Few-to-few Cross-domain Object Matching

Aditya Jitta ADITYA.JITTA@CS.HELSINKI.FI
Arto Klami ARTO.KLAMI@CS.HELSINKI.FI
Helsinki Institute for Information Technology HIIT, Department of Computer Science
University of Helsinki, Helsinki (Finland)

Abstract

Cross-domain object matching refers to the task of inferring unknown alignment between objects in two data collections that do not have a shared data representation. In recent years several methods have been proposed for solving the special case that assumes each object is to be paired with exactly one object, resulting in a constrained optimization problem over permutations. A related problem formulation of cluster matching seeks to match a cluster of objects in one data set to a cluster of objects in the other set, which can be considered as many-to-many extension of cross-domain object matching and can be solved without explicit constraints. In this work we study the intermediate region between these two special cases, presenting a range of Bayesian inference algorithms that work also for few-to-few cross-domain object matching problems where constrained optimization is necessary but the optimization domain is broader than just permutations.

Keywords: cross-domain object matching; Bayesian canonical correlation analysis; microclustering; integer programming.

1. Introduction

Yamada and Sugiyama (2011) define cross-domain object matching (CDOM) as the task of inferring an unknown alignment of objects in two parallel data collections, without assumptions on the relationship between the objects or their feature representations. It can be formulated as learning a match for a bipartite graph where every object corresponds to a node, and the objects are represented by some features that are shared by nodes within each set but not across the sets. In other words, we are simply given two (typically) real-valued data matrices $\mathbf{X} \in \mathbb{R}^{D_x \times N}$ and $\mathbf{Y} \in \mathbb{R}^{D_y \times N}$ and the knowledge that each column in $\mathbf{X}$ can be paired with some column in $\mathbf{Y}$. For example, the objects in $\mathbf{X}$ could be words in one language and the objects in $\mathbf{Y}$ words in another language, represented by their native contexts that do not share any common features (Haghighi et al., 2008). The problem formulation extends that of record linkage or entity resolution by allowing arbitrary feature representations; typical record linkage solutions (Steorts et al., 2016) assume that both data matrices share (noisy or corrupted) features or that some approximative distance measure is provided across the sets.

The seemingly impossible problem of learning the alignment without such a distance measure can be solved by pairing the columns of $\mathbf{Y}$ with those of $\mathbf{X}$ such that the pairing results in maximal statical dependency. Haghighi et al. (2008) developed matching canonical correlation analysis (CCA) for solving the problem, Klami (2013) used a latent variable model building on the Bayesian interpretation of CCA, Quadrianto et al. (2010) minimized

a kernelized independence measure, and Djuric et al. (2012) provided a convex formulation of their work. Although the technical details of the models differ greatly, the key element is that they perform constrained optimization over the space of permutations, solving a task that is equivalent to learning a structure of a undirected network under the constraint that all edges are between the two sets and every node is constrained to have exactly one edge.

While the strict assumption of *one-to-one matching* helps in solving the problem by pruning out majority of possible structures, it is unrealistic in practical applications. For example, when matching words in different languages we know that typical objects in $\mathbf{X}$ should be paired with multiple objects in $\mathbf{Y}$ because of synonyms and multiple equivalent translations. Iwata et al. (2013, 2016) recognized this limitation and proposed models for *many-to-many matching*, allowing multiple objects in $\mathbf{X}$ to match multiple objects in $\mathbf{Y}$. In practice, they assume a joint subspace clustering model with Dirichlet process prior for the shared clusters, jointly clustering the samples and matching the clusters. Their model allows unconstrained optimization since each object can be allocated into one of the clusters independently. However, it does not solve problems like the language example above; instead of linking each word to a small set of its synonyms, it creates large clusters that group together possibly hundreds of words and pairs those clusters.

The existing literature omits a range of possible cross-domain object matching problems, namely those that assume *few-to-few matching* between the objects. For example, we might want to allow some objects in $\mathbf{X}$ to be paired with two or three objects in $\mathbf{Y}$. Another example relates to cluster-matching with small clusters, for example when searching for teams of individuals to compete against each other; the model by Iwata et al. (2016) cannot guarantee small clusters and hence is not applicable.

In this work we present a unified Bayesian model to solve the whole spectrum of cross-domain object matching problems. For one-to-one problems our proposed model improves on the model of Klami (2013) by using collapsed inference. For many-to-many problems with large clusters it is similar to Iwata et al. (2016) but performs full posterior inference instead of partial maximum likelihood. More importantly, we demonstrate for the first time a model capable of solving few-to-few matching problems that neither of the existing models is suitable for. The model builds on two core building blocks: The statistical dependency between the data sources is modeled with a variant of Bayesian CCA (Klami et al., 2013), extended to support latent variables shared by multiple objects. To solve the few-to-few matching problems we control the size of the clusters using a *microclustering* property (Miller et al., 2015). Microclustering refers to clustering models for which the size of the clusters grows sublinearly with the number of data, and the practical models allow explicitly controlling the size via soft and hard constraints. Our final model couples two microclusterings together with CCA, resulting in a unified subspace clustering model.

The model requires posterior inference over constrained sets, known as weighted discrete sampling (Chakraborty et al., 2014). For this intractable problem, we present a collection of Gibbs-sampling algorithms, some of them approximative, that shine under different conditions. The main empirical conclusion is that the special cases treated by previous solutions are easier to solve than the few-to-few case, but many few-to-few instances are solvable. To increase the accuracy in solving few-to-few cases we also describe and demonstrate a partially supervised variant that takes as input a few pairs of objects known to match.

2. Problem Formulation

Yamada and Sugiyama (2011) defines cross-domain object matching (CDOM) as the task of finding one-to-one match between two sets of objects. We consider the following generalizations of this problem that learn the matching between sets of objects instead:

Definition 1 Generalized CDOM*: Let* $\mathbf{X} \in R^{D_x \times N_x}$ *and* $\mathbf{Y} \in R^{D_y \times N_y}$ *denote two input matrices with no alignment provided for the rows or the columns. Define two cluster indicator matrices* $\Pi^x \in [0,1]^{K \times N_x}$ *and* $\Pi^y \in [0,1]^{K \times N_y}$ *with unit sums for the columns. The task in generalized CDOM, or cluster matching, is to learn* Π^x *and* Π^y *such that* $\mathbf{X}\Pi^{x^T} \in \mathbb{R}^{D_x \times K}$ *is statistically maximally dependent with* $\mathbf{Y}\Pi^{y^T} \in \mathbb{R}^{D_y \times K}$*.*

Definition 2 Constrained CDOM*: If we further constrain that each object can match with at most* U *objects, by requiring that the row-sums of* Π^x *and* Π^y *are at most* U*, we call the problem constrained CDOM.*

The definition for one-to-one CDOM is obtained as a special case of Constrained CDOM by setting $U = 1$, whereas Iwata et al. (2013, 2016) solve the Generalized CDOM. The constrained case with $U > 1$ is a novel definition, requiring new solutions. In this work we limit to two object sets represented with real-valued features; see Tripathi et al. (2011) and Quadrianto et al. (2010) for generalizations for multiples sets and structured inputs.

The solution is provided by the indicators Π^x and Π^y that reveal the structure of the bipartite graph specifying the dependencies in the joint density. In this work we consider posterior inference over the structure, instead of searching for the optimal one.

3. Background

Our solution combines a probabilistic interpretation of CCA with a microclustering property, and to understand the full model we here briefly go re-cap these two building blocks.

3.1 Canonical Correlation Analysis

Canonical correlation analysis (CCA) is a method that seeks for linear projections of $\mathbf{X} \in \mathbb{R}^{D_x \times N}$ and $\mathbf{Y} \in \mathbb{R}^{D_y \times N}$ such that they are maximally correlating. Here the columns of the two matrices are aligned, representing co-occurring objects. For CDOM solutions it is useful to consider the Bayesian interpretation of CCA (BCCA) (Klami et al., 2013), which more readily enables various extensions. Instead of explicitly maximizing the correlation, BCCA uses a joint latent subspace to model the two data sources. The generative model is

$$\mathbf{X} = \mathbf{W}^x\mathbf{Z} + \epsilon_x, \quad \mathbf{Y} = \mathbf{W}^y\mathbf{Z} + \epsilon_y, \quad \mathbf{Z}_{.i} \sim \mathcal{N}(0, \mathbf{I}_L),$$

where $\mathbf{Z} \in \mathbb{R}^{L \times N}$ represents L-dimensional latent variables, $\mathbf{W}^x \in \mathbb{R}^{D_x \times L}$ and $\mathbf{W}^y \in \mathbb{R}^{D_y \times L}$ are linear mappings, and ϵ_x and ϵ_y are white Gaussian noise. Group-wise sparsity prior, typically an automatic relevance determination (ARD) prior of the form

$$\mathbf{W}^x_{k,:} \sim \mathcal{N}(0, \mathrm{diag}(\alpha_x)), \quad \alpha_x \sim \mathrm{Gamma}(\alpha_0, \beta_0),$$

makes $\mathbf{W}^x$ and $\mathbf{W}^y$ column-sparse and hence determines which dimensions in $\mathbf{Z}$ correspond to correlations between the sources and which describe variation specific to each source.

3.1.1 Microclustering

Typical clustering models, parametric and non-parametric alike, tend to assume some specific size distribution for the clusters, and in particular increasing the amount of data increases the size (the number of objects) in a typical cluster. *Microclustering* (Miller et al., 2015) refers to models for which the cluster size grows sublinearly with the number of data points. Additionally, we can explicitly control the size of the clusters, instead of assuming the objects are assigned independently into the clusters according to some prior process. Miller et al. (2015) used negative-binomial distributions over the number of objects in a cluster, whereas Klami and Jitta (2016) assumed uniform probability over a range of cluster sizes. Both solutions help in finding small clusters even amongst large object collections.

By denoting a clustering matrix by $\Pi \in [0,1]^{K \times N}$ with unit column sums, a microclustering model can be written as

$$p(\Pi) = \prod_{k=1}^{K} p_K(\sum_{n=1}^{N} \Pi_{kn} | \phi), \quad p(\mathbf{X}_{:,n} | \Pi_{kn} = 1) = p_X(\mathbf{X}_{:,n} | \theta_k),$$

where $p_K(\cdot|\phi)$ denotes some probability distribution over non-negative integers (the number of samples in the cluster) with parameters ϕ and $p_X(\cdot|\theta_k)$ is a generating density (such as a Gaussian distribution) with parameters θ_k associated with the kth cluster. To produce microclusters the density $p_K(\cdot|\phi)$ should be chosen to give high probability for small clusters or to even exclude large clusters altogether. Inference for microclustering models is harder than for standard clustering models, because the prior does not factorize over the samples.

4. Model

Our model builds on BCCA and microclustering described above, and is closely related to existing CDOM models by Klami (2013) and Iwata et al. (2016). Both of them assume the data is generated by a latent linear Gaussian model:

$$X \sim N(\mathbf{W}^x \mathbf{Z} \Pi^x, \tau_x^{-1} I), \quad Y \sim N(\mathbf{W}^y \mathbf{Z} \Pi^y, \tau_y^{-1} I),$$

where τ_x and τ_y are precision parameters for the residual noise. Klami (2013) solved one-to-one problems and hence could assume $\Pi^x = \mathbf{I}$ while constraining Π^y to be a permutation matrix. The model of Iwata et al. (2016), in turn, can be written in this format by converting the cluster allocations made by the Dirichlet process into indicator matrices.

Following the above models we propose the natural generalization for the constrained CDOM case. We use the same linear Gaussian model, but replace the hard permutation constraint or the Dirichlet process clustering property with a process that implements the microclustering concept, allowing us to control the size of the matching groups. The model formulation is hence relatively straightforward unification of prior work. Posterior inference, however, is non-trivial and will be discussed in more detail in Section 5.

The proposed generative model that implements generalized CDOM is defined by

$$\begin{aligned} \mathbf{X} \sim \mathcal{N}(\mathbf{W}^x \mathbf{Z} \Pi^x, \tau_x^{-1} \mathbf{I}), \quad \mathbf{Y} \sim \mathcal{N}(\mathbf{W}^y \mathbf{Z} \Pi^y, \tau_y^{-1} \mathbf{I}), \quad \mathbf{Z}_{.k} \sim N(0, \mathbf{I}_L), \\ \mathbf{W}^x \sim \prod_{l=1}^{L} N(0, \alpha_{xl}^{-1} \mathbf{I}_{D_x}), \quad \mathbf{W}^y \sim \prod_{l=1}^{L} N(0, \alpha_{yl}^{-1} \mathbf{I}_{D_y}), \end{aligned} \tag{1}$$

where the priors for $\tau_x, \tau_y, \alpha_{xl}$ and α_{yl} are all Gamma distributions.

The missing part in the above formulation is the prior for Π^x and Π^y, which we set uniform over clustering matrices that satisfy the constraint that no cluster holds more than U objects in either data set. This corresponds to a special case of the constrained microclustering model by Klami and Jitta (2016); we do not consider minimum constraints as they did. If we here set $U = 1$ for all samples and require $N_x = N_y$, the model reduces exactly to the one Klami (2013) proposed for one-to-one matching. If we set $U = \infty$ the model becomes a coupled subspace clustering model, which is very similar in motivation to Iwata et al. (2016). However, it replaces their Dirichlet process prior with a uniform one. If one seeks to specifically solve only many-to-many instances, their model is preferable to our formulation but for few-to-few cases the constraints become crucial.

5. Inference

In this paper we perform Gibbs sampling to draw samples from the posterior distribution of the model parameters conditional on the observed data. Given the current cluster allocations Π^x and Π^y inference for the rest of the parameters is straightforward, and consequently only briefly summarized in the Appendix. Instead, we focus on the challenge of drawing samples for Π^x and Π^y conditional on the rest of the model, presenting four algorithms for the task. We first discuss three generic algorithms that are applicable for all problem instances (but are not efficient for all), and then discuss separately the case of one-to-one matching for which more efficient algorithms can be developed. For all cases the equations are written for Π^x, with direct analogues for Π^y.

5.1 Algorithm 1: Direct Gibbs

The most straightforward algorithm directly draws samples from the conditional density $p(\Pi^x|\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \Pi^y, \text{rest})$ (and alternatingly does the same for the Y side), where "rest" refers to the hyper-parameters. Since we are conditioning on all other variables it is easy to compute the relative likelihoods of all possible choices for Π_{ki} that refer to the nth object being mapped to the kth cluster. In case we are given no constraints ($U = \infty$), we can simply draw each sample independently.

Given the constrains, the task corresponds to discrete weighted sampling (Chakraborty et al., 2014). The problem has exponential complexity and no efficient solutions for the general case exist, but the special case of clustering in real-valued spaces seems to behave well in practice. Klami and Jitta (2016) showed that a simple rejection sampling algorithm is sufficient for many problems. Following their results, we use rejection sampling as follows:

1. Randomly select a subset of S objects to be allocated
2. Allocate them independently, according to the relative likelihoods
3. Accept the allocation if constraints are satisfied; if not, try again
4. Finally, in case we reach a given number (in our experiments 100) rejections, we keep the previous allocation

This algorithm is efficient for many-to-many cases with loose constraints, since it typically can accept already the first proposal and hence the overall complexity is that of a regular clustering model. With very tight constraints the next algorithm is often more efficient.

5.2 Algorithm 2: Approximative Gibbs Via Constrained Optimization

Given the logarithmic likelihoods $\mathbf{C}_{ik} = \log p(\Pi^x_{ik}| \dots)$ computed for all pairs of $i \in 1, \dots, N_x$ and $k \in 1, \dots, K$, we can also directly search for a solution Π^x that satisfies the constraints and maximizes the total likelihood by solving the constrained problem

$$\max \sum_{i=1}^{N_x} \sum_{k=1}^{K} \mathbf{C}_{ik} \Pi^x_{ik}, \quad \sum_i \Pi^x_{ki} \leq U \quad \forall k, \quad \sum_k \Pi^x_{ki} = 1 \quad \forall i \tag{2}$$

with a generic integer programming solver.

Directly allocating the samples based on maximum likelihood within a Gibbs sampler does not result in valid posterior samples, but can still produce a good CDOM algorithm. Klami (2013) showed that this kind of an approximative sampler works for one-to-one cases, and Iwata et al. (2016) use maximum likelihood for $\mathbf{W}^x$ and $\mathbf{W}^y$ in their model, demonstrating that even the existing state-of-the-art relies on approximate samplers.

The approximation can be made more justified by considering the set of best M solutions for the problem (2) instead of greedily choosing the best one. We can find consecutive solutions by solving the same problem again with one additional constraint preventing duplication of the previous solution, eventually sampling Π based on the relative total likelihoods of these solutions. For drawing samples from the full posterior the parameter M would often need to be impractically large, but as demonstrated in Section 6, already considering some small M improves the results.

This algorithm is typically more efficient than direct sampling when the constraints are tight, because the solver finds probable allocations much more efficiently than rejection sampling used in Algorithm 1.

5.3 Algorithm 3: Collapsed Gibbs

Collapsing over latent variables, when possible, typically leads to more efficient algorithms. In the case of Gaussian latent variable models we can easily integrate over $\mathbf{Z}$ when computing the relative likelihoods for alternative cluster allocations, which should be particularly useful with small clusters. In practice, we simply replace $\mathbf{C}_{ik}$ with the marginal likelihood

$$\begin{aligned}
\mathbf{C}_{ik} &= \frac{|\tau_x I|^{\frac{1}{2}} \, |\Sigma_0|^{\frac{1}{2}}}{(2\pi)^{\frac{D_z}{2}} \, |\Sigma_{zi}|^{\frac{1}{2}}} e^{\frac{-S_i}{2}}, \quad \text{where} \\
S_i &= \tau_x x_j^T x_j + \mathbf{M}_{zi}^T \Sigma_{zi}^{-1} \mathbf{M}_{zi} - \mu_0^T \Sigma_0^{-1} \mu_0, \\
\Sigma_0 &= (\tau_x \mathbf{W}^{xT} \mathbf{W}^x + \Sigma_{zi}^{-1})^{-1}, \quad \mu_0 = \Sigma_0 (\tau_x (\mathbf{W}^x)^T x_j + \Sigma_{zi}^{-1} \mathbf{M}_{zi}),
\end{aligned} \tag{3}$$

and $\mathbf{M}$ is defined as in the Appendix. In absence of constraints we can then draw each sample independently based on $p(\Pi^x_{ik} = 1) \propto e^{C_{ik}}$. This corresponds exactly to classical collapsed Gibbs sampling.

For the constrained CDOM problem we need to sample each assignment at a time, since computing the marginal likelihood requires knowing which other objects are assigned to the same cluster. With tight constraints this approach is likely to result in poor mixing, and in the extreme case of one-to-one matching the sample is guaranteed to always remain in its current cluster since it cannot fit to any of the other clusters. This issue can be remedied

by considering joint allocation of more than one sample: Instead of removing one sample and computing its marginal likelihood, we can remove S samples and compute the marginal likelihoods of all possible assignments of them. This approach works in principle for any S, but unfortunately has exponential complexity.

Nevertheless, for small values of S this can still result in a practical algorithm. We illustrate the algorithm for the special case of $S = 2$ to highlight its advantage over the classical take-one-out, but note that values up to 4-5 would also be efficient to compute. The algorithm treats separately the cases where only one of the two samples ends up into a given cluster, using (3) to compute it, and the case where both samples are being assigned to the same cluster. For that the marginal likelihood is

$$\hat{\mathbf{C}}_{ik} = \frac{|\tau_x I|^{\frac{1}{2}} \, |\Sigma_{od}|^{\frac{1}{2}}}{(2\pi)^{\frac{D_z}{2}} \, |\Sigma_{zi}|^{\frac{1}{2}}} e^{\frac{-S_{di}}{2}}, \text{ where } S_{di} = \tau_x x_j^T x_j + \tau_x x_k^T x_k + \mathbf{M}_{zi}^T \Sigma_{zi}^{-1} \mathbf{M}_{zi} - \mu_{od}^T \Sigma_{od}^{-1} \mu_{od},$$
$$\Sigma_{od} = (2\tau_x \mathbf{W}^{xT} \mathbf{W}^x + \Sigma_{zi}^{-1})^{-1}, \text{ and } \mu_{od} = \Sigma_{od}(\tau_x (\mathbf{W}^x)^T x_j + \tau_x (\mathbf{W}^x)^T x_k + \Sigma_{zi}^{-1} \mathbf{M}_{zi}).$$

We can now compute a $K \times K$ matrix $\mathbf{E}$ of the marginal log-probabilities of the joint allocation by taking the outer sum of $\mathbf{C}_{i:}$ and $\mathbf{C}_{j:}$ and replacing its diagonal with $\hat{\mathbf{C}}_{i:}$. Now all elements of $\mathbf{E}$ refer to valid probabilities and we can sample i and j.

5.4 Algorithm 4: One-to-one CDOM

The most constrained case of one-to-one CDOM deserves a special treatment. While all of the above algorithms are applicable to that scenario as well, the rigid structure makes it possible to marginalize out $\mathbf{Z}$ while still assigning all samples simultaneously.

We first compute the marginalized likelihood matrix $\mathbf{C}_{ik}$ using (3) for all possible allocations. Since each column corresponds to a likelihood of assigning an individual sample to a cluster no other sample is being assigned to, independent choice of maximal elements would result in inconsistency. However, if we now solve for an assignment that satisfies the one-to-one constraint using (2), we end up choosing exactly one element from each column, and hence the full solution is consistent. Note that this cannot be done with the general case since we typically assign multiple samples into one cluster and the marginal likelihoods computed by (3) would then be incorrect.

Klami (2013) made the same observation that constrained optimization helps in finding a consistent solution, but used it to jointly sample $\mathbf{Z}$ and Π instead of marginalizing over $\mathbf{Z}$. Our solution improves on theirs by both considering marginal likelihoods, and by allowing discrete sampling over alternative permutations as explained in Section 5.2; Klami (2013) could not do this because they used the Hungarian algorithm for finding the permutation instead of a generic integer programming solver.

5.5 Initialization And Partial Supervision

As pointed out by most earlier sources, matching solutions are very sensitive to initialization. For one-to-one case we follow the suggestion of Klami (2013); we make a brief run with the algorithm for multiple random initializations and then form a consensus of the results of these preliminary runs (Tripathi et al., 2011). We then initialize one long final sampling run based on that consensus.

We are not aware of easy ways of finding the consensus of multiple solutions to be used as initialization for the other cases, and hence we use pure random initialization for Π satisfying the constraints. Further research on the initialization would be very useful for improving the robustness of the method; the algorithms struggle to escape the local mode that corresponds to not modeling correlations but work well once they escape that.

One potential strategy for initialization is to provide some seed matches. Already Haghighi et al. (2008) pointed out that one-to-one CDOM can easily be generalized for partially supervised setups. Any known pairs are simply provided as fixed allocations in Π and they help in learning the parameters of the CCA model. For few-to-few matching a reasonable partially supervised setting assumes we know a few individual pairs of $\mathbf{x}$ and $\mathbf{y}$ that should be matched together. That is, we do not assume knowledge of the cluster structure within either set, but only imagine that individual pairs can be matched with manual effort or additional side information.

6. Experiments

We now compare the proposed algorithms under various scenarios, highlighting their relative merits for different kind of problems. As baselines we implicitly use Klami (2013) and Iwata et al. (2016) that correspond to state-of-the-art solutions for the one-to-one and many-to-many cases; in practice these are implemented as special cases of our model. Consequently, the baseline methods do not exactly correspond to the ones presented in the literature, but for the purpose of these comparisons the differences are small.

We measure the quality of the matching by recall of the correct shared cluster identities; for each pair of objects in different domains we increase the score by one and finally re-normalize the result between zero and one, with one referring to a perfect result. Typically we should not expect perfect match for data drawn from the model; normal distribution has very high probability mass around its mean and it would not be reasonable to expect all of those samples to be matched exactly with their correct counterparts.

Since the main goal of the work is to address combinatorial challenges in inference, we evaluate the models only in one-to-one and few-to-few scenarios. We also verified the algorithms correctly solve the many-to-many matching problem for the artificial data collections studied by Iwata et al. (2016), but omit the detailed results to save space.

6.1 One-to-one Matching

For the one-to-one case we created data sets by drawing samples randomly from the model (1), using $N = 80$ samples. For each data set we randomly chose the number of dimensions D_x and D_y from $[10, 100]$ and the residual noise parameters were drawn from Gamma$(1, 10)$ to create data sets of varying difficulty.

We initialize each algorithm based on a consensus of 20 initial runs terminated after just 20 iterations and then proceed to sample 300 samples with the algorithm. The results are summarized in Table 1. The main conclusion is that Algorithms 1-3 do not work for the tight constraints, as expected. The special algorithms developed for the one-to-one case are clearly more accurate.

The previous state-of-the-art by Klami (2013) (Gibbs-hard in the table) greedily picks the most likely permutation. Our Algorithm 4 adds marginalization over $\mathbf{Z}$ and also supports

Table 1: Accuracies for one-to-one matching. Algorithm 4 that is specifically designed for one-to-one matching works clearly better than Algorithms 1-3, and it also matches the accuracy of Gibbs-hard by Klami (2013).

Algorithm	Parameters	Accuracy
Algorithm 1	L = 2	0.01
Algorithm 1	L = 8	0.03
Algorithm 2	M = 1	0.06
Algorithm 2	M = 2	0.05
Algorithm 2	M = 4	0.06
Algorithm 3	take-two-out	0.03
Algorithm 4	M = 1	**0.12**
Algorithm 4	M = 2	**0.12**
Algorithm 4	M = 4	0.08
Gibbs-hard	M = 1	0.11
Gibbs-hard	M = 2	0.11
Gibbs-hard	M = 4	0.04

using $M > 1$ alternative permutations to be considered, but this does not seem to notably increase the accuracy in these experiments. Furthermore, using $M > 1$ seems to decrease the performance of both algorithms; we presume this is because the hard choice helps to regularize the algorithm during the early stages. Further experimentation with higher M and more data sets is an interesting future endeavour, but already these results demonstrate we can match the state-of-the-art.

6.2 Few-to-few Matching

The most interesting case considers scenarios where a few objects are to be matched to a few objects in the other data set, since no solutions for this problem have been presented previously. We created artificial data sets with varying total number of objects, so that the correct solution in each case always pairs two objects in $\mathbf{X}$ to two objects in $\mathbf{Y}$. We solve the problem assuming maximum count of $U = 3$, to allow some flexibility.

Figure 1 (left) plots the mean accuracy averaged over 40 randomly created data sets of each size, with the other properties of the data sets drawn randomly as in the one-to-one case. The results hence showcase the average behavior over different kinds of data sets; an algorithm can here be good by either being robust or by being very accurate for the cases it works. The result is that Algorithm 1 is not sufficient for solving the problem; it is consistently the worst together with Algorithm 3 using S=1. In other words, we need to allocate larger chunks of objects at once.

The collapsed Gibbs sampler (Algorithm 3) with $S = 2$ is consistently good, as is Algorithm 2 with $M = 8$. The former is computationally more efficient in most cases and hence the recommended choice, but for maximal accuracy one could try using Algorithm 2 with even larger M to better approximate the posterior.

6.3 Partially Supervised Few-to-few Matching

The above experiments considered purely unsupervised CDOM problems for data generated from the model. Next we illustrate partially supervised matching on a more realistic (but still artificially constructed) example of matching words across two languages. The words are represented using 64-dimensional embedding vectors provided for each language by polyglot (Al-Rfou et al., 2013). Since the embeddings are computed for each language separately, they are not represented in the same space and hence no direct distance measure exists.

We constructed a corpus of 800 English words grouped into sets of semantically related words, obtained by retrieving 20 neighbors (in the embedding space) for each of 40 words selected as seeds. The seed words were then translated to German and analogous set of 800 words were created. Besides the seed words the samples are not direct translations. We then extracted smaller subsets of these conceptual clusters to create few-to-few matching tasks of varying size. In the end, for example a four-to-four matching problem would seek to identify relationships like the set {'blue', 'purple', 'red', 'white'} matching with {'blau', 'gelb', 'schwarz', 'braun'}, without knowing the grouping in either language.

Following the results of the previous section, we choose Algorithm 3 with $S = 2$ as the inference solution and measure the accuracy using the adjusted Rand index that rewards for good performance in both parts of the task. Figure 1 (right) reports the mean accuracy over 20 randomly created data sets extracted from the general corpus of 800 words by picking different subsets of the words for each concept.

To study the partially supervised variant we assumed increasing number of individual word pairs (at most one for each concept) as known. The Rand index improves when more supervision is used, but importantly the score for the fully unsupervised case is not much worse than the case where we assume that a single pair is known for all of the true 40 clusters. Another observation is that larger clusters are easier to match; this is understandable since the posterior over the assignments is very broad for small clusters.

To illustrate the importance of imposing the microclustering property, we also ran the algorithm without constraints for all three problem instances, emulating the behavior of the model by Iwata et al. (2016). The Rand index drops considerably in all cases, and the resulting clusters severely violate the constraints, sometimes assigning more than ten words in a single cluster despite the true and desired cluster size being only four words.

7. Discussion

In this work we formulated the constrained cross-domain object matching problem, to enable solving *few-to-few* matching problems. It naturally complements the body of literature for one-to-one CDOM problems as well as those that address the cluster-matching problem where large groups of objects are matched together. We presented unified model for all cases and provided practical Gibbs sampling algorithms, which were demonstrated on particularly challenging problem instances to highlight the differences. Importantly, we developed the first algorithms applicable for the few-to-few problems.

The work leaves open two main concerns: How to better improve the robustness of the methods by better initialization, and how to more efficiently utilize constrained optimization solvers as part of the solution. For one-to-one problems an approximative solver that greedily picks the most likely cluster assignment is as good as more justified variants, but for

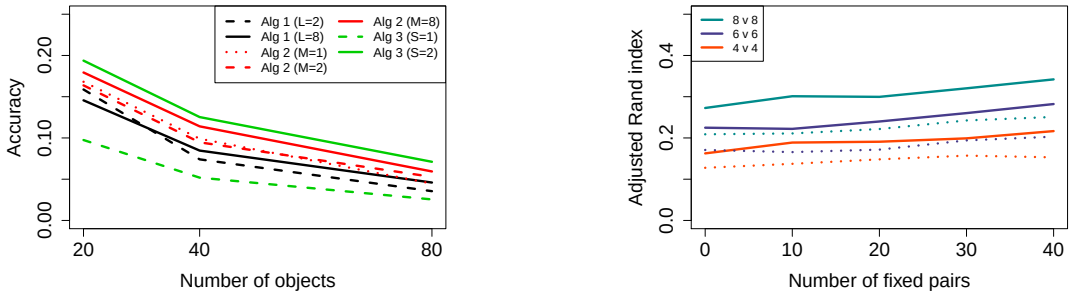

Figure 1: **Left**: Average accuracy over randomly generated few-to-few matching problems of varying difficulty. Algorithms 3 (take-two-out) and 2 (M=8) are the best results; whereas standard collapsed Gibbs sampling (Alg. 3; take-one-out) and the direct rejection samplers (Alg. 1) are clearly less accurate. **Right**: Illustration of partially supervised few-to-few matching for pairing concepts (groups of related words) across two languages, for three different problem instances corresponding to varying size of the true clusters. Using the constraints to enforce few-to-few matching (solid lines) clearly improves from ignoring the constraints (dotted lines). Partial supervision by fixing individual pairs (not full clusters) helps, but importantly the performance is good already for the fully unsupervised case.

few-to-few instances we showed how it pays off to avoid the greedy step and consider more accurate inference over the constrained set. The few-to-few task is harder than the special cases treated previously, but we demonstrated acceptable solutions for a task of matching clusters of words across two languages with no prior information.

Acknowledgments

The work was supported by Academy of Finland (grants 251170 and 266969).

Appendix

Given the current allocations Π the inference for the rest of the parameters are given below. By denoting $c_{xi} = \sum_{j=1}^{N_x} \Pi_{ij}^x$ and $c_{yi} = \sum_{j=1}^{N_y} \Pi_{ij}^y$ we get the update for Z as $p(Z_{.i} \mid \cdots) \sim N(Z_{.i} \mid \mathbf{M}_{zi}, \Sigma_{zi})$, where

$$\Sigma_{zi} = (I + c_{xi}\tau_x \mathbf{W}^{xT}\mathbf{W}^x + c_{yi}\tau_y \mathbf{W}^{yT}\mathbf{W}^y)^{-1}, \quad \mathbf{M}_{zi} = \Sigma_{zi}(\tau_x \mathbf{W}^{xT} \sum_{j:\Pi_{ij}^x==1} X_{.j} + \tau_y \mathbf{W}^{yT} \sum_{j:\Pi_{ij}^y==1} Y_{.j}).$$

The distribution for $\mathbf{W}^x$ is $p(\mathbf{W}^x{}_{j.} \mid \cdots) \sim N(\mathbf{W}_{j.}^x \mid \mathbf{M}_{\mathbf{W}_{j.}^x}, \Sigma_{\mathbf{W}^x})$, where

$$\Sigma_{\mathbf{W}^x} = \left(\hat{\alpha_x} + \tau_x \sum_{i_x=1}^{N_x} \hat{Z}_{.i_x}\hat{Z}_{.i_x}^T\right)^{-1}, \quad \mathbf{M}_{\mathbf{W}^x{}_{j.}} = \tau_x \Sigma_{\mathbf{W}^x} \sum_{j=1}^{D_x} \mathbf{W}_{j.}^{xT} \left(\sum_{i_x=1}^{N_x} X_{ji_x}\hat{Z}_{.i_x}\right),$$

and analogous expression is given for $\mathbf{W}^y$. The updates for the noise precisions (τ_x, τ_y) and the ARD parameters (α_x, α_y) are identical to those provided by Klami et al. (2013).

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the 7th Conference on Computational Natural Language Learning*, pages 183–192, 2013.

Supratik Chakraborty, Daniel J. Fremont, Kuldeep S. Meel, Sanjit A. Seshia, and Moshe Y. Vardi. Distribution-aware sampling and weighted model counting for SAT. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.

Nemanja Djuric, Mihajlo Grbovic, and Slobodan Vucetic. Convex kernelized sorting. In *Proceedings of 26th AAAI Conference on Artificial Intelligence*, pages 893–899, 2012.

Aria Haghighi, Percy Liang, Taylor Berh-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 771–779, June 2008.

Tomoharo Iwata, Tsutomu Hirao, and Naonori Ueda. Unsupervised cluster matching via probabilistic latent variable models. In *Proceedings of 27th AAAI Conference on Artificial Intelligence*, 2013.

Tomoharu Iwata, Tsutomu Hirao, and Naonori Ueda. Unsupervised many-to-many object matching via probabilistic latent variable models. *Information Processing & Management*, 52(4):682–697, 2016.

Arto Klami. Bayesian object matching. *Machine learning*, 92(2):225–250, 2013.

Arto Klami and Aditya Jitta. Probabilistic size-constrained microclustering. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, 2016.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14:965–1003, 2013.

Jeffrey Miller, Brenda Betancourt, Abbas Zaidi, Hanna Wallach, and Rebecca C. Steorts. Microclustering: When the cluster sizes grow sublinearly with the size of the data set. *arXiv:1512.00792*, 2015.

Novi Quadrianto, Alex J. Smola, Le Song, and Tinne Tuytelaars. Kernelized sorting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1809–1821, 2010.

Rebecca C. Steorts, Rob Hall, and Stephen E. Fienberg. A Bayesian approach to graphical record linkage and de-duplication. *Journal of the American Statistical Association*, 2016.

Abhishek Tripathi, Arto Klami, Matej Orešič, and Samuel Kaski. Matching samples of multiple views. *Data Mining and Knowledge Discovery*, 23:300–321, 2011.

Makoto Yamada and Masashi Sugiyama. Cross-domain object matching with model selection. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 807–815, 2011.