J Intell Inf Syst (2012) 38:209–239
DOI 10.1007/s10844-011-0151-x

# On context-aware co-clustering with metadata support

**Claudio Schifanella · Maria Luisa Sapino · K. Selçuk Candan**

Received: 26 August 2010 / Revised: 20 January 2011 / Accepted: 25 January 2011 /
Published online: 8 February 2011


**Abstract** In traditional co-clustering, the only basis for the clustering task is a given relationship matrix, describing the strengths of the relationships between pairs of elements in the different domains. Relying on this single input matrix, co-clustering discovers relationships holding among groups of elements from the two input domains. In many real life applications, on the other hand, other background knowledge or metadata about one or more of the two input domain dimensions may be available and, if leveraged properly, such metadata might play a significant role in the effectiveness of the co-clustering process. How additional metadata affects co-clustering, however, depends on how the process is modified to be context-aware. In this paper, we propose, compare, and evaluate three alternative strategies (*metadata-driven*, *metadata-constrained*, and *metadata-injected* co-clustering) for embedding available contextual knowledge into the co-clustering process. Experimental results show that it is possible to leverage the available metadata in discovering contextually-relevant co-clusters, without significant overheads in terms of information theoretical co-cluster quality or execution cost.

**Keywords** Co-clustering · Metadata · Constraints · Context-aware clustering · Concept alignment

This work is partially supported by NSF Grant NSF-III1016921. "One Size Does Not Fit All: Empowering the User with User-Driven Integration."

C. Schifanella (✉) · M. L. Sapino
University of Torino, Corso Svizzera 185, 10149, Torino, Italy
e-mail: schi@di.unito.it

M. L. Sapino
e-mail: mlsapino@di.unito.it

K. S. Candan
Arizona State University, 699 S. Mill Avenue 553, Tempe, AZ 85281, USA
e-mail: candan@asu.edu

## 1 Introduction

Given a set of elements in a feature space and a distance measure defined on this data space, *clustering* techniques (such as K-means (Bishop 2006), spectral clustering (Luxburg 2007; Ng et al. 2001) and Non-negative Matrix Factorization (Lee and Seung 2000; Xu et al. 2003)) can be used for partitioning the input set in such a way that elements that are closer to each other (according to the distance measure) are placed into the same cluster, while elements that are further from each other are placed in different clusters. In many applications, however, we are given two sets of objects and their inter-relationships and need to cluster the objects in each set based on their relationships to the objects in the other set. For example, in market-basket analysis, one can try to understand what groups of products are being bought by what groups of customers by studying the customer-product purchase data (Baier et al. 1997). In this case, the clusters of customers are defined based on what clusters of products they purchase and, at the same time, the clusters of products can be identified based on what groups of customers purchase them. Thus, independently clustering the product and customer sets and then trying to identify which product-clusters correspond to which customer-clusters and vice versa may not be effective.

Techniques which simultaneously discover relationships holding on different groups of elements from the two input sets are known as *co-clustering* (also called biclustering, bidimensional clustering, and subspace clustering) techniques (Madeira and Oliveira 2004). In co-clustering, the input information expresses the strength of the point-to-point relationships between pairs of elements in two (possibly coinciding) domains. This is usually represented as a bi-dimensional matrix (the *relationships matrix*), whose rows are associated to the elements from one domain, while columns correspond to elements from the other domain, and whose numeric element values quantify the relationships existing between the corresponding row and column elements. Given a domain specific objective function (measuring the quality of co-clusters in grouping related data elements), *co-clustering* algorithms partition rows and columns into clusters respectively, in such a way that when these clusters are considered as pairs, they minimize the given objective value (Fig. 1; see also Section 2.1 for more details).

In the last few years the co-clustering approach has been successfully applied in many domains from text mining (Dhillon 2001) to bioinformatics (Cheng and Church 2000; Cho et al. 2004; Madeira and Oliveira 2004). For example, in natural language processing, tokens and their contexts, can be co-clustered to discover their inherent relationships (Freitag 2004; Li and Abe 1998). Similarly, in recommender systems,

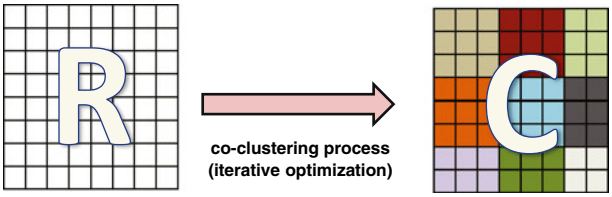



**Fig. 1** The relationship matrix which describes the relationships between entities in two different domains (such as customers and products), provided as input, is (often iteratively) modified, seeking an optimum configuration, to obtain a co-clustering matrix

users and objects (such as movies) can be co-clustered to identify losses of users and their object preferences (George and Merugu 2005). One common shortcoming of the existing co-clustering algorithms, however, is that they consider only a single information source (relationship matrix) to partition object sets into clusters. However, in many application scenarios background knowledge (or contextual metadata) about one or more of the domains may be available and, if leveraged properly, such metadata might play a significant role in the effectiveness of the resulting co-clusters within the given application domain. For example, in market-analysis there can be an a-priori classification of the products based on their price ranges; in movie recommendation domain, there can be a taxonomy of users based on their age or their education degree, as well as a classification of the movies according to their genre. Such a-priori information might complement the relationships matrix to support a contextually-relevant co-clustering, in which the metadata impacts (and possibly corrects) the purely relationships-matrix based grouping of the items.

*Example 1* (Metadata supported co-clustering) Consider, the geographical concepts *"United States"* and *"Afghanistan"*. While geographically distant from each other, in the current geo-political context, these two concepts are related to each other through non-geographical concepts, such as *"taliban"*, *"Bin Laden"*, and *"terrorism"*. Obviously, knowledge about these non-geographical relationships are critical in the design of effective search engines, for example when a user is exploring the geographical concept *"United States"*, related news and blog entries can be fetched and displayed (Cataldi et al. 2009).

Such relationships between concepts in different domains (e.g., the geographical terms *"United States"* and *"Afghanistan"* on one side and the political terms *"terrorism"* and *"Bin Laden"* on the other in Fig. 2) can be discovered through the analysis of an appropriate document corpus, such as newspaper articles. Given a relationship matrix describing the co-occurrences of geographical and political concepts in a newspaper archive, this can be achieved, for example, by using co-clustering algorithms described in Section 2.

Simple co-clustering, however, would fail to account for additional domain-knowledge, such as geographic relationships between countries (*"New York"* being in the *"United States"*). Naturally, if one could leverage geographical data, in addition to the co-occurrence information in the corpus, the relationships one could discover among the sets of concepts would likely be more contextually informed (Candan et al. 2008).

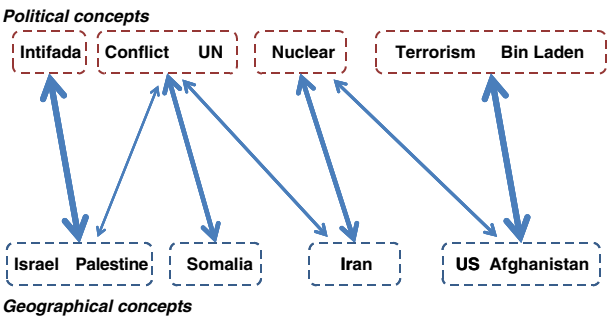



**Fig. 2** Correspondences between sample concepts from two different domains (the thickness of the edge denotes the strength of the correspondence)

### 1.1 Contributions of this paper: co-clustering in the presence of contextual metadata

As mentioned above, if available, additional metadata can be used to condition or *contextualize* the co-clustering process. Of course, an important question that needs to be answered is how do different ways in which additional information is used during co-clustering process impact the resulting co-clusters (Fig. 3).

As we will see in this paper, available secondary metadata can be embedded in the co-clustering process in different ways. We note that, in most existing works, as in the semi-supervised clustering algorithms, such as Basu et al. (2002, 2004), Klein et al. (2002), Wagstaff et al. (2001) and Struyf and Dzeroski (2007), this situation is often handled in an ad hoc and domain-specific manner without a principled understanding of why a particular approach is selected. In fact, as discussed in detail in Section 2.3, existing semi-supervised co-clustering approaches are mostly limited to "constraint-based" approaches to incorporating contextual metadata. In this paper, our goal is not presenting one single preferred approach, but, starting form a well-understood co-clustering reference algorithm, introduce, discuss and compare alternative strategies to understand when and why different strategies can be used in incorporating contextual metadata into the co-clustering process. In particular, in this paper we formally encode metadata as matrices, that express external relationships among elements involved in the co-clustering process, and show different ways in which these matrices can be used to enrich the process of co-clustering. To summarize our contribution, we classify the possible approaches to the problem of co-clustering in the presence of contextual metadata into three broad classes:

- *Metadata-driven co-clustering*: One way to leverage metadata in co-clustering is to modify existing (iterative) co-clustering algorithms in such a way that, at every step, among all the alternative moves which improve the value of the objective function, the selected row/columns move is the one that best preserves the contextual relationships implied by the available metadata. Based on this observation, we first introduce a semi-supervised *search* algorithm in which additional metadata, formally encoded in proper metadata matrices, are leveraged to *affect* the co-clustering process (Fig. 4a and Section 3.3);
- *Metadata-constrained co-clustering*: Existing literature (Pensa and Boulicaut 2008; Chen et al. 2010; Ma et al. 2010; Song et al. 2010) has shown that metadata

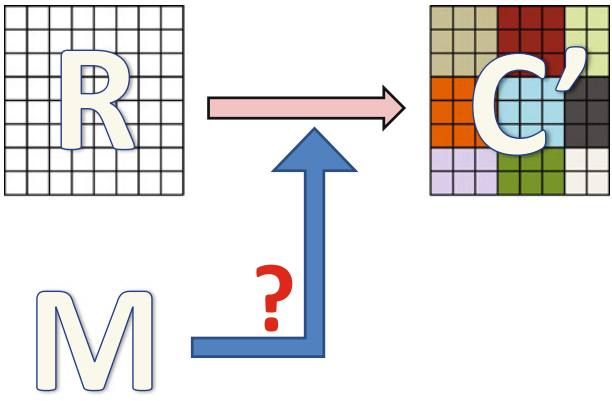



**Fig. 3** If available, additional metadata may help *improve* or *contextualize* the co-clustering process; the major question (which we address in this paper) is how this additional information should be used during co-clustering

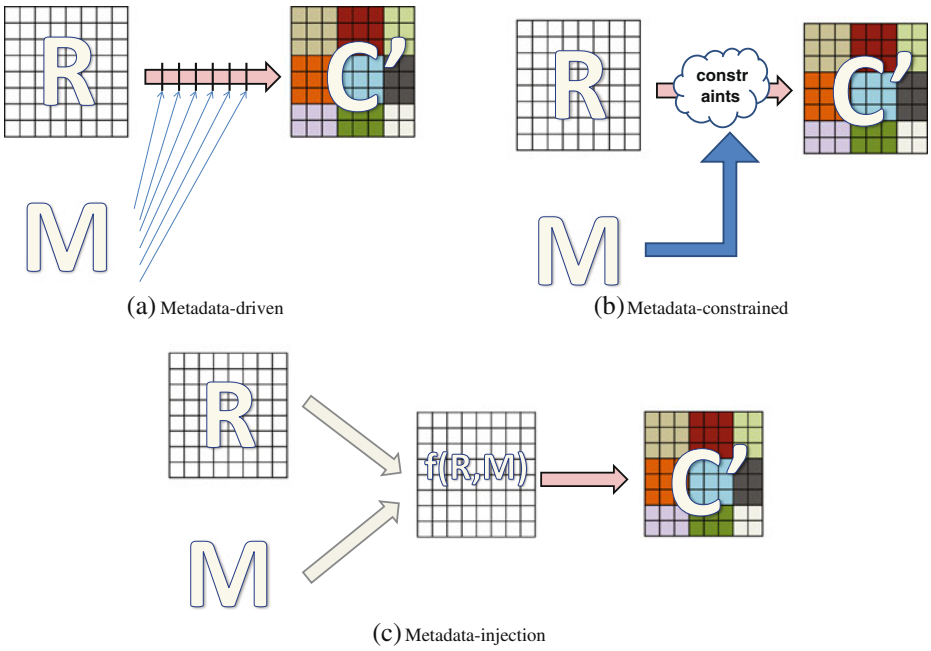



**Fig. 4** In this paper, we partition the possible approaches to the problem of co-clustering in the presence of secondary metadata into three broad classes: **a** metadata-driven, **b** metadata-constrained and **c** metadata-injection

information can be translated into a set of constraints, which are used to limit, at every step of the co-clustering process, the admissible moves. Enforcing these constraints might prevent row/columns moves that would be implied by the pure relationships based optimization process or may imply additional moves that are not required by the original relationship matrix. Following these constraint-based approaches, we discuss how metadata-matrices can be used for leveraging contextual metadata encoded as constraints (Fig. 4b and Section 3.4).

– *Metadata-injection co-clustering*: A third possible strategy we discuss in this paper is to reflect the metadata information into the given relationships matrix, before the co-clustering process is started. Then, a standard co-clustering algorithm can be applied on this *metadata-injected matrix*, whose values are *combinations* of the initial relationship values with the information extracted from the metadata. Thus, we discuss alternative combination schemes for integrating metadata matrices with the relationship matrix (Fig. 4c and Section 3.5).

In the rest of this paper, we present, compare, and evaluate these alternative strategies for embedding the available metadata information into the co-clustering process. Without loss of generality, we rely on the well understood co-clustering algorithm by Dhillon et al. (2003) as our reference approach for basic, non metadata-supported, co-clustering (see Sections 2 for more details). In Section 3 we present in detail the above three different approaches for metadata supported co-clustering. In Section 4.1, we introduce a sample application (concept alignment) where metadata-supported co-clustering is needed and in Sections 4.2 and 4.3, using this application,

we evaluate and compare the effectiveness of alternative schemes under different conditions. Note that, since we have two different sources of information (the original relationship-matrix and the metadata), we need to consider both the co-clustering objective function as well as the agreement with the original metadata in the assessment of the alternative schemes. As we see, different approaches perform differently with respect to the co-clustering objective function versus the results agreement with the metadata. We end the paper with our conclusions in Section 5.

## 2 Related works and background

Co-clustering research has a long history. Hartigan presented one of the first co-clustering algorithms, called *direct clustering*, in early 70's (Hartigan 1972). Hartigan (1972) also introduced various co-clustering models, including an algorithm to perform hierarchical co-clustering. Gaul and Schader (1996) introduced an alternating exchange algorithm for two-mode data. Baier et al. (1997) developed a two-mode (non-)overlapping additive clustering technique and Vichi (2001) proposed an alternating least squares algorithm for the double $k$-means model. Cho et al. (2004) proposed two different algorithms based on two similar squared residue measures: the first is based on the partitional model proposed by Hartigan (1972) while the second is based on the squared residue formulated by Cheng and Church (2000). Dhillon et al. (2003) introduced an *information-theoretic* approach to co-clustering that treats co-clustering as an information theoretical optimization problem. More recently, Banerjee et al. (2007) proposed a generalized approach to co-clustering based on a large class of loss functions called Bregman divergences. More generally, co-clustering approaches can be divided in probability-based models (Hofmann and Puzicha 1999; Kemp et al. 2006; Shan and Banerjee 2008), information-theoretic-based models (Dhillon et al. 2003; Banerjee et al. 2007; Gao et al. 2006), and graph theoretic approaches (Dhillon 2001; Gao et al. 2005). A survey of these co-clustering techniques was presented in Madeira and Oliveira (2004). In this paper, without loss of generality, we use the *information-theoretic* approach proposed by Dhillon et al. (2003) as the basic common building block, necessary to compare the three different metadata techniques introduced in Section 1.1.

### 2.1 Information theoretic co-clustering

In this section, we provide an overview of the basic information-theoretic co-clustering process. In particular, we introduce the proposal of Dhillon et al. (2003) which will be used in this paper as reference algorithm for different approaches to metadata integration.

As we mentioned earlier, the co-clustering process relies on a *relationship matrix*:

**Definition 1** (Relationships matrix) Let $X = \{x_1, ...., x_m\}$ and $Y = \{y_1, ..., y_n\}$ be two domains. A *relationships matrix* of values in these two domains, is an $m \times n$ matrix $R^{\text{rel}}$, where each entry, $r_{ij}^{\text{rel}}$, $1 \leq i \leq m$, $1 \leq j \leq n$ quantifies the strength of the relationship between the $i$-th row element (denoting the domain element $x_i \in X$) and the $j$-th column element (denoting the domain element $y_j \in Y$).

Naturally, the relationships captured by the matrix is domain specific. In the market-basket analysis domain, for example, $X$ can be a set of customers, $Y$ can be a set of products, and the matrix values may quantify each user's purchases.

Given the relationship matrix, $R^{\text{rel}}$, the co-clustering process returns two clusterings, defined by two mappings, $C_X$ and $C_Y$:

- $C_X : \{x_1, ...., x_m\} \rightarrow \mathcal{R} = \{\hat{x}_1, ..., \hat{x}_k\}$ is a mapping which associates each matrix row (i.e., element of the domain $X$) to one of the $k$ row clusters it identifies, and
- $C_Y : \{y_1, ...., y_n\} \rightarrow \mathcal{C} = \{\hat{y}_1, ..., \hat{y}_l\}$ is a mapping which associates each matrix column (i.e., element of the domain $Y$) to a column cluster.

Each row- and column-cluster combination is a *co-cluster* in the relationship matrix, $R^{\text{rel}}$. The strength of each co-cluster depends on the strength of the relationships between the corresponding row-elements and column-elements in the relationship matrix, $R^{\text{rel}}$.

The *information-theoretic co-clustering* algorithm presented by Dhillon et al. (2003) treats the relationships matrix as a joint probability distribution $p(X, Y)$ between two discrete random variables, one associated to the rows and the other to the columns. The input to the algorithm is a non-negative relationship matrix $R^{\text{rel}}$, normalized to 1. Dhillon et al. (2003) define the *optimal co-clustering* as the pair, $(C_X, C_Y)$, of mappings which minimizes the difference between the *mutual information* between the original random variables and the mutual information between the clustered random variables. In other words, the optimal co-clustering minimizes the mutual information loss,

$$MIL = I(X; Y) - I\big(\hat{X}; \hat{Y}\big),$$

where $\hat{X} = C_X(X)$ and $\hat{Y} = C_Y(Y)$. Dhillon et al. (2003) prove that this loss in mutual information can also be expressed in term of the Kullback–Leibler divergence $D_{KL}(p(X, Y) \parallel q(X, Y))$ (also known as the KL-distance or relative entropy) between the joint probability distribution $p(X, Y)$ and the corresponding co-clustered approximation $q(X, Y)$, where

$$q(x, y) = p(x|\hat{x})\ p(y|\hat{y})\ p(\hat{x}, \hat{y}), \quad x \in \hat{x}, y \in \hat{y}.$$

The algorithm operates in stages, where in each stage first the row-clusters are updated in a way that minimizes the Kullback–Leibler divergence function and then the column-clusters are updated under the same criterion. The row- and column-clustering stages are iterated, alternatively considering the marginal distributions of rows and columns, until a locally optimal co-clustering is found. Figure 5 presents this process in pseudocode. Note that the complexity of the algorithm depends on the numbers of rows and columns in the relationship matrix and the numbers of iterations and moves required until the local optimum is found.

### 2.2 Measuring the co-clustering quality

As described above, traditional information theoretic co-clustering algorithms, such as the proposal of Dhillon et al. (2003), treat the relationships matrix, $R^{\text{rel}}$, as a joint probability distribution and define the *optimal co-clustering* as the pair, $(C_X, C_Y)$, of

**Fig. 5** The pseudo-code of the information-theoretic co-clustering algorithm (Dhillon et al. 2003)

Inputs

- let $objFunction(X, Y, \hat{X}, \hat{Y}) = D_{KL}(p(X, Y) \parallel q(X, Y))$
- initial rows assignment: $x_i \rightarrow \hat{x}_g$
- Initial columns assignment: $y_j \rightarrow \hat{y}_h$

**do** {

- $objValue \leftarrow objFunction(X, Y, \hat{X}, \hat{Y})$
- **for each** row $x_i$, $x_i \rightarrow \hat{x}_g$ with $g = argmin_{g=1}^{k} objFunction(X, Y, \hat{X}_{x_i \rightarrow \hat{x}_g}, \hat{Y})$
- **for each** column $y_j$, $y_j \rightarrow \hat{y}_h$ with $h = argmin_{h=1}^{l} objFunction(X, Y, \hat{X}, \hat{Y}_{y_j \rightarrow \hat{y}_h})$
- $newObjValue \leftarrow objFunction(X, Y, \hat{X}, \hat{Y})$

} while $(objValue - newObjValue > threshold)$

mappings which minimizes the difference between the *mutual information* between the original random variables and the mutual information between the clustered random variables. Since the loss of mutual information can also be expressed in terms of the Kullback–Leibler divergence between the joint probability distribution $p(X, Y)$ and the corresponding co-clustered approximation $q(X, Y)$ the quality of the co-clustering solution can be measured in terms of how small the KL-divergence between $p(X, Y)$ and $q(X, Y)$ is; in other words the co-clustering quality can be defined as

$$quality = D_{KL}(p(X, Y) \parallel q(X, Y)).$$

In Section 4.3, we will use this as one of the co-clustering quality measures.

### 2.3 Higher-order co-clustering and semi-supervised co-clustering

High-order co-clustering approaches (Gao et al. 2005, 2006; Long et al. 2006, 2007) also add additional information to the co-clustering process. A major difference between higher-order co-clustering (or multi-relational co-clustering) and the problem we consider in this paper is that in higher-order co-clustering the relationship is defined among the members of more than two domains (e.g., customer-product-season) instead of defining two distinct sources of information (one primary, the other secondary) among the members of a given pair of domains.

Note that if the available metadata are also in the form of a matrix between the members of two domains (e.g., costumer-product), one can potentially represent all available information using a multi-relational representation (e.g., costumer-product-info_source), where the original relationship matrix and the metadata matrix are included as two distinct slices of a 3-mode tensor (see Kolda and Bader (2009) for a review of tensors or high-order matrices). Again, potentially, higher-level co-clustering techniques, such as Tucker-decomposition (Tucker 1966), can then be used to identify clusters of elements based on the similarities of their relationships across these distinct slices. This approach, however, does not apply to the problem we aim to address in this paper: higher-order co-clustering treats the different information sources as equals and does not provide a way to distinguish their roles in the co-clustering process and treats one as the *primary* relationship matrix and the other

as the *secondary* metadata information source that contextualizes the primary information source. One consequence of this is that the resulting co-clusters would be overly constrained: for example, in the costumer-product-info_source example, all the customer-product pairs in a resulting co-cluster would need to have similar relationship vectors with respect to the two information sources. While this might be desirable in some applications of co-clustering with meta-data support, not all applications would require such a limitation.

In semi-supervised clustering and co-clustering approaches, available additional information are used to *drive* the clustering process (rather than for external validation), providing a limited form of supervision. In the clustering literature we can find two kinds of methods: in *similarity-adapting* methods (Basu et al. 2004; Klein et al. 2002; Bilenko and Mooney 2003; Xing et al. 2002) the objective function (i.e. the similarity measure) is modified to directly include additional information, while in *search-based* methods the clustering algorithm itself is modified to bias the search of an appropriate clustering with the help of additional information represented commonly as constraints (*must-link* or *cannot-link*) (Bilenko et al. 2004; Ruiz et al. 2007; Struyf and Dzeroski 2007; Wagstaff et al. 2001; Basu et al. 2002; Demiriz et al. 1999). To the best of our knowledge, in co-clustering literature we can find only proposals that refer to the latter category. In particular, starting from the co-clustering algorithm proposed by Cho et al. (2004), Pensa and Boulicaut (2008) proposed the introduction of constraints (*must-link* or *cannot-link*) among elements to limit the set of resulting co-cluster configurations of gene expression data. Chen et al. (2008, 2009, 2010) propose a non-negative matrix factorization framework in which constraints link together data elements. A constraint-oriented data matrix is then used in the tri-factorization process. A non-negative matrix factorization approach that leverages *must-link* and *cannot-link* constraints is also used by Ma et al. (2010) for word-document co-clustering. Song et al. (2010) propose yet another approach to incorporate constraints into information theoretic co-clustering; they achieve this by using a two-sided hidden Markov random field that allows modelling of both document and word constraints.

As we mentioned in Section 1.1, in this paper we discuss and compare three different mechanism to leveraging metadata for co-clustering, including a *metadata-constrained* method that use must-link and cannot-link constraints as the extends the information theoretic co-clustering algorithm by following the constraint-based approaches mentioned before (Pensa and Boulicaut 2008; Chen et al. 2010; Ma et al. 2010; Song et al. 2010). In contrast, in the *metadata-driven* technique, we introduce a new semi-supervised *search-based* approach to co-clustering (Section 3.3). Finally, in the *metadata-injection* we propose to merge the primary information source (the relationship matrix) with the additional metadata in order to create a new combined input matrix to the information theoretic co-clustering.

## 3 Co-clustering with metadata support

As mentioned in the previous section, most existing co-clustering algorithms do not take into account any information other than the values included in the relationships matrix. As we mentioned in Section 1, however, in many applications, additional metadata may be available and may need to be used to inform the co-clustering

process. Moreover, existing approaches that leverage additional information in co-clustering use only *must-link* and *cannot-link* constraints (Pensa and Boulicaut 2008; Chen et al. 2010; Ma et al. 2010; Song et al. 2010). In this paper, we note that such metadata can be taken into account during co-clustering in three different ways (Fig. 4): first of all, (i) we can modify the objective function in such a way that the moves that are coherent with the metadata are promoted. Alternatively, (ii) we can introduce metadata-based constraints which limit the options or which trigger additional row/column moves, in addition to the ones chosen by the original co-clustering algorithm, to ensure that the moves are coherent with the metadata. Finally, (iii) we can incorporate the metadata directly into the relationships matrix and, thus, define information loss in a way that not only reflects the original relationships, but also the metadata.

In this section, we introduce and discuss these three alternative methods, corresponding the three aforementioned approaches, to include metadata information in the co-clustering process. We first introduce the relevant terminology and notations we will use in the rest of the paper.

### 3.1 Metadata matrices

In this section, without loss of generality, we propose to formally encode the metadata relating pairs of objects in the given domains as matrices:

**Definition 2** (Row- and column-metadata matrices) Given the row domain $X = \{x_1, ...., x_m\}$, the *row-metadata matrix* is an $m \times m$ matrix $R^{\text{row,meta}}$, in which each value, $r_{ij}^{\text{row,meta}}$, $1 \leq i, j \leq m$, quantifies the degree of closeness, between the domain elements $x_i, x_j \in X$. The $n \times n$ *column-metadata matrix*, $R^{\text{col,meta}}$, is defined similarly over the elements in the column domain $Y = \{y_1, ..., y_n\}$.

In addition to the row- and column-matrices, we may also have access to a joint metadata matrix:

**Definition 3** (Joint metadata matrix) Given two domains $X = \{x_1, ...., x_m\}$ and $Y = \{y_1, ..., y_n\}$, the joint metadata matrix is an $m \times n$ matrix $R^{\text{meta}}$, in which each value, $r_{ij}^{\text{meta}}$, $1 \leq i \leq m$, $1 \leq j \leq n$ quantifies the degree of metadata-based correspondence between the $i$-th row element (denoting the domain element $x_i \in X$) and the $j$-th column element (denoting the domain element $y_j \in Y$).

As the original relationship matrix, the joint metadata matrix also relates the elements of the two different domains; the correspondences captured by this matrix, however, are different from the correspondences captured by the original relationship matrix.

### 3.2 Measuring the influence of the metadata

We are thus dealing with a *bi-criteria* co-clustering problem, where we need to take into account both the original relationship matrix and the metadata matrices.

As described in Section 2.2, the quality of the co-clustering process with respect to the original relationship matrix can be quantified by measuring the difference

between the mutual information among the original random variables and the mutual information among the clustered random variables. In the context of co-clustering with metadata, we also need quality measures that measure how well the co-clusters align with the information given by the metadata provided by the user. To measure the overall coherence of the co-clusters with respect to the metadata, we thus define the *average metadata variance* as the average variance of the similarities of the members of the co-clusters with respect to the metadata:

$$metadata\_variance = \sum_{g=1}^{k}\sum_{h=1}^{l} \frac{\sum_{x\in\hat{x}_g, y\in\hat{y}_h}\left(sim(\mathbf{mv}(x),\mathbf{mv}(y)) - \overline{avgSim}_{(\hat{x}_g,\hat{y}_h)}\right)^2}{|\hat{x}_g||\hat{y}_h|\,k\,l}.$$

where $\mathbf{mv}(x)$ and $\mathbf{mv}(y)$ are respectively the row vectors $R^{\text{row,meta}}|_x$ and $R^{\text{col,meta}}|_y$, and

$$\overline{avgSim}_{(\hat{x}_g,\hat{y}_h)} = \frac{\sum_{x\in\hat{x}_g, y\in\hat{y}_h} sim(\mathbf{mv}(x),\mathbf{mv}(y))}{|\hat{x}_g||\hat{y}_h|}$$

In Section 4.3, in addition to the information theoretic measures described in Section 2.2, we will use this as one of the co-clustering quality measures.

### 3.3 Alternative approach I: metadata-driven co-clustering

In metadata-driven co-clustering, the relationship matrix, $R^{\text{rel}}$, is treated as the matrix on which the main information-theoretical objective function is evaluated at every iteration of the algorithm. One or more of the metadata matrices, as they are available, are used inside the main loop of the algorithm as an information source governing which rows and columns movements are controlled. Rows and columns of the metadata matrix are treated as vectors associated to domains objects, and thus, by transitivity, to the rows and columns in the relationships matrix. More specifically, each element $x_i \in X$ is associated with a row vector $\mathbf{mv}(x_i)$ either based on the row metadata matrix (in terms of other elements in X) or, if available, based on the joint metadata matrix (in terms of elements in $Y$). Thus, for each row co-cluster $\hat{x}$ (and, similarly, each column co-cluster $\hat{y}$) of vectors, a centroid can be defined.

**Definition 4** (Metadata-centroid) Given two domains $X$ and $Y$ let $(C_X, C_Y)$ be a co-clustering of $X$ and $Y$ with respect to $R^{\text{rel}}$. For any row co-cluster $\hat{x} \in \hat{X}$ and any column co-cluster $\hat{y} \in \hat{Y}$, their metadata-centroids are defined as

$$\mathbf{centr}(\hat{x}) = \frac{1}{|\,\hat{x}\,|}\sum_{x\in\hat{x}} \mathbf{mv}(x)$$

$$\mathbf{centr}(\hat{y}) = \frac{1}{|\,\hat{y}\,|}\sum_{y\in\hat{y}} \mathbf{mv}(y).$$

The *metadata-driven co-clustering scheme*, depicted in pseudo-code in Fig. 6, uses row and column centroids (in addition to the information theoretical objective function) to drive which rows and columns are clustered together. To start the process the metadata driven version of the algorithm randomly chooses an initial co-clustering. Then, at each iteration, for the current configuration, the row and column cluster

Input

- let $objFunction(X, Y, \hat{X}, \hat{Y}) = D_{KL}(p(X, Y) \parallel q(X, Y))$
- Initial rows assignment: $x_i \rightarrow \hat{x}_g$
- Initial columns assignment: $y_j \rightarrow \hat{y}_h$

do {

- $objValue \leftarrow objFunction(X, Y, \hat{X}, \hat{Y})$
- **for each** row $x_i$
  compute centroids **centr**$(\hat{x})$
  assign $x_i \rightarrow \hat{x}_g$ such that:
  - $g = argmax_{g=1}^{k} cos(\mathbf{centr}(\hat{x}_g), \mathbf{mv}(x_i))$ and
  - $objFunction(X, Y, \hat{X}_{x_i \rightarrow \hat{x}_g}, \hat{Y}) < objValue$
- compute centroids **centr**$(\hat{y})$
- assign $y_j \rightarrow \hat{y}_h$ such that:
  - $h = argmax_{h=1}^{l} cos(\mathbf{centr}(\hat{y}_h), \mathbf{mv}(y_j))$ and
  - $objFunction(X, Y, \hat{X}, \hat{Y}_{y_j \rightarrow \hat{y}_h}) < objValue$
- $newobjValue \leftarrow objFunction(X, Y, \hat{X}, \hat{Y})$

} while $(objValue - newobjValue > threshold)$

**Fig. 6** The pseudo-code for metadata-driven co-clustering: among all the moves that imply a decrease in the objective function, the one that brings the candidate row/column vector closer to the centroid of the target cluster is selected. Note that the algorithm can also be modified to consider only top few alternatives in terms of the minimization of the objective function

centroid vectors are computed. Differently from the purely information theoretic algorithm proposed by Dhillon et al. (2003), at each iteration, the selected row/column move is not necessarily the one which ensures the highest decrease in the KL-divergence based objective function. Instead, among all the moves that imply a drop in the KL-divergence, the one that brings the candidate row/column vector closer to the centroid of the target cluster is chosen. This corresponds to optimizing the metadata-based correspondences between the row and column elements, while also improving the objective function. In the pseudo-code in Fig. 6, the algorithm chooses any movement as long as the move implies a reduction in the objective value. In general, however, the choice can be simply limited to top few ($c$) candidates in terms of the KL-distance minimization. As we will show in Section 4.3, considering few top alternatives is often sufficient for improving the metadata-based co-clustering quality.

Note that, in this scheme, the centroids have to be re-computed for each iteration, potentially increasing the cost of the overall process on a per iteration basis.

### 3.4 Alternative approach II: metadata-constrained co-clustering

In some clustering applications, integration of the metadata to the clustering framework in the form of instance-level constraints has proven to be successful (Bilenko et al. 2004; Ruiz et al. 2007; Struyf and Dzeroski 2007; Wagstaff et al. 2001). Along similar lines, Pensa and Boulicaut (2008) introduced a constraint based algorithm building on the co-clustering framework proposed by Cho et al. (2004). As mentioned before, constraints are also used in other different semi-supervised co-clustering frameworks by Chen et al. (2010), Ma et al. (2010) and Song et al. (2010). In the *constraint*-based approach, the metadata information is used to limit (rather

than drive) the row and column configurations that the co-clustering process can consider. For example, Pensa and Boulicaut (2008) exploit user-defined constraints while minimizing the co-clustering objective function (using the minimum sum-squared residue co-clustering approach (Cho et al. 2004)). Constraint-based approaches to co-clustering can use two kinds of constraints: *must-link* and *cannot-link*. A *must-link* constraint involves two row (or column) elements and states that these elements must be included in the same cluster. In contrast, a *cannot-link* constraint expresses the fact that two row or column elements cannot be together.

In this paper, we generalize the approach proposed by Pensa and Boulicaut (2008) by considering constraints extracted from the metadata as opposed to being provided by the user. In particular we define a set of *must-link* constraints by considering row vectors introduced earlier as follows:

$$ML_X = \left\{(x_i, x_j) \in X \times X \mid i \neq j \ \wedge sim(\mathbf{mv}(x_i), \mathbf{mv}(x_j) > \theta\right\}$$

where $\theta$ is a lowerbound and $sim()$ is a similarity function (such as *cosine*()) that compares the two vectors. The *must-link* set, $ML_Y$, is similarly defined for column vectors. By definition, *must-link* shows a transitive closure property; if row $x_i$ must-link to $x_j$ and $x_j$ must-link to $x_l$, then $x_i$ must-link to $x_l$ as well. Therefore, for each $x_i \in X$, we also compute a transitive closure, $TR_i^X$. The transitive closure, $TR^Y$, of column vectors is defined similarly. In addition, constraints of type *cannot-link* can also be defined and enforced similarly using an upperbound threshold.

In Fig. 7, we present the pseudocode of the metadata-constrained co-clustering with *must-link constraints*. As in the original algorithm, at each iteration, each row/column is moved in the row/column cluster that causes the best reduction in the objective function value. If a row/column is involved in a *must-link* constraint, the

**Fig. 7** Pseudo-code for metadata-constrained co-clustering (for simplicity, this pseudo-code only considers "*must-link*" constraints)

Inputs

- let $objFunction(X, Y, \hat{X}, \hat{Y}) = D_{KL}(p(X, Y) \parallel q(X, Y))$
- initial rows assignment: $x_i \rightarrow \hat{x}_g$
- Initial columns assignment: $y_j \rightarrow \hat{y}_h$

compute $TR_X$
compute $TR_Y$
**do** {

- $objValue \leftarrow objFunction(X, Y, \hat{X}, \hat{Y})$
- **for each** row $x_i$
  - **if** $(\exists TR_s^X \in TR_X \mid x_i \in TR_s^X)$ assign $TR_s^X \rightarrow \hat{x}_g \mid$ $g = argmin_{g=1}^{k} objFunction(X, Y, \hat{X}_{TR_s^X \rightarrow \hat{x}_g}, \hat{Y}) < objValue$
  - **else** assign $x_i \rightarrow \hat{x}_g \mid$ $g = argmin_{g=1}^{k} objFunction(X, Y, \hat{X}_{x_i \rightarrow \hat{x}_g}, \hat{Y}) < objValue$
- **for each** column $y_j$
  - **if** $(\exists TR_s^Y \in TR_Y \mid y_j \in TR_s^Y)$ assign $TR_s^Y \rightarrow \hat{y}_h \mid$ $h = argmin_{h=1}^{l} objFunction(X, Y, \hat{X}, \hat{Y}_{TR_s^Y \rightarrow \hat{y}_h}) < objValue$
  - **else** assign $y_j \rightarrow \hat{y}_h \mid$ $h = argmin_{h=1}^{l} objFunction(X, Y, \hat{X}, \hat{Y}_{y_j \rightarrow \hat{y}_h}) < objValue$
- $newobjValue \leftarrow objFunction(X, Y, \hat{X}, \hat{Y})$

} **while** $(objValue - newobjValue > threshold)$

algorithm choses the best move of all the rows/columns involved in the corresponding transitive closure along with the selected row/column. Note that after each iteration, all *must-link* and, if available *cannot-link*, constraints are fulfilled.

### 3.5 Alternative approach III: metadata-injection based co-clustering

So far, we have considered two different ways in which metadata can be considered during co-clustering. In the first scheme, metadata is used as a guiding principle, while in the second approach, metadata provides explicit constraints that hold before and after each iteration of the co-clustering process. In this subsection, we will consider a third way to integrate metadata information into the co-clustering process: to combine the initial relationship matrix and the metadata matrix into a single unified matrix which is then subjected to co-clustering under a KL-divergence based optimization function. In other words, given the original relationship matrix $R^{\text{rel}}$ and the metadata matrix $R^{\text{meta}}$ (assuming that it exists), a new matrix, $R^{\text{inject}} = R^{\text{rel}} \oplus_f R^{\text{meta}}$, is created and the information-theoretical co-cluster algorithm is executed, unchanged, with input the matrix $R^{\text{inject}}$. Here, $\oplus_f$ is a matrix combination function, defined as follows.

**Definition 5** (Matrix combination function) Given two domains $X = \{x_1, ...., x_m\}$ and $Y = \{y_1, ..., y_n\}$, the $m \times n$ relationship and metadata matrices, $R^{\text{rel}}$ and $R^{\text{meta}}$, the combination function, $\oplus_f$, is such that for all $1 \leq i \leq m$, $1 \leq j \leq n$

$$r_{ij}^{\text{inject}} = f\left(r_{ij}^{\text{rel}}, r_{ij}^{\text{meta}}\right),$$

for a monotonic combination function $f()$.

In general, $f()$ can be any monotonic function. Hanisch et al. (2002), for example, combine information from gene expression data and biological networks using a domain specific distance function. In this paper, we consider more general combination functions, including *min*(), *max*(), *sum*(), *average*(), and *product*(). The *min*() function, for example, takes a *conservative* view on the correspondences between row and column elements. When $r_{ij}^{\text{rel}}$ and $r_{ij}^{\text{meta}}$ are interpreted as joint probabilities, the *product*() function also takes a conservative meaning (where the correspondence between $x_i$ and $y_j$ holds if it holds under both original relationship *and* metadata matrices). The *max*() and *sum*() functions on the other hand are more *optimistic* in nature; *max*() for example assumes that the stronger correspondence implied by one of the two matrices holds. Note that $f()$ can also impose explicit, user provided, weights to the two information sources (Candan and Li 2001).

## 4 Comparison of the three metadata supported co-clustering approaches

Since the aim of this work is to evaluate and compare different mechanisms for leveraging metadata in co-clustering process, in this section we first provide an application within which we can study the impact of the alternative approaches.

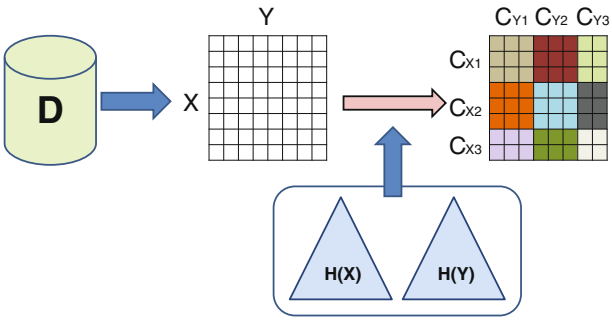



**Fig. 8** Evaluation setting: $X$ and $Y$ are two sets of concepts, the relationship matrix is obtained using the analysis of a document corpus $D$, and two taxonomies, $H(X)$ and $H(Y)$ are provided as metadata

### 4.1 Application: co-clustering-based concept alignment

Let us consider two domains of concepts $X$ and $Y$ and let us assume that we would like to measure the similarities between these concepts based on their co-occurrences in a given document corpus, $D$, and (co-)cluster them based on these similarities. Let us also assume that we are given two taxonomies, $H_X(X, E_X)$ and $H_Y(Y, E_Y)$, representing the background knowledge related to these two domains. Therefore, we can leverage these metadata to *contextualize* the co-clustering process (Fig. 8).

In our evaluation, we use different sets of geographic concepts (each with average 180 concepts) as the $X$ and $Y$ domains. As the document corpus, $D$, we use the 300,000 newspaper articles from the New York Times collection,[1] with over 100,000 unique keywords. The geographic domain taxonomies, $H_X(X, E_X)$ and $H_Y(Y, E_Y)$, are extracted from the DMOZ[2] open directory categorization and have an average depth of 5 and a branching factor of 3.75. Here, the co-occurrences in the corpus is the primary information source, whereas the domain specific metadata (geographic knowledge) provides the secondary, contextual knowledge (Fig. 8). Next we describe how these information are encoded for metadata supported co-clustering.

#### *4.1.1 Encoding of the relationship matrix*

We encode the first of these, the co-occurences in the corpus, in the form of a relationship matrix $R^{\text{rel}}$: i.e, each element $r_{ij}^{\text{rel}}$ is equal to the number of documents in $D$ that contain both concepts $x_i$ and $y_j$.

#### *4.1.2 Encoding of the row- and column-metadata matrices*

Given a domain hierarchy, $H_X$ or $H_Y$, the closeness among the elements in the domain, $X$ or $Y$, can be measured in various ways. Approaches, such as Valtchev and Euzenat (1997), compute dissimilarities between nodes in a given hierarchy using a (weighted) count of edges between the nodes or the average distance from a common ancestor. CP/CV technique, proposed by Kim and Candan (2006), on the other hand

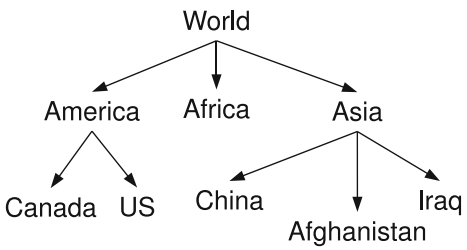
[1] http://archive.ics.uci.edu/ml/datasets/Bag+of+Words.

[2] http://www.dmoz.org

uses a spreading-activation based approach to annotate concept nodes with concept-vectors, which can then be used to measure concept-to-concept similarities. More specifically, authors introduce a concept propagation (CP) scheme, which relies on the structural relationships between concepts implied by the hierarchy, to annotate each concept-node with a concept-vector (CV). Figure 9 provides an example: here, each concept in the taxonomy fragment (containing nine concept nodes) is mapped into a 9-dimensional vector. For example, the root is represented by the vector in which the first component (the one associated to the concept *"world"*), dominates over the others that contribute to the definition of the concepts. The second, third and fourth components reflect the weight of *"Asia"*, *"Africa"* and *"America"* respectively in the semantic characterization of *"world"*, while the remaining components represent the weights of the three descendants of *"Asia"* and of the two descendants of *"America"*. Relying on data from user studies, Kim and Candan (2006) showed that semantic similarities of the concepts can be computed using the cosine similarities of the concept vectors and that such similarity measurements are often in line with human judgments. Therefore, without loss of generality, we rely on the CP/CV technique to construct the row- and column-metadata matrices ($R^{\text{row,meta}}$ and $R^{\text{col,meta}}$) by measuring intra-taxonomy concept similarities.

### *4.1.3 Encoding of the joint metadata matrix*

Unlike the *metadata-driven* and *metadata-constrained* algorithms (which require row- and column-metadata matrices that capture intra-domain relationships between the concepts), the metadata injection requires a joint metadata matrix that captures the *a priori* knowledge of inter-domain relationships among the concepts. Moreover, while the original relationship matrix captures co-occurrences, the joint metadata matrix needs to capture the similarities between the domain elements in terms of the domain taxonomies.




| | world | Asia | Africa | America | Afgh. | Iraq | China | Canada | US |
|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{cv}_{world}$ | 0.450 | 0.169 | 0.141 | 0.158 | 0.018 | 0.018 | 0.018 | 0.021 | 0.021 |
| $\mathbf{cv}_{Asia}$ | 0.052 | 0.469 | 0.006 | 0.006 | 0.156 | 0.156 | 0.156 | 0.0003 | 0.0003 |
| $\mathbf{cv}_{Africa}$ | 0.100 | 0.012 | 0.873 | 0.012 | 0.0006 | 0.0006 | 0.0006 | 0.0007 | 0.0007 |
| $\mathbf{cv}_{America}$ | 0.057 | 0.007 | 0.007 | 0.520 | 0.0003 | 0.0003 | 0.0003 | 0.204 | 0.204 |
| $\mathbf{cv}_{Afgh.}$ | 0.004 | 0.100 | 0.0002 | 0.0002 | 0.872 | 0.012 | 0.012 | 0 | 0 |
| $\mathbf{cv}_{Iraq}$ | 0.004 | 0.100 | 0.0002 | 0.0002 | 0.012 | 0.872 | 0.012 | 0 | 0 |
| $\mathbf{cv}_{China}$ | 0.004 | 0.100 | 0.0002 | 0.0002 | 0.012 | 0.012 | 0.872 | 0 | 0 |
| $\mathbf{cv}_{Canada}$ | 0.006 | 0.0003 | 0.0003 | 0.165 | 0 | 0 | 0 | 0.806 | 0.023 |
| $\mathbf{cv}_{US}$ | 0.006 | 0.0003 | 0.0003 | 0.165 | 0 | 0 | 0 | 0.023 | 0.806 |

**Fig. 9** The concept vectors identified for the taxonomy on the left hand side (example taken from Cataldi et al. 2009)

In this paper, the joint metadata matrix is created by combining information coming from the row- and column-metadata matrices introduced in Section 4.1.2. The idea of this approach is that the relationship between two concepts coming from two different taxonomies can be expressed by combining the corresponding CP/CV vectors. More specifically, the joint metadata matrix $R^{\text{meta}}$ is computed as follow:

$$R^{\text{meta}} = R^{\text{row,meta}} \times U \times R^{\text{col,meta}}$$

where the central matrix U is constructed as follows:

$$u_{ij} = \begin{cases} 1 & \text{if } x_i \in X \text{ and } y_j \in Y \text{ represent geographical concepts with the same label,} \\ 0 & \text{otherwise.} \end{cases}$$

Let us assume that B is a concept shared between row and column metadata. Then, if $R^{\text{row,meta}}$ relates row-concept A to row-concept B and if $R^{\text{col,meta}}$ relates the column-concept B to column-concept C, the combined $R^{\text{meta}}$ matrix will capture this indirect relationship between row-concept A and column-concept C.

#### *4.1.4 Encoding of the must-link constraints*

As introduced in Section 3.4, row/column *must-link* constraints are computed starting from row/column vectors of the corresponding metadata matrices. In this paper row/column vectors are represented by the CP/CV vectors (see Section 4.1.2), thus we define a set of *must-link* constraints by considering the cosine similarity as similarity function:

$$ML_X = \left\{(x_i, x_j) \in X \times X \mid i \neq j \ \wedge \cos\left(\mathbf{cv}(x_i), \mathbf{cv}(x_j)\right) > \theta\right\}$$

where $\mathbf{cv}(k)$ represents the CP/CV vector of the concept $k$ in the given taxonomy.

The *must-link* set $ML_Y$, is similarly defined for column vectors. Table 1 reports information about row and column constraints in the scenario depicted in Section 4.1. Since the *metadata-constrained* algorithm leverages *must-links* by using the corresponding set of transitive closures $TR_X$ and $TR_Y$, Table 1 reports also information about that. As can be seen, the number of *must-link* constraints increases when the threshold value $\theta$ decreases, while the number of transitive closure increases until a specific value of $\theta$ and decreases when the threshold value causes the creation of closures containing an high number of concepts.

**Table 1** Number of row and column constraints with different values of the threshold $\theta$

| $\theta$ | # Row constr. | # Row trans. closures | # Column constr. | # Column trans. closures |
|---|---|---|---|---|
| 0.9 | 0 | 0 | 0 | 0 |
| 0.8 | 12 | 12 | 15 | 15 |
| 0.7 | 12 | 12 | 15 | 15 |
| 0.6 | 33 | 23 | 34 | 25 |
| 0.5 | 36 | 23 | 37 | 23 |
| 0.4 | 66 | 31 | 55 | 28 |
| 0.3 | 93 | 35 | 82 | 33 |
| 0.2 | 106 | 32 | 101 | 24 |
| 0.1 | 185 | 1 | 175 | 1 |

### 4.2 Evaluation metrics

We compare alternative approaches to metadata-supported co-clustering using the following quality and cost measures:

- Information theoretical objective function (KL-divergence) on the relationship matrix measures the impact of the main co-occurrence data (see Section 2.2). The smaller the KL-divergence value is the *better* the resulting clusters are with respect to the input relationship matrix.
- Row- and column-metadata similarities and the average metadata variance of the co-clusters measure how much the resulting co-clusters respect the metadata (see Section 3.2). For each row and column cluster, the agreement with the metadata is measured using the average distance to the common ancestor in the corresponding taxonomy; i.e., if $\hat{x}_g$ is a row or column cluster,

$$\overline{\Delta}_{(\hat{x}_g)} = \frac{\sum_{x_1,x_2 \in \hat{x}_g} d\big(x_1, ca(x_1,x_2)\big) + d\big(x_2, ca(x_1,x_2)\big)}{|\hat{x}_g||\hat{x}_g|},$$

  where $ca(x_1, x_2)$ is the closest common ancestor of $x_1$ and $x_2$ in the taxonomy, then a small $\overline{\Delta}_{(\hat{x}_g)}$ indicates co-clustering which respects the metadata.
- Execution time and the number of moves needed to complete the co-clustering process measure the efficiency of a given algorithm.

Note that the first two measures, KL-divergence and metadata-variance, are often in conflict with each other: without any metadata, co-clustering algorithms can more easily reduce the KL-divergence, but the results would not reflect the background knowledge provided by the metadata. Therefore, in addition to reporting the KL-divergence and metadata-variance values individually, we also report a score (Aslandogan et al. 1995) that combines the KL-divergence and metadata based performances of the algorithms:

$$F_\beta = \frac{\big(1+\beta^2\big) * \big(KL\_perf * metadata\_perf\big)}{\big(\beta^2 * KL\_perf + metadata\_perf\big)}$$

Here, higher $\beta$ value indicates more emphasis on metadata and lower $\beta$ value indicates less emphasis on metadata ($\beta = 1$ indicates equal emphasis). Note that, the higher the $F_\beta$ score of a given algorithm is, the better are its *combined* KL-divergence and metadata-variance behavior.

In order to improve charts' readability, input data have been normalized with respect to the reference values $KL\_perf_{\text{default}}$ and $metadata\_perf_{\text{default}}$, corresponding to the performances of the default, *no-metadata* approach:

$$KL\_perf_{\text{normalized}} = 1 - \frac{KL\_perf}{\alpha * KL\_perf_{\text{default}}}$$

and

$$metadata\_perf_{\text{normalized}} = 1 - \frac{metadata\_perf}{\alpha * metadata\_perf_{\text{default}}}$$

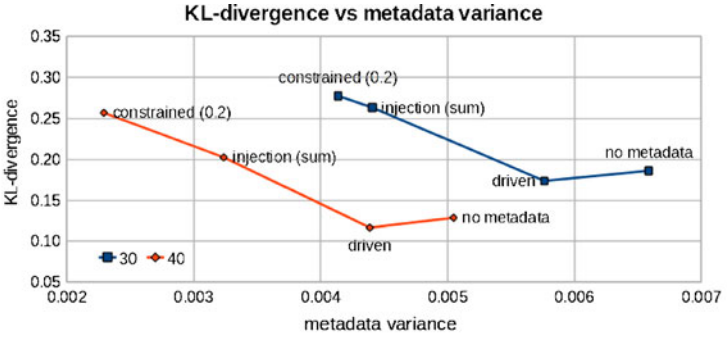
In these experiments, we use $\alpha = 2.5$; thus, in the $F_\beta$ charts, the default *no-metadata* approach appears as a fixed horizontal line with height 0.6.

### 4.3 Discussion of the results

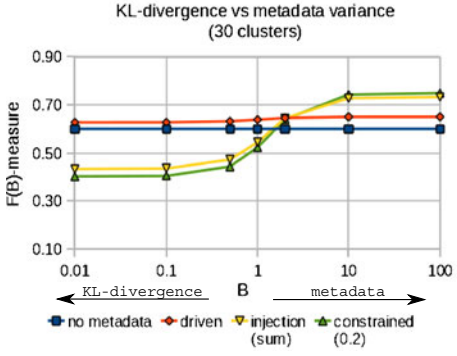
For each of the three approaches experimented within this paper, we varied the corresponding parameters to observe their impacts on the results. For the *metadata-driven* algorithm presented in Section 3.3, we varied $c$; i.e., the number of top information theoretical alternatives considered for metadata optimization (default is $c = \infty$; i.e., no limit on the number of alternatives). For the *metadata-constrained* algorithm presented in Section 3.4, we varied the similarity threshold, $\theta$, (default is $\theta = 0.2$). For *metadata-injection* algorithm in Section 3.5, we varied the underlying combination functions (default is the *sum()* function).

#### *4.3.1 Main results*

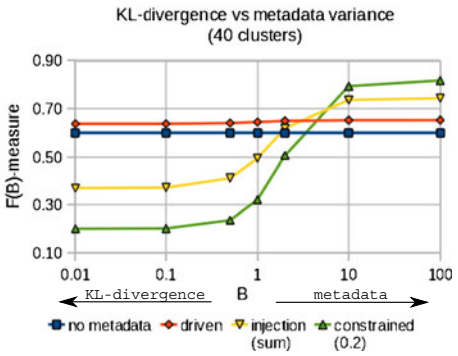
In Fig. 10a, we compare the KL-divergence based objective function and metadata variance for the alternative approaches, for 30 and 40 target row and column clusters. In this scatter-plot, we can note that the differences in the number of clusters do not affect the relative behaviors of the algorithms. As expected, all the metadata-based algorithms help reduce metadata variance of co-clusters, with the metadata-constrained one providing the highest reduction. However, in the case of metadata-constrained and metadata-injection based algorithms, this reduction comes with a




(a) KL-divergence vs. metadata-variance scatter-plot




(b) $F_\beta$ plot for 30 clusters




(c) $F_\beta$ plot for 40 clusters

**Fig. 10** KL-divergence vs. metadata variance for different algorithms

relatively significant increase in KL-distances with respect to the case where no metadata is considered.

Figure 10b and c plot the same results in terms of the combined $F_\beta$ values for different $\beta$ weights (from KL-divergence being 100 times more important to metadata-variance being 100 times more important):

- A key observation here is that *there is no single $\beta$ value for which no metadata approach is more desirable*; for all $\beta$ values in this range, the *metadata-driven* strategy provides a better combined score.
- A second observation is that metadata-constrained and metadata-injection based schemes become more attractive only when the importance of metadata becomes more dominant. In particular, metadata-injection becomes the most effective approach quickly when the importance of metadata exceeds the importance of the relationship matrix, while the metadata-constrained approach becomes more desirable, when the metadata is significantly ($> 10\times$) more important than the relationship matrix.

Note that since metadata is inherently secondary to the relationship matrix, these results indicate that (unless there is a strong reason to impose respect to the metadata) the metadata-driven approach is the most effective approach among all alternatives (including ignoring the metadata). This is because, as shown in Fig. 10a, the *metadata-driven* scheme not only reduced the metadata variance, but also the KL-divergence; and this indicates that when handled properly, consideration of background knowledge (i.e., the metadata) can help improve co-clustering results.

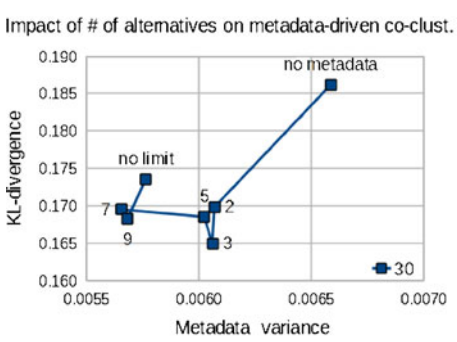
In terms of the execution time (Fig. 11), the *metadata-constrained* scheme, which has to compute constraints to be imposed, and the *metadata-driven* scheme, which has to recompute centroids, observe the biggest jumps. In contrast, the *metadata-injected* scheme shows a more similar execution time respect to the original co-clustering algorithm. In terms of moves, the metadata-driven scheme (which is able to optimize both KL-distance and metadata variance) sees the highest jump. The number of moves for metadata-injection is only slightly higher than the base scheme which does not use metadata, while the metadata-constrained scheme represents an halfway point.

### *4.3.2 Impact of the number of candidates (c) on metadata-driven co-clustering*

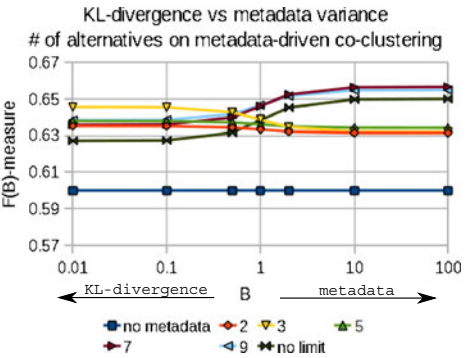
Figure 12a shows the impact of increasing the number of alternatives considered by the metadata-driven co-clustering scheme. As can be seen here, while increasing the number of candidates initially helps in terms of metadata variance, after a while the benefits wore off. One interesting aspect to note is that just adding one more alternative (i.e., considering the top-2 candidates instead of the top-most one)

**Fig. 11** Overview of the execution time comparison between different algorithms

| Time (sec) | | | | |
|---|---|---|---|---|
| #cl. | **w/o meta.** | **m.-driven** | **m.-injected** | **m.-const.** |
| 30 | 1.79 | 2.70 | 2.02 | 8.85 |
| 40 | 3.84 | 6.01 | 4.44 | 20.37 |
| Moves | | | | |
| #cl. | **w/o meta.** | **m.-driven** | **m.-injected** | **m.-const.** |
| 30 | 298.6 | 444.5 | 306.2 | 349.2 |
| 40 | 277.7 | 445.0 | 288.1 | 317.4 |




(a)




(b)

| Time (sec) | | | | | |
|---|---|---|---|---|---|
| #cl. | **w/o meta.** ($c = 1$) | $c = 2$ | $c = 5$ | $c = 9$ | **no limit** |
| 30 | 1.79 | 2.24 | 2.35 | 2.53 | 2.70 |
| Moves | | | | | |
| #cl. | **w/o meta.** ($c = 1$) | $c = 2$ | $c = 5$ | $c = 9$ | **no limit** |
| 30 | 298.6 | 329.3 | 386.7 | 417.9 | 444.5 |

(c) Complexity

**Fig. 12** Impact of the number of candidates ($c$) on metadata-driven co-clustering

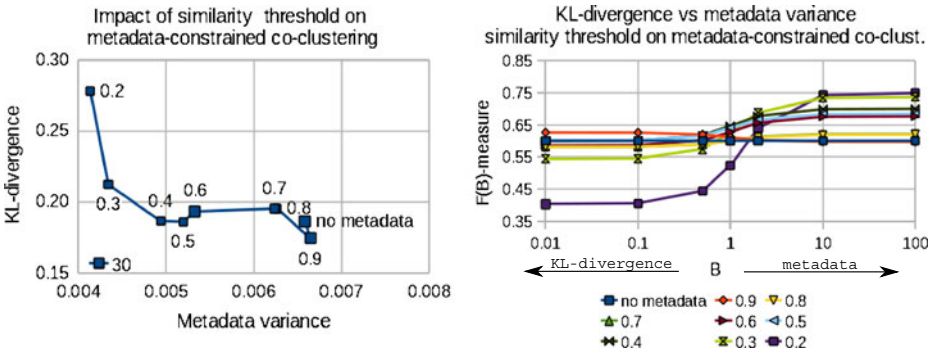
is sufficient in helping reduce the *KL-distance* (which is an information theoretic measure not directly optimized by the metadata).

Figure 12b confirms this behavior: as can be seen here, while considering more than one candidate is sufficient in obtaining significantly better performance than no-metadata case; moreover, for high values of $\beta$, the best result is obtained when $c = 7$ is imposed on the alternatives. As can be seen in Fig. 12a, the KL-divergence value does not differ (except for $c = 3$) when the number of considered candidates are limited ($c = 2, 5, 7, 9$), while $c = 7, 9, \infty$ provide the best results for metadata variance.

Figure 12c shows that the use of metadata increases the number of moves needed to converge on a result proportionally with the increase of the number of candidates, which may partially explain why the KL-divergence itself improves when using metadata. This, however, cannot be the only factor, because, as one would intuitively expect, the co-occurrences in the document corpus is related with the spatial relationships between the geographic concepts captured by the metadata.

### *4.3.3 Impact of the similarity threshold ($\theta$) on metadata-constrained co-clustering*

Figure 13a shows the impact of varying the underlying parameter for the metadata-constrained co-clustering. The drop in metadata-variance increases when the similarity threshold considered for the creation of the constraints is relaxed (i.e., when the number of constraints is increased): the value of $\theta = 0.2$ provides the best results for what concerns the metadata variance. Moreover, the KL-divergence value increases when the value of $\theta$ decreases, with a significant change for $\theta = 0.2$. This behavior is paralleled by the execution time and number of moves results reported in Fig. 13c:



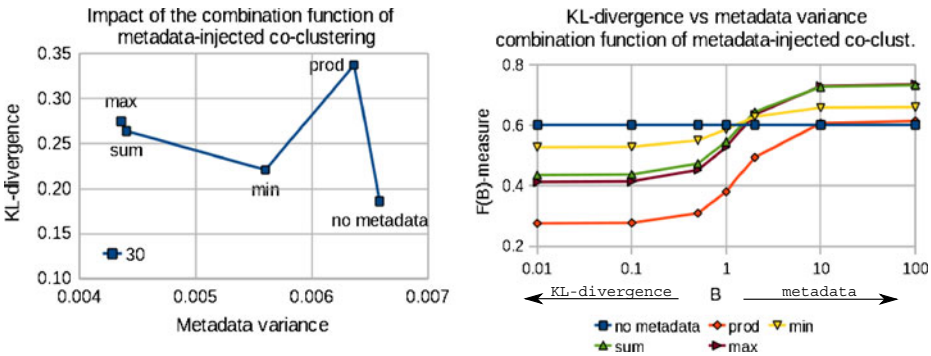

(a) (b)

| Time (sec) | | | | |
|---|---|---|---|---|
| #cl. | **w/o meta.** | $cos(\theta) = 0.9$ | $cos(\theta) = 0.6$ | $cos(\theta) = 0.3$ |
| 30 | 1.79 | 2.04 | 2.91 | 4.36 |
| Moves | | | | |
| #cl. | **w/o meta.** | $cos(\theta) = 0.9$ | $cos(\theta) = 0.6$ | $cos(\theta) = 0.3$ |
| 30 | 298.6 | 323.4 | 367.2 | 379.0 |

(c) Complexity

**Fig. 13** Impact of the similarity threshold ($\theta$) on metadata-constrained co-clustering

they both increase when the number of constraints increases. Figure 13b shows that metadata-constraint becomes the most effective approach quickly when the importance of metadata exceeds the importance of the relationship matrix.




(a) (b)

| Time (sec) | | | | | |
|---|---|---|---|---|---|
| #cl. | **w/o meta.** | **prod** | **min** | **max** | **sum;avg** |
| 30 | 1.79 | 1.31 | 1.98 | 1.82 | 2.02 |
| Moves | | | | | |
| #cl. | **w/o meta.** | **prod** | **min** | **max** | **sum;avg** |
| 30 | 298.6 | 205.7 | 324.5 | 307.5 | 306.2 |

(c) Complexity

**Fig. 14** Impact of the combination function on metadata-injected co-clustering


| Selected concept: India | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| bangalore | 0.948 | 0.488 |
| calcutta | 0.9291 | 0.488 |
| afghanistan | 0.5625 | 3.52E-3 |
| iraq | 0.5179 | 3.52E-3 |
| warsaw | 0.402 | 3.45E-7 |
| vienna | 0.3405 | 6.47E-7 |
| average | 0.6167 | 0.1638 |

(a) pure cooccurrence based


| Selected concept: India | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| bangalore | 0.948 | 0.488 |
| calcutta | 0.9291 | 0.488 |
| burma | 0.804 | 3.52E-3 |
| bombay | 0.6323 | 0.488 |
| north korea | 0.6103 | 6.8E-4 |
| china | 0.5821 | 3.52E-3 |
| taiwan | 0.5032 | 3.52E-3 |
| south korea | 0.4669 | 6.8E-4 |
| average | 0.6845 | 0.1845 |

(b) Metadata-driven

**Fig. 15** Graphical and tabular representations of row-clusters containing the concept *India* obtained by pure *cooccurrence* based and *metadata-driven* co-clustering

### *4.3.4 Impact of the combination function on metadata-injection based co-clustering*

Experiments with different combination functions confirm that metadata-injection always provides a descrease in metadata variance and an increase in KL-divergence


| Selected concept: India | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| bangalore | 0.948 | 0.488 |
| calcutta | 0.9291 | 0.488 |
| bombay | 0.6323 | 0.488 |
| north korea | 0.6103 | 6.8E-4 |
| south korea | 0.4669 | 6.8E-4 |
| korea | 0.4097 | 2.82E-3 |
| average | 0.666 | 0.2447 |

(a) Metadata-injection


| Selected concept: India | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| bangalore | 0.948 | 0.488 |
| calcutta | 0.9291 | 0.488 |
| burma | 0.804 | 3.52E-3 |
| bombay | 0.6323 | 0.488 |
| average | 0.8283 | 0.3669 |

(b) Metadata-constrained

**Fig. 16** Graphical and tabular representations of row-clusters containing the concept *India* obtained by pure *metadata-injection* and *metadata-constrained* co-clustering


| Selected concept: Mexico City | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| cancun | 0.8313 | 0.0815 |
| mexico | 0.7666 | 0.4878 |
| veracruz | 0.7032 | 0.0815 |
| argentina | 0.4155 | 2.82E-6 |
| chile | 0.3806 | 2.85E-6 |
| cuba | 0.3705 | 7.78E-4 |
| buenos aires | 0.3382 | 2E-8 |
| santiago | 0.3292 | 3.29E-7 |
| la habana | 0.2818 | 1.6E-4 |
| pampas | 0.195 | 1.27E-7 |
| average | 0.4612 | 0.0652 |

(a) pure cooccurrence based


| Selected concept: Mexico City | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| cancun | 0.8313 | 0.0815 |
| mexico | 0.7666 | 0.4878 |
| veracruz | 0.7032 | 0.0815 |
| jamaica | 0.5556 | 1.05E-3 |
| united states | 0.4796 | 8.51E-4 |
| average | 0.6673 | 0.1305 |

(b) Metadata-driven

**Fig. 17** Graphical and tabular representations of row-clusters containing the concept *Mexico City* obtained by pure *cooccurrence* based and *metadata-driven* co-clustering

value. As shown in Fig. 14a, the highest drop in metadata variance is obtained using the *optimistic* (or *disjunctive*) combination function, *max*(). This is followed very closely by the *sum*() (or equivalently[3]) *avg*() combination function. The *pessimistic* (or *conjunctive*) combination functions *min*() and *product*() are not competitive in terms of gain in the metadata-based quality. Moreover, the losses in the information-theoretical objective function is more evident for *product*(). Figure 14b confirms

[3]The matrix is re-normalized after the application of the combination function to ensure that information-theoretic co-clustering, which treats the values in the matrix as probability distributions, can be applied. Due to this renormalization, the combination function *sum*() is equivalent to the *average*() (the two functions would differ for a scaling factor 2, which is absorbed by re-normalization).

the fact that, among the different combination functions, the best performance is obtained by the *max*() and *sum*().

Note that, as shown in Fig. 14c, the metadata benefits of the *sum*() (or *avg*()) combination function comes with only a modest increase in the number of moves (indeed, in terms of the execution time, in the experiments, *max*() based metadata-injection took slightly more than the base scheme without metadata). Moreover, the drop in execution times for what concerns *product*() is reflected to a drop in the average number of moves.

### *4.3.5 Sample clusters in the geographical application domain*

Finally, in Figs. 15, 16, 17, 18, 19 and 20 we present, sample clustering results obtained using the different techniques introduced in this paper: these figures provides sample cases within the application domain to help observe the impact of contextual metadata information (the geographical relationships among concepts) used in addition to the primary co-occurrence data reflecting co-occurrences of the geographic concepts in a collection of New York Times news articles (note that there are 155 geographic concepts and the target number of resulting geographic row-clusters is set to 30). In these figures, we consider three geographic concepts (*India*, *Mexico City*, and *Rio de Janeiro*) and report the resulting row-clusters that contain each selected concept. On the map-based graphical representation, the selected concept is connected to the concepts in its clusters by lines, where the thickness of the line represents the


| Selected concept: Mexico City | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| mexico | 0.7666 | 0.4878 |
| venezuela | 0.4287 | 3.96E-6 |
| colombia | 0.4037 | 2.87E-6 |
| average | 0.533 | 0.1626 |

(a) Metadata-injection


| Selected concept: Mexico City | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| cancun | 0.8313 | 0.0815 |
| mexico | 0.7666 | 0.4878 |
| veracruz | 0.7032 | 0.0815 |
| average | 0.7671 | 0.2169 |

(b) Metadata-constrained

**Fig. 18** Graphical and tabular representations of row-clusters containing the concept *Mexico City* obtained by pure *metadata-injection* and *metadata-constrained* co-clustering


| Selected concept: Rio de Janeiro | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| sao paulo | 0.6528 | 0.1828 |
| paraguay | 0.5763 | 1.2E-3 |
| argentina | 0.5274 | 8.72E-4 |
| brasil | 0.4834 | 0.6375 |
| buenos aires | 0.4629 | 1.83E-5 |
| chile | 0.4455 | 8.85E-4 |
| santiago | 0.3765 | 1.93E-4 |
| south africa | 0.3613 | 2.17E-8 |
| pampas | 0.3246 | 7.86E-5 |
| johannesburg | 0.3213 | 4.63E-10 |
| patagonia | 0.3071 | 1.03E-4 |
| cape town | 0.2742 | 4.63E-10 |
| siberia | 0.2722 | 2.06E-9 |
| pretoria | 0.2192 | 4.63E-10 |
| average | 0.4003 | 0.0588 |

(a) pure cooccurrence based


| Selected concept: Rio de Janeiro | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| sao paulo | 0.6528 | 0.1828 |
| montevideo | 0.5947 | 1.2E-3 |
| paraguay | 0.5763 | 1.2E-3 |
| argentina | 0.5274 | 8.72E-4 |
| brasil | 0.4834 | 0.6375 |
| south america | 0.476 | 0.0915 |
| buenos aires | 0.4629 | 1.83E-5 |
| peru | 0.4459 | 8.96E-4 |
| chile | 0.4455 | 8.85E-4 |
| ecuador | 0.4307 | 1.2E-3 |
| colombia | 0.4293 | 8.85E-4 |
| venezuela | 0.4261 | 1.2E-3 |
| santiago | 0.3765 | 1.93E-4 |
| south africa | 0.3613 | 2.17E-8 |
| bogota | 0.3443 | 1.03E-4 |
| johannesburg | 0.3213 | 4.63E-10 |
| patagonia | 0.3071 | 1.03E-4 |
| cuzco | 0.2791 | 6.31E-5 |
| cartagena | 0.2784 | 1.03E-4 |
| cape town | 0.2742 | 4.63E-10 |
| machu picchu | 0.2626 | 6.31E-5 |
| pretoria | 0.2192 | 4.63E-10 |
| average | 0.408 | 0.0419 |

(b) Metadata-driven

**Fig. 19** Graphical and tabular representations of row-clusters containing the concept *Rio de Janeiro* obtained by pure *cooccurrence* based and *metadata-driven* co-clustering

strength of co-occurrence similarity between the connected concepts (i.e. cosine similarity between the cooccurrences row-vectors) in the co-occurrence relationship matrix, indicating that the two concepts co-occur with similar concepts in the article collection. The tables on the right side, on the other hand, report these co-occurrence similarity scores as well as the geographical similarity values (obtained using the metadata) between the selected concept and the other concepts in the cluster.

As these sample results show, when metadata is used, row-clusters are more geographically sound. As expected, the metadata-constrained approach provides the geographically tightest clusters. Metadata-driven and metadata-injected approaches provide results that are in between the two extremes of no-metadata and metadata-contrained schemes. This is also quantitatively verified in Table 2 which reports that


| Selected concept: Rio de Janeiro | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| sao paulo | 0.6528 | 0.1828 |
| montevideo | 0.5947 | 1.2E-3 |
| paraguay | 0.5763 | 1.2E-3 |
| brasil | 0.4834 | 0.6375 |
| south america | 0.476 | 0.0915 |
| ecuador | 0.4307 | 1.2E-3 |
| venezuela | 0.4261 | 1.2E-3 |
| cuba | 0.3512 | 4.99E-6 |
| bahamas | 0.297 | 6.93E-6 |
| la habana | 0.2167 | 5.54E-7 |
| average | 0.4505 | 0.0917 |

(a) Metadata-injection


| Selected concept: Rio de Janeiro | | |
|---|---|---|
| **concept** | **cooccur. sim.** | **geo. sim.** |
| sao paulo | 0.6528 | 0.1828 |
| montevideo | 0.5947 | 1.2E-3 |
| paraguay | 0.5763 | 1.2E-3 |
| argentina | 0.5274 | 8.72E-4 |
| brasil | 0.4834 | 0.6375 |
| buenos aires | 0.4629 | 1.83E-5 |
| chile | 0.4455 | 8.85E-4 |
| ecuador | 0.4307 | 1.2E-3 |
| santiago | 0.3765 | 1.93E-4 |
| pampas | 0.3246 | 7.86E-5 |
| patagonia | 0.3071 | 1.03E-4 |
| average | 0.4711 | 0.0751 |

(b) Metadata-constrained

**Fig. 20** Graphical and tabular representations of row-clusters containing the concept *Rio de Janeiro* obtained by pure *metadata-injection* and *metadata-constrained* co-clustering

the *average distance to the closest common geographical ancestor* (see Section 4.2) in the input geographical taxonomy is largest when no metadata information is used and drops to its minimum in the case of metadata-constrained scheme.

Note that an interesting result in Figs. 15–20 is that when considering the metadata, not only the average geographical similarity in the resulting clusters have improved, but also the co-occurrence similarities have seen improvements. This is further verified in Table 3a and b: let $mean\_row\_sim(c)$ be the average co-occurrence similarity of the concept $c$ to all other concepts in the same row-cluster as $c$; the table reports the averages of all $mean\_row\_sim$ values for all concepts for different co-clustering techniques. As can be seen in this table, for both 30 and 40 target row cluster cases, the averages improve when metadata is used in co-clustering and they are highest for the metadata constrained co-clustering strategy. This reflects the fact that (as one would expect) co-occurrences in the news articles are correlated with the geographical relationships between the geographical concepts and, if selected

**Table 2** Average (row-)cluster common ancestor distances for different numbers of clusters

| #cl. | W/o meta. | M.-driven | M.-injected | M.-const. |
|---|---|---|---|---|
| 10 | 4.41 | 3.42 | 3.74 | 3.03 |
| 20 | 3.23 | 2.43 | 2.43 | 2.01 |
| 30 | 2.38 | 1.99 | 1.72 | 1.47 |

**Table 3** Average value and standard deviation of co-occurrence similarities in the different approaches

| | W/o metadata | M.-driven | M.-injected | M.-constrained |
|---|---|---|---|---|
| (a) Row clusters: 30 | | | | |
| Average | 0.4875 | 0.5047 | 0.5137 | 0.5412 |
| Standard dev. | 0.1420 | 0.1386 | 0.1061 | 0.1177 |
| (b) Row clusters: 40 | | | | |
| Average | 0.5144 | 0.5286 | 0.5503 | 0.5792 |
| Standard dev. | 0.1502 | 0.1485 | 0.1132 | 0.1244 |

carefully, contextual metadata may in fact help improve the qualities of the resulting clusters also based on the primary dominant relationship (in this case co-occurrence) as well.

## 5 Conclusions

In this paper, we proposed and evaluated three alternative strategies (namely *metadata-driven*, *metadata-constrained*, and *metadata-injected* co-clustering) for enriching the co-clustering process with metadata about the rows and columns of the given relationship matrix. Experimentals show that it is possible to leverage available metadata in obtaining contextually-relevant co-clusters. In particular, while the *metadata-constrained* approach provides biggest gains in terms of metadata-preservation, this comes with a relatively high loss in information theoretic objective function. The *metadata-injected* co-clustering scheme provides a reasonable trade-off between preservation of the information theoretic co-clustering quality and enforcement of structures implied by the available metadata especially in scenarios in which there is the need to consider the metadata information more important with respect to the objective function value. In general, the *metadata-driven* scheme represents a good choice in all scenarios, since it improves *both* metadata and information theoretic objective functions and it always outperforms the original co-clustering algorithm.

## References

Alp Aslandogan, Y., Thier, C., Yu, C. T., Liu, C., & Nair, K. R. (1995). Design, implementation and evaluation of score (a system for content based retrieval of pictures). In *ICDE '95: Proceedings of the eleventh international conference on data engineering* (pp. 280–287). Washington: IEEE.

Baier, D., Gaul, W., & Schader, M. (1997). Two-mode overlapping clustering with applications to simultaneous benefit segmentation and market structuring. In R. Klar, & O. Opitz (Eds.), *Classification and knowledge organization: Recent advances and applications* (pp. 557–566). Springer.

Banerjee, A., Dhillon, I., Ghosh, J., Merugu, S., & Modha, D. S. (2007). A generalized maximum entropy approach to bregman co-clustering and matrix approximation. *Journal of Machine Learning Research, 8*, 1919–1986.

Basu, S., Banerjee, A., & Mooney, R. J. (2002). Semi-supervised clustering by seeding. In *ICML'02: Proceedings of the 9th international conference on machine learning* (pp. 27–34). San Francisco: Morgan Kaufmann.

Basu, S., Bilenko, M., & Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 59–68). New York: ACM.

Bilenko, M., Basu, S., & Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *ICML* (pp. 81–88).

Bilenko, M., & Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 39–48). New York: ACM.

Bishop, C. M. (2006). *Pattern recognition and machine learning (Information science and statistics)*. New York: Springer.

Candan, K. S., Cataldi, M., Sapino, M. L., & Schifanella, C. (2008). Structure- and extension-informed taxonomy alignment. In *Proceedings of the 4th international VLDB workshop on ontology-based techniques for databases in information systems and knowledge systems, ODBIS 2008, Auckland, New Zealand, 23 August 2008, co-located with the 34th international conference on very large data bases* (pp. 1–8).

Candan, K. S., & Li, W.-S. (2001). On similarity measures for multimedia database applications. *Knowledge and Information Systems, 3*(1), 30–51.

Cataldi, M., Schifanella, C., Candan, K. S., Sapino, M. L., & Di Caro, L. (2009). Cosena: A context-based search and navigation system. In *The first international acm conference on management of emergent digital ecosystems (MEDES)*. Lyon: ACM.

Chen, Y., Dong, M., & Wan, W. (2009). Image co-clustering with multi-modality features and user feedbacks. In *Proceedings of the seventeen ACM international conference on multimedia, MM '09* (pp. 689–692). New York: ACM. ISBN 978-1-60558-608-3. doi:10.1145/1631272.1631389. URL http://doi.acm.org/10.1145/1631272.1631389.

Chen, Y., Wang, L., & Dong, M. (2008). A matrix-based approach for semi-supervised document co-clustering. In *Proceeding of the 17th ACM conference on information and knowledge management, CIKM '08* (pp. 1523–1524). New York: ACM. ISBN 978-1-59593-991-3.

Chen, Y., Wang, L., & Dong, M. (2010). Non-negative matrix factorization for semisupervised heterogeneous data coclustering. *IEEE Transactions on Knowledge and Data Engineering, 22*, 1459–1474. ISSN 1041-4347. doi:10.1109/TKDE.2009.169.

Cheng, Y., & Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the eighth international conference on intelligent systems for molecular biology* (pp. 93–103). AAAI Press.

Cho, H., Dhillon, I. S., Guan, Y., & Sra, S. (2004). Minimum sum-squared residue co-clustering of gene expression data. In M. W. Berry, U. Dayal, C. Kamath, & D. B. Skillicorn (Eds.), *SDM*. SIAM.

Demiriz, A., Bennett, K. P., & Embrechts, M. J. (1999). Semi-supervised clustering using genetic algorithms. In *Artificial neural networks in engineering (ANNIE-99)* (pp. 809–814). ASME Press.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 269–274). New York: ACM.

Dhillon, I. S., Subramanyam, M., & Modha Dharmendra, S. (2003). Information-theoretic co-clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 89–98). New York: ACM.

Freitag, D. (2004). Trained named entity recognition using distributional clusters. In *Proceedings of the conference on empirical methods in natural language processing, EMNLP* (pp. 262–269). Barcelona, Spain.

Gao, B., Liu, T.-Y., & Ma, W.-Y. (2006). Star-structured high-order heterogeneous data co-clustering based on consistent information theory. In *Proceedings of the 6th IEEE international conference on data mining (ICDM 2006), 18–22 December 2006, Hong Kong, China* (pp. 880–884). IEEE Computer Society.

Gao, B., Liu, T.-Y., Zheng, X., Cheng, Q.-S., & Ma, W.-Y. (2005). Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *KDD '05: Proceedings of the 11th ACM SIGKDD int. conference on knowledge discovery in data mining* (pp. 41–50). New York: ACM.

Gaul, W., & Schader, M. (1996). A new algorithm for two-mode clustering. In H. Hermann, & W. Polasek (Eds.), *Data analysis and information systems* (pp. 15–23). Springer.

George, T., & Merugu, S. (2005). A scalable collaborative filtering framework based on co-clustering. In *ICDM '05: Proceedings of the fifth IEEE international conference on data mining* (pp. 625–628). Washington: IEEE.

Hanisch, D., Zien, A., Zimmer, R., & Lengauer, T. (2002). Co-clustering of biological networks and gene expression data. In *ISMB (Supplement of bioinformatics)* (pp. 145–154).

Hartigan, J. A. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association, 67*(337), 123–129.

Hofmann, T., & Puzicha, J. (1999). Latent class models for collaborative filtering. In *Proceedings of the sixteenth international joint conference on artificial intelligence, IJCAI '99* (pp. 688–693). San Francisco: Morgan Kaufmann. ISBN 1-55860-613-0.

Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., & Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st national conference on artificial intelligence* (Vol. 1, pp. 381–388). AAAI Press. ISBN 978-1-57735-281-5.

Kim, J. W., & Candan, K. S. (2006). Cp/cv: Concept similarity mining without frequency information from domain describing taxonomies. In *CIKM '06* (pp. 483–492).

Klein, D., Kamvar, S. D., & Manning, C. D. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *ICML '02: Proceedings of the nineteenth international conference on machine learning* (pp. 307–314). San Francisco: Morgan Kaufmann.

Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review, 51*(3), 455–500.

Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems 13, papers from neural information processing systems (NIPS) 2000, Denver, CO, USA* (pp. 556–562). MIT Press.

Li, H., & Abe, N. (1998). Word clustering and disambiguation based on co-occurrence data. In *Proceedings of the 17th international conference on computational linguistics* (pp. 749–755). Morristown: Association for Computational Linguistics.

Long, B., Zhang, Z. M., Wú, X., & Yu, P. S. (2006). Spectral clustering for multi-type relational data. In *Proceedings of the 23rd international conference on machine learning, ICML '06* (pp. 585–592). New York: ACM. ISBN 1-59593-383-2. doi:10.1145/1143844.1143918. URL http://doi.acm.org/10.1145/1143844.1143918.

Long, B., Zhang, Z. M., & Yu, P. S. (2007). A probabilistic framework for relational clustering. In *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, KDD '07* (pp. 470–479). New York: ACM. ISBN 978-1-59593-609-7. doi:10.1145/1281192.1281244. URL http://doi.acm.org/10.1145/1281192.1281244.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17*(4), 395–416.

Ma, H., Zhao, W., Tan, Q., & Shi, Z. (2010). Orthogonal nonnegative matrix tri-factorization for semi-supervised document co-clustering. In M. Zaki, J. Yu, B. Ravindran, & V. Pudi (Eds.), *Advances in knowledge discovery and data mining. Lecture Notes in Computer Science* (Vol. 6119, pp. 189–200). Berlin: Springer.

Madeira, S. C., & Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1*(1), 24–45.

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems 14* (pp. 849–856). MIT Press.

Pensa, R. G., & Boulicaut, J.-F. (2008). Constrained co-clustering of gene expression data. In *Proceedings of the SIAM international conference on data mining, SDM 2008, 24–26 April 2008, Atlanta, Georgia, USA* (pp. 25–36). SIAM.

Ruiz, C., Spiliopoulou, M., & Ruiz, E. M. (2007). C-dbscan: Density-based clustering with constraints. In A. An, J. Stefanowski, S. Ramanna, C. J. Butz, W. Pedrycz, & G. Wang (Eds.), *RSFDGrC. LNCS* (Vol. 4482, pp. 216–223). Springer.

Shan, H., & Banerjee, A. (2008). Bayesian co-clustering. In *Proceedings of the 2008 eighth IEEE international conference on data mining* (pp. 530–539). Washington: IEEE Computer Society. ISBN 978-0-7695-3502-9.

Song, Y., Pan, S., Liu, S., Wei, F., Zhou, M. X., & Qian, W. (2010). Constrained coclustering for textual documents. In M. Fox, & D. Poole (Eds.), *AAAI*. AAAI Press.

Struyf, J., & Dzeroski, S. (2007). Clustering trees with instance level constraints. In J. N. Kok, J. Koronacki, R. López de Mántaras, S. Matwin, D. Mladenic, & A. Skowron (Eds.), *ECML. LNCS* (Vol. 4701, pp. 359–370). Springer.

Tucker, L. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika, 31*(3), 279–311.

Valtchev, P., & Euzenat, J. (1997). Dissimilarity measure for collections of objects and values. In X. Liu, P. R. Cohen, & M. R. Berthold (Eds.), *IDA. LNCS* (Vol. 1280, pp. 259–272). Springer.

Vichi, M. (2001). Double k-means clustering for simultaneous classification of objects and variables. In *Advances in classification and data analysis* (pp. 43–52). Springer.

Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In C. E. Brodley, & A. P. Danyluk (Eds.), *ICML* (pp. 577–584). Morgan Kaufmann.

Xing, E. P., Ng, A. Y., Jordan, M. I., & Russell, S. (2002). Distance metric learning, with application to clustering with side-information. *Advances in neural information processing systems* (pp. 505–512). MIT Press

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference* (pp. 267–273). New York: ACM.