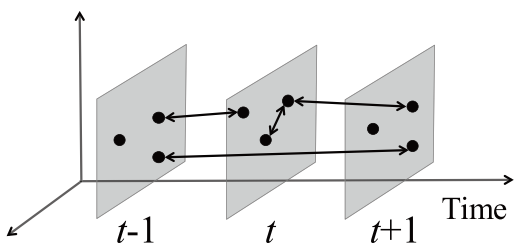
Link prediction is the task of inferring the existence or absence of certain relationships, such as identity, interaction, and collaboration, among data objects. In a link prediction problem, data objects and the relationships among them are considered nodes and edges in a graph. Link prediction is found 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.

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]–[7]. In this *temporal link prediction*, only links among nodes at the same time point are considered (Fig. 1 (a)). In the work reported here, we dealt with cross-temporal links, i.e., links among nodes at different time points (Fig. 1 (b)). 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



(a) Temporal link prediction



(b) Cross-temporal link prediction

Fig. 1 Link prediction problems for dynamically changing data.

Manuscript received January 31, 2012.
Manuscript revised June 18, 2012.

[†]The author is with the Graduate School of Information Science and Technology, Hokkaido University, Sapporo-shi, 060–0814 Japan.
[††]The authors are with the Graduate School of Information Science and Technology, The University of Tokyo, Tokyo, 113–8656 Japan.
[†††]The author is with Basic Research Programs PRESTO, Synthesis of Knowledge for Information Oriented Society.

DOI: 10.1587/transinf.E95.D.2664

Copyright © 2012 The Institute of Electronics, Information and Communication Engineers

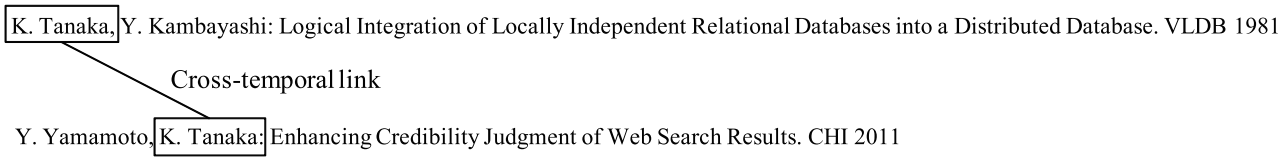


Fig. 2 Example of cross-temporal entity resolution.

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 Fig. 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.

In social media like blogs and Twitter, the posts and tweets sometimes refer to information sources without explicitly linking to them. These reference relationships are called *implicit links*, and it is important to consider not only explicit links but also implicit links when analyzing information flows in social media to identify important users and entries [8]. Data objects in social media, such as blog posts, tweets, and news articles, usually have time stamps. This means that links among them can be considered cross-temporal links. The discovery of implicit links has been of interest to the hypertext community for a long time because of their use in automatically generating links [9]. Most methods for automated link generation use the similarities between data objects. If we take into account the time-evolving effects when computing the similarity, we can generate more accurate links.

The problem of inferring unobserved asynchronous communications in social networks, for example, inferring the existence or nonexistence of an email between two particular people, can also be regarded as a cross-temporal link prediction problem. Analysis of email communications is pervasive in studies using social network analysis. Since all emails among the target individuals are not necessarily available for analysis, considering unobserved emails as well is important for identifying the communication structure. In an asynchronous communication system like email, 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 (Fig. 3). This problem can be considered an extension of prediction of email messages formulated as temporal link prediction [3].

Another example of cross-temporal link prediction is research paper recommendation [10]. A user inputs a paper title as a query, and the system recommends papers that are topically related to the paper but not cited in it. Citations among research papers with different publication dates can also be considered cross-temporal links. Terms used

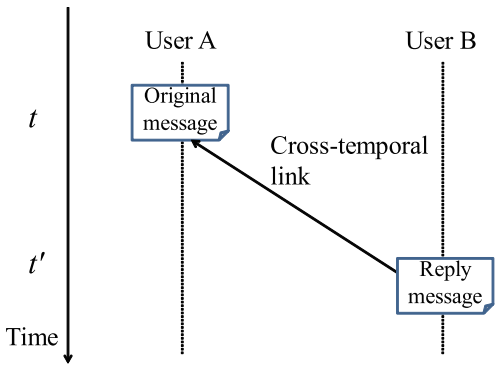


Fig. 3 Asynchronous communication as a cross-temporal link.

in papers dealing with the same topic change over time, so cross-temporal link prediction can play an important role in recommender systems.

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 Fig. 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 [11], 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 [12], [13]. 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. As an example asynchronous communications inference task, we used unknown message inference in email data. We conducted experiments using real-world data obtained from the Enron dataset. The proposed method outperformed the baseline method.

2. 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.

In the data mining community, the link prediction problem is being studied as one of the fundamental tasks for link mining. There are several methods that use only structural information, such as link metrics (e.g., [14]). Matrix factorization approaches [15], [16] are also considered topological-information-based methods. Tensor factorization methods [17] have recently been studied extensively and applied to analysis of relationships among multiple objects. Tucker decomposition [18] is a widely used tensor decomposition method. In temporal link prediction, a time-evolving network is represented by a third-order tensor, where one dimension corresponds to the time frames and the other two dimensions correspond to the node IDs [7]. For example, an email message is represented by

$$(Time, Address1, Address2),$$

where $Time$ is the time when the message was sent, $Address1$ is the email address of the sender, and $Address2$ is the email address of the recipient [3]. The value of an element of a tensor is set to 1 if a corresponding message is observed and 0 otherwise. Inference of the existence of an unknown message is done by tensor completion, i.e., filling in the entries of the tensor.

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 [19]–[21]. There are also several supervised learning methods using node information as well as topological information [22], [23]. Several previous studies, e.g., [24], [25], applied the statistical relational learning framework to link prediction. A similar framework using the exponential random graph model is used in social network analysis [26].

Dimension-reduction-based link prediction is also used in various domains. Our cross-temporal link prediction method is an extension of one such method [11], 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 [27]. Khoshneshin and Street proposed a method for implementing collaborative filtering by mapping users and items in a common Euclidean space [28]. 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 [29]–[33]. 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.* [34], 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 [12] improves prediction accuracy by jointly learning models for multiple related tasks. A multi-task learning approach proposed by Micchelli and Pontil [13] 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.

Evolutionary clustering [35], [36] deals with the problem of clustering temporal data streams such as blogs, where clustering results can change over time. Temporal smoothness is imposed for the difference between clustering results for adjacent time frames. A clustering decision is made between streams of time-stamped data objects, such as blogs consisting of many entries. Here we deal with a different problem—the matching is done between individual time-stamped data objects such as blog entries.

Link analysis is widely used to evaluate the importance of various types of entities. Google's PageRank algorithm, for example, is used to evaluate the importance of Web

pages. A more recently developed link analysis method, which is an extension of the PageRank algorithm, evaluates the significance of historical entities such as people and events [37]. The importance of historical entities varies depending on the time and location. Their relative importance is computed using the links between Wikipedia articles describing them. The links between articles for entities in different time periods can be considered cross-temporal links, and used for propagating the temporal importance scores among the entities.

The paper is an extended version of the preliminary conference version [38]. The main differences between this paper and the preliminary version are as follows: (1) In the preliminary version, we only dealt with cross-temporal entity resolution as an application of cross temporal link prediction. In this paper, we introduce another application: asynchronous communication inference in social networks. We gave a formulation of the problem and experimental results with the Enron Email Dataset as a new section (Sect. 6). (2) The experiments on cross-temporal link prediction have been enriched. More detailed analysis for various experimental settings with visualized results has been provided in Sect. 5. (3) The survey of related works has been extensively expanded to include more complete literature on link prediction, entity resolution, and temporal data analysis.

3. 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 [11] uses locality preserving projections [39][†] 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

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

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. A naive implementation of LPP would therefore require $O(D^3)$ operations. In practice, however, using incomplete Cholesky decomposition can significantly reduce the computational complexity [40].

4. 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.

[†]Vert & Yamanishi [11] do not explicitly interpret their approach as an LPP method.

4.1 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:

$$\widetilde{\mathbf{\Phi}} \equiv \begin{array}{|c|c|c|c|}\hline \mathbf{\Phi}^{(1)} & & & \\ \hline & \mathbf{\Phi}^{(2)} & & \\ \hline & & \ddots & \\ \hline & & & \mathbf{\Phi}^{(T)} \\ \hline \end{array}\ .$$

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.

4.2 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. For example, in bibliographic data, terms characterizing an author's research topics will gradually change over time. 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 a matrix $\mathbf{\Lambda}$ with size $TD \times TD$ for temporal regularization:

$$\mathbf{\Lambda} \equiv \begin{array}{|c|c|c|c|c|c|}\hline \mathbf{I} & -\mathbf{I} & & & & \\ \hline -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} & & & \\ \hline & -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} & & \\ \hline & & \ddots & & & \\ \hline & & & -\mathbf{I} & 2\mathbf{I} & -\mathbf{I} \\ \hline & & & & -\mathbf{I} & \mathbf{I} \\ \hline \end{array}\ ,$$

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. In the optimization problem, the temporal smoothness of the projection matrices is in trade off with the fitting to the training data. Thus if the characteristics of the data change drastically between two adjacent time frames, the projection matrix may also change largely.

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 $\mathbf{Aw} = \lambda\mathbf{Bw}$ is mathematically equivalent to finding the largest positive eigenvalues for $\mathbf{Bw} = \mu\mathbf{Aw}$ by taking $\lambda = 1/\mu$.

CT-LPP can be kernelized in a way similar to that of [39] and [11]. 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 [27].

In Eq. (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, Eq. (4), can no longer be reduced to an eigenvalue problem but requires iterative optimization.

5. 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[†], 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. The number of papers, the number of distinct authors, and the time range of the papers in years for each abbreviated name are shown in Table 1. 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. The names of co-authors were also abbreviated in the same way as we abbreviated the target author names. 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.

Table 1 Statistics of the DBLP data set.

Author name	No. of papers	No. of authors	Time range of papers
J. Anderson	178	27	40
A. Gupta	398	40	20
D. Johnson	226	30	32
J. Mitchell	268	20	37
M. Sato	157	20	27
J. Smith	389	71	45
H. Suzuki	82	19	19
K. Tanaka	176	27	35
Y. Wang	546	148	29
H. Zhang	310	52	20

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 otherwise. 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 Fig. 4. The constant σ for the strength of the temporal regularization in Eq. (5) was set to 0.01. Each AUC value is the average of the AUC values

[†]http://dblp.uni-trier.de/

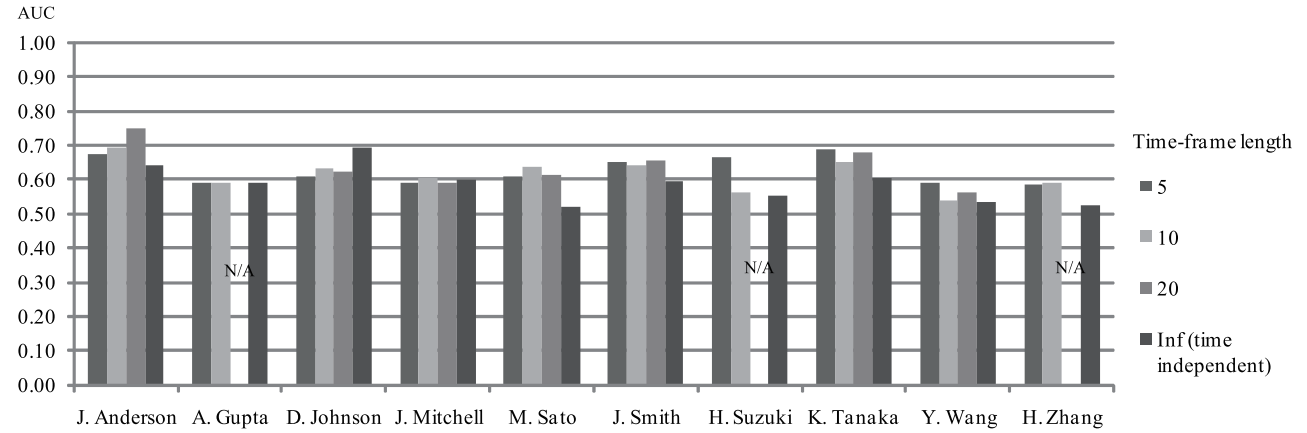


Fig. 4 Predictive performance on the entity resolution task. For each of the ten computer science researchers, AUCs for different time frame lengths are shown. The right-most bar for each name represents the result by the baseline that neglects the time stamps, which is equivalent to the conventional LPP. Use of the time-dependent model results in better prediction accuracy in many cases. 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.

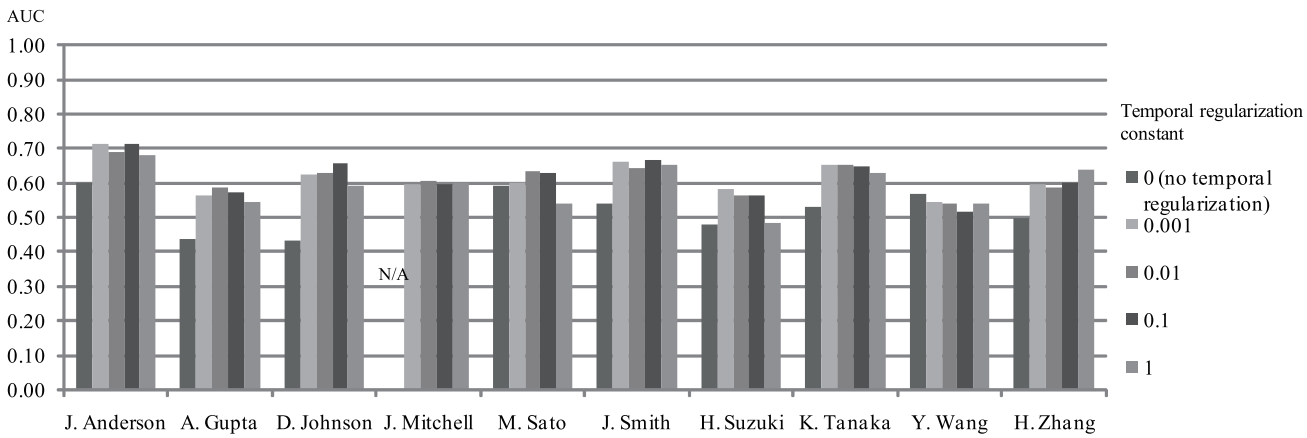


Fig. 5 Predictive performance on the entity resolution task. For each of the ten computer science researchers, AUCs for different temporal regularization constants are shown. Assuming temporal smoothness results in better prediction accuracy in many cases. For "J. Mitchell," there was no training data for a certain time frame, so a projection matrix could not be found when temporal regularization was not used.

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.

Figure 5 shows the AUC values obtained for a time frame of ten years and different values of the temporal regularization constant. The leftmost column shows the results without temporal regularization ($\sigma = 0$ in Eq. (5). In eight cases, the results without temporal regularization were worse than those with it. For "J. Mitchell," there was no training data for a certain time frame, so a projection matrix could not be found when temporal regularization was not used. On the other hand, when the regularization was too strong, the accuracy of the link prediction was degraded, so sometimes the results were worse than when time variation was not considered. This is apparently because the optimization problem given in Eq. (5) with $\sigma \to \infty$ is not equivalent to that given in Eq. (2), so a σ that is too large makes the left-hand matrix of Eq. (5) block-diagonal dominant, which causes the optimization problem to place importance on only temporal smoothness and to ignore the training data.

There was no optimal single parameter setting for all example cases, suggesting that the proposed method is not particularly sensitive to the choice of parameters. As shown in Fig. 4, it outperformed the baseline method in seven of

the ten cases for three time frame lengths. Similarly, it outperformed the baseline method in seven of the ten cases for $\sigma = 0.001$ and 0.01 (Fig. 5).

Our experiments showed that the optimal length of a time frame depends on the data set. This means that adapting the time segmentation to each data set by, for example, using a change detection technique [41] will lead to more accurate link prediction. Automatically setting the temporal regularization constant to different values for different time frame pairs would also increase the accuracy.

6. Experiments on Unobserved Communications Inference

We also experimentally evaluated the ability of our CT-LPP method to infer unobserved communications in a social network. Unlike the entity resolution problem, since each node has no features, we used IDs (i.e., email addresses) as features. The feature vector for a node is an N'-dimensional binary vector (N' is the number of distinct email addresses in the data) in which the value of the feature corresponding to its email address of the node is set to 1 and the values of the other features are set to zero. In this representation, feature vectors of nodes are linearly independent and equally distant from each other. Low-dimensional feature projection places nodes that have a high possibility of mutual communication near each other.

The data used for the experiments was taken from the Enron Email Dataset[†]. The emails that were taken were sent during a 181-day period (1 January to 30 June 2001). From these, we used for link prediction those sent from the 92 email addresses from which ten or more emails were sent and to which ten or more emails were sent. In total, 1118 pairs of original and reply emails sent during a 178-day period were used.

The set of pairs was randomly divided into five subsets, and five-fold cross validation was performed by training with 80% of the pairs and testing with 20% of the pairs. The length of the time frame was set to one day, and the dimension of the mapped space was set to ten. The CT-LPP method learned 178 feature mappings. The value of the temporal regularization constant σ was set to 0.001.

The email address of the original email in each test pair was mapped into a low-dimensional space by using the learned projection for the time frame to which it belonged. Then, every email address was mapped by the projection of every time frame that was on and after the time frame of the original email. If the original email belonged to the i-th time frame, $(T-i+1)\times N'$ combinations of mapped images of the nodes were generated in the low-dimensional feature space. The mapped nodes were sorted in ascending order by their distance from the image of the email address of the original message. The AUC was calculated by comparing the sorted (node ID, time frame) pairs with the fact whether a reply from the node during the time frame was actually observed in the test set.

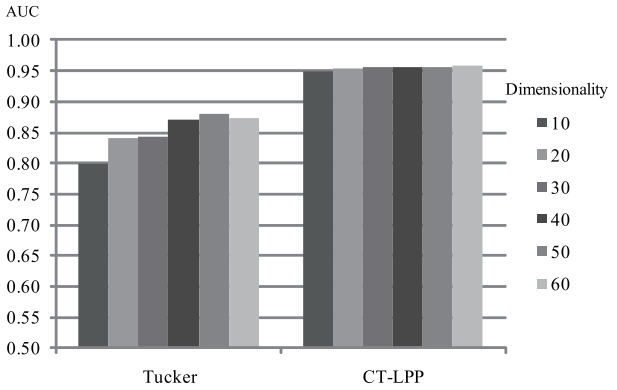


Fig. 6 Predictive performance on the unobserved communication inference task using the Enron dataset. AUCs for different dimensionality are shown for the proposed method and a standard tensor decomposition method (Tucker decomposition). The proposed method consistently outperforms the tensor decomposition method.

We used a method using tensor completion as a baseline method for comparison. In a cross-temporal link prediction problem, data can be represented by a fourth-order tensor (see also Fig. 3):

$$(Time1, Address1, Time2, Address2),$$

where $Time1$ is the time the original message was sent, $Address1$ is the sender of the original message (the recipient of the reply message), $Time2$ is the time the reply message was sent, and $Address2$ is the sender of the reply message. The size of the tensor is $T \times N' \times T \times N'$ where $T(= 178)$ is the number of days and $N'(= 92)$ is the number of distinct email addresses. A straightforward approach to this cross-temporal link prediction problem is using tensor completion. Initially, the values of the tensor's elements corresponding to the pairs of emails in the training data are set to 1, and the values of the other elements are set to 0. Decomposing the tensor into low-rank matrices and recomposing them back into a tensor fills in the unobserved (zero-valued) part of the original tensor. For each original email in the test set, $(T - i + 1) \times N'$ elements of the tensor are sorted in descending order by value, and the AUC is calculated in the same way as described above. We used Tucker decomposition implemented in MATLAB Tensor Toolbox[††].

The AUC value was calculated for each original email in the test set, and the average of all AUC values (micro-average) was computed. The dimensionality of each mode in the Tucker decomposition and the dimensionality of feature projections in CT-LPP were varied from 10 to 60 by every 10 values. The micro-average AUC values for each dimensionality are presented in Fig. 6. The proposed method outperformed the baseline method in all cases.

The relatively high AUC values are attributed to the observation that it is not uncommon to see multiple reply messages to an email from the same email address in the same day. In such cases, the same data can appear in both

[†]http://www.cs.cmu.edu/~enron/
[††]http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/

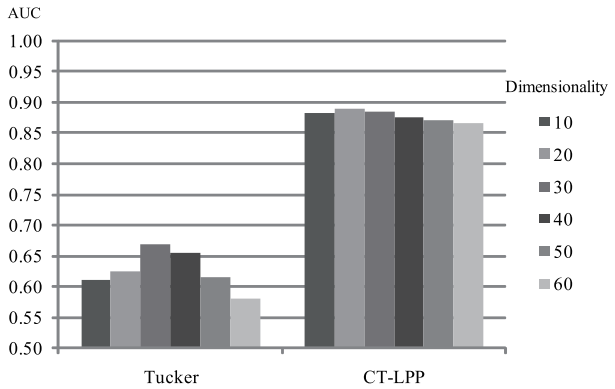


Fig. 7 Predictive performance on the unobserved communication inference task using the Enron dataset with no duplicates. AUCs for different dimensionality are shown for the proposed method and the Tucker decomposition. The proposed method consistently outperforms the tensor decomposition method.

the training and test sets. To eliminate this effect, when we found more than one reply message from the same email address to an email in the same day, we kept only one message and removed the others from the data set, which reduced the number of pairs of original and reply emails to 434. We split them into 80% training data and 20% test data and conducted cross-validation as in the previous experiment. As shown in Fig. 7, although the AUC values were reduced, the proposed method still outperformed the baseline method.

7. Conclusion

Cross-temporal links appear in various domains such as bibliographic data, social media, asynchronous communications, and e-commerce. 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 examples of cross-temporal link prediction for cross-temporal entity resolution and inference of unobserved asynchronous communications showed that both time-dependent feature projection and temporal regularization improve the accuracy of link prediction.

Acknowledgements

SO is supported by a Grant-in-Aid for Scientific Research (No.24650061) and by a grant from the Artificial Intelligence Research Promotion Foundation. 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," Proc. 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp.452–462, 2009.
[4] T.R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher, "MetaFac: Community discovery via relational hypergraph factorization," Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp.527–536, 2009.
[5] R. Raymond and H. Kashima, "Fast and scalable algorithms for semi-supervised link prediction on static and dynamic graphs," Proc. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), pp.131–147, 2010.
[6] Z. Huang and D.K.J. Lin, "The time-series link prediction problem with applications in communication surveillance," INFORMS J. Computing, vol.21, pp.286–303, 2009.
[7] D.M. Dunlavy, T.G. Kolda, and E. Acar, "Temporal link prediction using matrix and tensor factorizations," ACM Trans. Knowledge Discovery from Data (TKDD), vol.5, no.2, pp.10:1–10:27, 2011.
[8] E. Adar, L. Zhang, L.A. Adamic, and R.M. Lukose, "Implicit structure and the dynamics of blogspace," Proc. WWW 2004 Workshop on the Weblogging Ecosystem, 2004.
[9] S.J. Green, "Building hypertext links by computing semantic similarity," IEEE Trans. Knowl. Data Eng., vol.11, no.5, pp.713–730, 1999.
[10] M.D. Ekstrand, P. Kannan, J.A. Stemper, J.T. Butler, J.A. Konstan, and J.T. Riedl, "Automatically building research reading lists," Proc. Fourth ACM Conference on Recommender Systems (RecSys), pp.159–166, 2010.
[11] J.P. Vert and Y. Yamanishi, "Supervised graph inference," Advances in Neural Information Processing Systems 17 (NIPS), pp.1433–1440, 2005.
[12] R. Caruana, "Multitask learning," Mach. Learn., vol.28, no.1, pp.41–75, 1997.
[13] C.A. Micchelli and M. Pontil, "Kernels for multi-task learning," Advances in Neural Information Processing Systems 17 (NIPS), pp.921–928, 2005.
[14] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," Proc. 12th International Conference on Information and Knowledge Management (CIKM), pp.556–559, 2003.
[15] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," Advances in Neural Information Processing Systems 13 (NIPS), pp.556–562, 2001.
[16] N. Srebro, J. Rennie, and T. Jaakkola, "Maximum-margin matrix factorization," Advances in Neural Information Processing Systems 17 (NIPS), pp.1329–1336, 2005.
[17] T.G. Kolda and B.W. Bader, "Tensor decompositions and applications," SIAM Review, vol.51, no.3, pp.455–500, 2009.
[18] L. Tucker, "Some mathematical notes on three-mode factor analysis," Psychometrika, vol.31, no.3, pp.279–311, 1966.

[19] J. Basilico and T. Hofmann, "Unifying collaborative and content-based filtering," Proc. 21st International Conference on Machine Learning (ICML), pp.9–16, 2004.
[20] S. Oyama and C.D. Manning, "Using feature conjunctions across examples for learning pairwise classifiers," Proc. 15th European Conference on Machine Learning (ECML), pp.322–333, 2004.
[21] A. Ben-Hur and W.S. Noble, "Kernel methods for predicting protein-protein interactions," Bioinformatics, vol.21, no.Suppl. 1, pp.i38–i46, 2005.
[22] M.A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," Proc. Workshop on Link Discovery: Issues, Approaches and Applications (LinkKDD), 2005.
[23] 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.
[24] A. Popescul and L.H. Ungar, "Statistical relational learning for link prediction," Proc. IJCAI Workshop on Learning Statistical Models from Relational Data, 2003.
[25] B. Taskar, M. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," Advances in Neural Information Processing Systems 16 (NIPS), pp.659–666, 2004.
[26] C.J. Anderson, S. Wasserman, and B. Crouch, "A p^* primer: Logit models for social networks," Social Networks, vol.21, pp.37–66, 1999.
[27] Y. Yamanishi, "Supervised bipartite graph inference," Advances in Neural Information Processing Systems 21 (NIPS), pp.1841–1848, 2009.
[28] M. Khoshneshin and W.N. Street, "Collaborative filtering via euclidean embedding," Proc. Fourth ACM Conference on Recommender Systems (RecSys), pp.87–94, 2010.
[29] M. Bilenko and R.J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp.39–48, 2003.
[30] W.W. Cohen and J. Richman, "Learning to match and cluster large high-dimensional data sets for data integration," Proc. 8th ACM SIGKDD Conference on Knowledge Discovery and Data mining (KDD), pp.475–480, 2002.
[31] H. Han, L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis, "Two supervised learning approaches for name disambiguation in author citations," Proc. 4th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL), pp.296–305, 2004.
[32] S. Sarawagi and A. Bhamidipaty, "Interactive deduplication using active learning," Proc. 8th ACM SIGKDD Conference on Knowledge Discovery and Data mining (KDD), pp.269–278, 2002.
[33] S. Tejada, C.A. Knoblock, and S. Minton, "Learning domain-independent string transformation weights for high accuracy object identification," Proc. 8th ACM SIGKDD Conference on Knowledge Discovery and Data mining (KDD), pp.350–359, 2002.
[34] S. Oyama, K. Shirasuna, and K. Tanaka, "Identification of time-varying objects on the Web," Proc. 8th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL), pp.285–294, 2008.
[35] D. Chakrabarti, R. Kumar, and A. Tomkins, "Evolutionary clustering," Proc. 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp.554–560, 2006.
[36] Y. Chi, X. Song, D. Zhou, K. Hino, and B.L. Tseng, "Evolutionary spectral clustering by incorporating temporal smoothness," Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp.153–162, 2007.
[37] Y. Takahashi, H. Ohshima, M. Yamamoto, H. Iwasaki, S. Oyama, and K. Tanaka, "Evaluating significance of historical entities based on tempo-spatial impacts analysis using wikipedia link structure," Proc. 22nd ACM Conference on Hypertext and Hypermedia (HT), pp.83–92, 2011.
[38] S. Oyama, K. Hayashi, and H. Kashima, "Cross-temporal link prediction," Proc. 11th IEEE International Conference on Data Mining (ICDM 2011), pp.1188–1193, 2011.
[39] X. He and P. Niyogi, "Locality preserving projections," Advances in Neural Information Processing Systems 16 (NIPS), pp.153–160, 2004.
[40] F.R. Bach and M.I. Jordan, "Kernel independent component analysis," J. Machine Learning Research, vol.3, pp.1–48, 2002.
[41] J.P. Vert and K. Bleakley, "Fast detection of multiple change-points shared by many signals using group LARS," Advances in Neural Information Processing Systems 23 (NIPS), pp.2343–2351, 2010.

Satoshi Oyama is an associate professor in the Graduate School of Information Science and Technology, Hokkaido University, Japan. He has been working on machine learning and data mining and their applications to the Web, including domain-specific search, relation discovery, and object identification. He received his B.Eng., M.Eng., and Ph.D. degrees from Kyoto University in 1994, 1996, and 2002, respectively. He was a research fellow of the Japan Society for the Promotion of Science from 2001 to 2002. He was an assistant professor in the Graduate School of Informatics at Kyoto University from 2002 to 2009. He was a visiting assistant professor in the Department of Computer Science at Stanford University from 2003 to 2004.

Kohei Hayashi received the B.Eng. degree from Ritsumeikan University in 2007, and M.Eng. and Ph.D. degrees from Nara Institute of Science and Technology in 2009 and 2012, respectively. He is currently a JSPS Postdoc at the University of Tokyo. His research interests are in machine learning, Bayesian modeling, and data mining.

Hisashi Kashima is an associate professor of Department of Mathematical Informatics, the University of Tokyo. Before joining to the faculty, he was a research staff member of Data Analytics Group in Tokyo Research Laboratory of IBM Research during 1999–2009. His research interest includes machine learning and data mining. He received his B.Eng., M.Eng., and Ph.D. degrees from Kyoto University in 1997, 1999, and 2007, respectively.