Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Convex Kernelized Sorting

Nemanja Djuric and **Mihajlo Grbovic** and **Slobodan Vucetic**
Department of Computer and Information Sciences, Temple University, USA
{nemanja.djuric, mihajlo.grbovic, vucetic}@temple.edu

Abstract

Kernelized sorting is a method for aligning objects across two domains by considering within-domain similarity, without a need to specify a cross-domain similarity measure. In this paper we present the Convex Kernelized Sorting method where, unlike in the previous approaches, the cross-domain object matching is formulated as a convex optimization problem, leading to simpler optimization and global optimum solution. Our method outputs soft alignments between objects, which can be used to rank the best matches for each object, or to visualize the object matching and verify the correct choice of the kernel. It also allows for computing hard one-to-one alignments by solving the resulting Linear Assignment Problem. Experiments on a number of cross-domain matching tasks show the strength of the proposed method, which consistently achieves higher accuracy than the existing methods.

Introduction

Object matching has been recognized as an important problem in many areas of machine learning. The problem can be described as follows: given two sets of objects, match the objects from the first set to the objects in the second set so that they are "similar". Applications and research areas as diverse as graph matching in computer vision (Luo and Hancock 2001), document alignment in natural language processing (Jagarlamudi, Juarez, and Daumé III 2010), and multiple sequence alignment in bioinformatics (Bacon and Anderson 1986), all rely on the assumption that there is an underlying correspondence between two sets of objects (possibly originating from two quite different domains) which can be learned. Object matching is also closely related to transfer learning (Wang and Yang 2011), as the cross-domain alignments can be used to transfer knowledge to new domains. This common thread can be extended to many other areas, and a number of recently published works on the topic indicates the importance of the matching problem.

Assuming that the objects from two sets are directly comparable, the task amounts to finding matches so that certain cross-domain measure of similarity is maximized. However, if the objects originate from different domains and are represented in different feature spaces, specifying a similarity measure can be very challenging. For instance, if we have a set of documents in French and a set of documents in Chinese language, it is not readily obvious how to find similar documents without a bilingual dictionary. Several ideas have been proposed to solve this unsupervised problem. Some approaches follow a simple rationale: match two objects that have comparable local neighborhoods, each in their own domain. A direct realization of this idea is Manifold Alignment (MA) (Wang and Mahadevan 2009), where the information about local relationships is translated into cross-domain measure of similarity. Following different line of reasoning, (Haghighi et al. 2008) and (Tripathi et al. 2011) describe a method that aligns objects by maximizing the correlation between two sets using Canonical Correlation Analysis (Hotelling 1936). In order to find one-to-one alignments they propose an EM-style algorithm, which iteratively finds new canonical variables and best alignments in a new common feature space until convergence.

Following the idea from MA, Kernelized Sorting (KS) (Quadrianto et al. 2010) tries to match objects with similar neighborhoods. However, unlike MA that uses only information about local context, KS method computes two kernel matrices, one for each object set. Then, using the kernel matrices, an alignment of objects across domains is found so that the dependency between matched pairs is maximized, measured in terms of Hilbert-Schmidt Independence Criterion (HSIC). Although conceptually very powerful, the technique is highly unstable as the object matching problem is defined as *maximization* (not minimization) over a convex function, subject to linear constraints. This renders KS algorithm very vulnerable to local optima and poor initialization. In (Jagarlamudi, Juarez, and Daumé III 2010), authors propose several modifications to the original algorithm to obtain a more stable method, but the local optima issue persists nevertheless. Note that some authors propose maximization of independence criteria other than HSIC (Yamada and Sugiyama 2011), but as the optimization problem is essentially unmodified, the methods are still plagued with the same instability problems. To solve this issue, we propose a convex formulation of the object matching problem, resulting in a robust algorithm guaranteed to find the optimal solution.

Kernelized Sorting

Let $X = \{x_1, x_2, ..., x_m\}$ and $Y = \{y_1, y_2, ..., y_m\}$ be two sets of equal size m of objects from two possibly different domains $\mathcal{X}$ and $\mathcal{Y}$, respectively. Further, let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be two kernels associated with domains $\mathcal{X}$ and $\mathcal{Y}$. Given kernel matrices K and L capturing the relationships between objects within X and Y sets, namely $K_{ij} = k(x_i, x_j)$ and $L_{ij} = l(y_i, y_j)$, the task is to find a one-to-one correspondence between objects in X and objects in Y. The cross-domain correspondence is encoded in $m \times m$ permutation matrix $\pi \in \Pi_m$, where $\pi_{ij} = 1$ if objects x_j and y_i are matched, $\pi_{ij} = 0$ otherwise, and Π_m is the set of all $m \times m$ permutation matrices. In the remainder of the paper, we will use notation π to denote both the permutation matrix and the corresponding object alignment, where the usage should be clear from the context.

In (Quadrianto et al. 2010), authors propose a method to find alignment π by maximizing the dependency between K and L as measured by Hilbert-Schmidt Independence Criterion. Given two random variables x and y drawn from some joint probability distribution $\mathbb{P}_{xy}$, HSIC (Smola et al. 2007) measures the dependence between variables x and y by computing the Hilbert-Schmidt norm Δ^2 of the cross-covariance operator between Reproducing Kernel Hilbert Spaces (RKHS) on $\mathcal{X}$ and $\mathcal{Y}$, and the norm equals 0 if and only if variables x and y are independent. If we define the projection matrix $H = I - 1/m$, which centers the data in the feature space, to obtain centered versions of K and L as $\overline{K} = HKH$ and $\overline{L} = HLH$, respectively, the Hilbert-Schmidt norm can be empirically estimated (Smola et al. 2007) as

$$\Delta^2 = m^{-2} \cdot \text{trace}(\overline{K} \cdot \overline{L}). \tag{1}$$

In order to find the best correspondence between objects across two domains, we seek such permutation π^* of rows and columns of kernel matrix $\overline{L}$ (i.e., the alignment between two sets of objects) that maximizes the dependency (1) of two kernel matrices. Thus, we obtain the following optimization problem,

$$\pi^* = \arg\max_{\pi \in \Pi_m} \text{trace}(\overline{K} \pi^{\mathrm{T}} \overline{L} \pi). \tag{2}$$

The optimization problem is an instance of quadratic assignment problem, therefore NP-hard in general, and the authors propose an iterative, approximate method for finding the matrix π^*. Given permutation matrix π_i at i^{th} iteration, the optimization problem (2) is converted into Linear Assignment Problem (LAP) and solved for optimal π_{i+1} using one of the available LAP solvers (Kuhn 1955; Jonker and Volgenant 1987). Note that the optimization problem (2) involves *maximization* over convex function of π. As a result, the optimization procedure is unstable and highly sensitive to initial value of the permutation matrix π_0 due to local optima issues.

Instead of HSIC, other independence criteria can also be used. In (Yamada and Sugiyama 2011), authors propose two variants of KS algorithm, using normalized cross-covariance operator and least-squares mutual information as measures of dependence between matrices K and L. However, as both the optimization problem and the optimization procedure are essentially the same as the ones used in KS method from (Quadrianto et al. 2010), the algorithms are still susceptible to local optima issues.

In order to mitigate the instability issue of KS methods, authors of (Jagarlamudi, Juarez, and Daumé III 2010) describe two modifications to the original algorithm. Although the authors were motivated by the problem of application of KS to Natural Language Processing (NLP), it is straightforward to use their method in other areas as well. The first improvement they propose is p-smooth, which involves using sub-polynomial kernel on the original kernel matrices. Given a kernel matrix M, each element of a kernel matrix is raised to the power of p (with $0 < p \leq 1$) to obtain a matrix M_p. This is followed by normalization of rows to unit length and calculation of a new kernel matrix as $M \leftarrow M_p \cdot M_p^{\mathrm{T}}$. The second improvement is to use a set of seed alignments, which will be favored during the optimization procedure. In this way, the effect of low-confidence, thus possibly incorrect alignments can be reduced during the optimization process. Although the modifications result in considerably more robust algorithm (with the cost of introducing an additional parameter p), the issue of local optima due to non-convex optimization problem remains.

Convex Kernelized Sorting

In this section, we present a method that is, unlike Kernelized Sorting described in the previous section, guaranteed to find a global optimum solution. We begin by observing that, given two $m \times m$ matrices $\overline{K}$ and $\overline{L}$, value of trace$(\overline{K} \cdot \overline{L})$ is maximized if rows and columns of $\overline{K}$ and $\overline{L}$, respectively, are permuted such that rows of $\overline{K}$ and corresponding columns of $\overline{L}$ are identical up to a constant multiplier. Then, we can define a convex optimization problem for finding the optimal permutation matrix π^* as a minimization of the difference between the left half and transpose of the right half of matrix product under trace operator in (2), or more formally

$$\underset{\pi \in \Pi_m}{\text{minimize}} \quad ||\overline{K} \cdot \pi^{\mathrm{T}} - (\overline{L} \cdot \pi)^{\mathrm{T}}||_F^2, \tag{3}$$

where $||\cdot||_F$ is Frobenius (or Hilbert-Schmidt) matrix norm. The product $\overline{K} \cdot \pi^{\mathrm{T}}$ permutes columns of $\overline{K}$, while $(\overline{L} \cdot \pi)^{\mathrm{T}}$ permutes rows of $\overline{L}$, effectively aligning objects in the same way as when both rows and columns of only one of the kernel matrices are permuted. The optimization problems defined in equations (2) and (3) are equivalent, as shown by the following lemma.

Lemma 1. *The problems (2) and (3) are equivalent.*

Proof. Let $A = \overline{K} \cdot \pi^{\mathrm{T}}$ and $B = \overline{L} \cdot \pi$, and let a_{ij} and b_{ij} be values in i^{th} row and j^{th} column of A and B. Further, let $\mathcal{L}_1 = ||\overline{K} \cdot \pi^{\mathrm{T}} - (\overline{L} \cdot \pi)^{\mathrm{T}}||_F^2$ and $\mathcal{L}_2 = \text{trace}(\overline{K} \pi^{\mathrm{T}} \overline{L} \pi)$. Then, it follows

$$\mathcal{L}_1 = \sum_{i=1}^{m} \sum_{j=1}^{m} (a_{ij} - b_{ji})^2 =$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}^2 + \sum_{i=1}^{m}\sum_{j=1}^{m} b_{ji}^2 - 2 \cdot \sum_{i=1}^{m}\sum_{i=1}^{m}(a_{ij} \cdot b_{ji}) =$$
$$\sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}^2 + \sum_{i=1}^{m}\sum_{j=1}^{m} b_{ji}^2 - 2 \cdot \mathcal{L}_2.$$

The first two elements of the final sum do not depend on π, as right multiplication with permutation matrix only permutes columns while not changing the values of the original elements of the matrix. Consequently, it follows

$$\arg\min_{\pi \in \Pi_m} \mathcal{L}_1 = \arg\max_{\pi \in \Pi_m} \mathcal{L}_2.$$

□

In order to obtain a convex optimization problem, we relax the constraint that π is strictly a permutation matrix, and define the following problem (by $\mathbf{1}_m$ we denote an m-dimensional column vector of all ones)

$$\begin{aligned} \underset{\pi}{\text{minimize}} \quad & ||\overline{K} \cdot \pi^{\mathrm{T}} - (\overline{L} \cdot \pi)^{\mathrm{T}}||_F^2 \\ \text{subject to} \quad & \pi_{ij} \geq 0, \text{for } i,j \in \{1,2,...,m\}, \\ & \pi \cdot \mathbf{1}_m = \mathbf{1}_m, \\ & \pi^{\mathrm{T}} \cdot \mathbf{1}_m = \mathbf{1}_m, \end{aligned} \tag{4}$$

where the permutation matrix binary constraint $\pi_{ij} \in \{0,1\}$ is replaced by the interval constraint $\pi_{ij} \in [0,1]$. The constraints in (4) ensure that the matrix π is doubly-stochastic (permutation matrix is a special case), meaning that it is a matrix with positive elements with both rows and columns summing up to 1. Thus, we are directly solving the problem (2) by converting it into convex problem (4). The resulting optimization problem is convex because the objective function is a composition of convex non-decreasing square function and convex norm function with affine argument, subject to linear equality and inequality constraints.

The relaxation of the constraint which restricts π to be strictly permutation matrix comes with several advantages. Mainly, as shown above, the problem of finding cross-domain object alignments can be stated as convex optimization problem, lending itself to simpler optimization, effective computational methods, and globally optimal solution (Boyd and Vandenberghe 2004). In this way we also avoid sensitivity to initialization from which the method from (Quadrianto et al. 2010) suffered. In addition, soft assignments sum up to 1 and, consequently, assignment π_{ij} can now be interpreted as a probability that objects x_j and y_i match. Therefore, the output of the algorithm is more informative than hard alignments, giving us a degree by which two objects should be aligned.

In order to numerically solve the problem (4), we compute first and second derivatives, with respect to the elements of matrix π, of the objective function $\mathcal{L} = ||\overline{K} \cdot \pi^{\mathrm{T}} - (\overline{L} \cdot \pi)^{\mathrm{T}}||_F^2$. The gradient can be calculated using

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \pi_{ij}} = 2 \cdot \sum_{k=1}^{m} \Big(& -\overline{L}_{ki} \cdot \sum_{l=1}^{m} (\overline{K}_{jl} \cdot \pi_{kl} - \pi_{lj} \cdot \overline{L}_{kl}) + \\ & \overline{K}_{kj} \cdot \sum_{l=1}^{m} (\overline{K}_{kl} \cdot \pi_{il} - \pi_{lk} \cdot \overline{L}_{il}) \Big), \end{aligned} \tag{5}$$

while a Hessian matrix of the objective function $\mathcal{L}$ can be calculated using

$$\begin{aligned} \frac{\partial^2 \mathcal{L}}{\partial \pi_{ij} \partial \pi_{rn}} = 2 \cdot \Bigg(& \sum_{k=1}^{m} \Big(I(i=r) \cdot \overline{K}_{kj} \cdot \overline{K}_{kn} + \\ & I(j=n) \cdot \overline{L}_{ki} \cdot \overline{L}_{kr} \Big) - 2 \cdot \overline{K}_{jn} \cdot \overline{L}_{ir} \Bigg), \end{aligned} \tag{6}$$

where $i,j,r,n \in \{1,2,...,m\}$, and $I(arg)$ is a binary indicator function equal to 1 if arg is true, and 0 otherwise.

Numerical optimization

The optimization problem (4) can be easily solved using one of the high-level optimization packages such as `CVX`[1], a package for specifying and solving convex programs. However, since the number of variables scales as $\mathcal{O}(m^2)$, this and other similar solvers are not suitable if we want to align larger sets of objects. To make the numerical optimization more scalable, we present another convex variant of the optimization problem (4). In order to reduce the number of constraints and retain only inequalities, we reformulate (4) into the following dual problem

$$\begin{aligned} \underset{\pi}{\text{minimize}} \quad & ||\overline{K} \cdot \pi^{\mathrm{T}} - (\overline{L} \cdot \pi)^{\mathrm{T}}||_F^2 + \\ & C \cdot \sum_{k=1}^{m} \left((\sum_{l=1}^{m} \pi_{kl} - 1)^2 + (\sum_{l=1}^{m} \pi_{lk} - 1)^2 \right) \\ \text{subject to} \quad & \pi_{ij} \geq 0, \text{for } i,j \in \{1,2,...,m\}, \end{aligned} \tag{7}$$

where we set regularization parameter C to some large value (we set $C = 10$ in our experiments). Then, since we now have only inequalities without any equality constraints, we can use the trust-region-reflective algorithm (Coleman and Li 1996) for solving convex problem (7) (algorithm implemented in, e.g., `fmincon` solver available in Matlab). With this modification, the gradient and the Hessian of the objective function are slightly different than (5) and (6). More specifically, $2 \cdot C \cdot (\sum_{k=1}^{m}(\pi_{jk} + \pi_{ki}) - 2)$ is added to (5), while Hessian is modified by adding $4 \cdot C$ to diagonal elements, and $2 \cdot C$ to off-diagonal elements if $(i = r)$ or $(j = n)$ in equation (6).

It is important to mention that, due to the nature of the trust region method and convexity of the optimization problem, approximate, diagonal-only Hessian can be used, where all off-diagonal elements are set to 0. Use of the exact Hessian leads to fewer iterations until convergence of the trust-region-reflective algorithm, while, on the other hand, the use of approximate Hessian requires significantly less memory and floating-point operations per iteration to reach globally optimal solution.

Note that after solving the convex problem (7), matrix π is not necessarily doubly-stochastic, although sums of rows and sums of columns will be very close to 1. Therefore, as a final step in our algorithm, we can project matrix π, the solution of problem (7), to the set of doubly-stochastic matrices. Finding this closest projection is an old problem, and we can

[1] http://cvxr.com/cvx/

use one of the proposed, approximate methods (Sinkhorn and Knopp 1967; Parlett and Landis 1982). Alternatively, we can define an additional convex optimization problem in order to find the optimal, closest doubly-stochastic matrix π^{DS} as follows

$$\begin{aligned} \underset{\pi^{DS}}{\text{minimize}} \quad & ||\pi - \pi^{DS}||_F^2 \\ \text{subject to} \quad & \pi_{ij}^{DS} \geq 0, \text{for } i,j \in \{1,2,...,m\}, \\ & \pi^{DS} \cdot \mathbf{1}_m = \mathbf{1}_m, \\ & (\pi^{DS})^{\mathrm{T}} \cdot \mathbf{1}_m = \mathbf{1}_m, \end{aligned} \tag{8}$$

which can be solved as a constrained linear least-squares problem or constrained Euclidean norm, and an optimal solution can be found by a number of solvers (e.g., `SeDuMi`[2] or `lsqlin` solver available in Matlab).

Soft and hard assignments

Soft assignments contained in matrix π provide deeper understanding of the cross-domain alignment between two sets of objects. As such, output of Convex Kernelized Sorting (CKS) can be used to rank the objects by sorting them by the probabilities that they should be aligned. For instance, if we are matching documents in French with documents written in Chinese language, for each French document we can provide a user with top 5 matches together with the degree showing how confident about each match are we.

Similarly to the existing methods, CKS can also give hard alignments, defined as a one-to-one correspondence between two sets of objects. In order to obtain the hard alignments, we use the Hungarian algorithm (Kuhn 1955) to solve LAP defined by the learned doubly-stochastic matrix π. The Hungarian algorithm is a polynomial-time algorithm for combinatorial optimization, efficiently solving the linear assignment problem. More specifically, if we have m workers each requiring certain pay to work on one of m jobs, the algorithm finds the lowest-cost one-to-one assignment of workers to jobs in $\mathcal{O}(m^3)$ time.

Finally, quality of the obtained alignments greatly depends on the choice of kernel functions k and l (Yamada and Sugiyama 2011). Using CKS we can visualize the alignments and thus provide a user with additional evidence that appropriate kernels were used. For instance, if Gaussian kernel is used to generate kernel matrices K and L, choice of kernel-width parameter will critically influence the alignment. One simple way to verify that appropriate value of the parameter is chosen is to plot matrix π (e.g., using function `imagesc` in Octave and Matlab) and visually check if there are clear matches between objects. By visualization of the results, a user can be more confident that appropriate kernel functions and parameters are used.

Experiments

In this section, we empirically evaluate performance of our CKS algorithm. We used the trust-region-reflective algorithm to solve convex problem (7), and stopped the optimization either after $10{,}000$ iterations or when objective function stopped decreasing by more than 10^{-6}. In all our experiments we used the diagonal approximation of the Hessian, as experiments showed that this leads to substantial gains in memory and time. Unlike the competing algorithms, CKS does not depend on the initialization of π, and as an initial point for the trust-region-reflective algorithm we simply set $\pi = \mathbf{1}_m \cdot \mathbf{1}_m^{\mathrm{T}}/m$. To find hard alignments, we used implementation of the Hungarian algorithm available online[3]. Lastly, we also note that the projection of nearly doubly-stochastic matrix π to the set of doubly-stochastic matrices was not necessary, since we found that this additional step did not significantly affect the results.

We compared CKS performance to Kernelized Sorting algorithm from (Quadrianto et al. 2010), and to p-smooth Kernelized Sorting (KS-p) method from (Jagarlamudi, Juarez, and Daumé III 2010). We also compared our algorithm to Least-Squares Object Matching method[4] (LSOM) (Yamada and Sugiyama 2011), which maximizes least-squares mutual information criterion instead of HSIC. Except for CKS, which requires only a single run, all experiments were repeated 10 times with both random and PCA-based initializations of matrix π, and the average and the best accuracy are reported. For each run of p-smooth KS method we tried all p values from 0.1 to 1 in increments of 0.1, and used the best performance as a result for that run. Manifold Alignment algorithm (Wang and Mahadevan 2009) was also considered, but this local-neighborhood method performed poorly in the experiments, which can be explained by the high dimensionality of the matching tasks.

First, we show application of CKS to data visualization, and then explore performance on the task of alignment of left and right halves of images from a set of images. Here, we used a set of 320 images from (Quadrianto et al. 2010). The images were processed in the same manner as in (Quadrianto et al. 2010). Namely, images were resized to 40×40 pixels, then converted from RGB to LAB color space and vectorized. For experiments in which we used the image set, kernel matrices were generated using Gaussian RBF kernel $k(x,x') = l(x,x') = exp(-\gamma \cdot ||x - x'||^2)$, setting γ to inverse median of $||x - x'||^2$. Finally, application to document alignment is presented, where we used linear kernel function to compute the kernel matrices.

Data visualization

One of possible applications of CKS is data visualization. Although we can use the existing algorithms for low-dimensional projection of high-dimension data (e.g., Principal Component Analysis), they cannot be used if we want to project data to arbitrary fixed structures (Quadrianto et al. 2010). KS methods, on the other hand, can be used for projection of data in such cases.

Figure 1 shows a low-dimensional projection of 320 images from the data set to a 2-dimensional "AAAI 2102" letter grid. The pattern in the layout is discernible, as similar images are closer in the grid, and lighter images are located

[2] http://sedumi.ie.lehigh.edu/

[3] http://www.mathworks.com/matlabcentral/fileexchange/20652-hungarian-algorithm-for-linear-assignment-problems-v2-3

[4] code from sugiyama-www.cs.titech.ac.jp/∼yamada

near the middle of the grid while darker ones are closer to the edges.

Figure 1: Alignment of 320 images to "AAAI 2012" letter grid

Image alignment

As done in (Quadrianto et al. 2010), we split the 320 images into left and right image halves. The task is to align a set of left halves with the set of right halves, and measure how many halves are correctly matched. We increased the number of images from 5 to 320 in increments of 5, and the performances of 4 algorithms for all image set sizes are given in Figure 2. It can be seen that CKS consistently recovers more original images than the competing algorithms. The problem with local optima of KS method is clearly illustrated in Figure 2a. The results for two consecutive image set sizes are very inconsistent, due to the sensitivity of the method. The modifications introduced by KS-p result in more robust algorithm (Figure 2b), but the method still achieves smaller number of correct matches than both LSOM and CKS. LSOM method is for most image set sizes quite competitive when considering the best achieved accuracy, although the average performance and the wide confidence intervals indicate the instability of the method, see Figure 2c. Figure 3 illustrates the outcome of matching of all 320 images using CKS, which had 64.38% matching accuracy after applying the Hungarian algorithm to compute hard assignments.

One of the advantages of CKS is that it outputs soft assignments. This can be used to rank the alignments by a probability of a match, and is illustrated in Figure 4a where we calculated recall for top N ranks. If we sort the soft assignments, then 81.25% of correct image pairs are ranked among the top 5, while for the top 13 matches the recall jumps to nearly 90%. Furthermore, for each left image half, Figure 4b shows the maximum learned probability among right image halves versus the learned probability of the correct match (as a reference we plot $y = x$ line). As can be seen, the probability of the correct match is either the highest, or is very close to the highest probability.

Finally, in order to visually verify quality of the alignment and evaluate appropriateness of kernel parameters, we can also plot the learned matrix π (see Figure 4c, darker color corresponds to higher probability of alignment). To make the result more clear, we arranged two sets of image halves in such a way so that the i^{th} image in the left-halves set should be matched to the i^{th} image in the right-halves set. Consequently, the perfect matrix π would be equal to the identity matrix. As shown in Figure 4c, CKS recovered π very close to the identity matrix. Too few or too many dark peaks in the plot can be used as a visual clue that the choice of parameters might be wrong, thus providing an additional, visual insight into the matching problem.

Multilingual document alignment

We evaluate CKS on the task of aligning multilingual documents. We used the Europarl Parallel Corpus[5] (Koehn 2005), a collection of proceedings of the European Parliament available in 10 different European languages. The objective was to align a set of English documents with a set of corresponding documents written in other languages. As a preprocessing step, stopwords were removed and stemming was performed using Snowball[6]. Then, the documents were represented using TF-IDF features of a bag-of-words kernel, followed by ℓ_2 normalization of feature vectors to unit length. As the resulting kernel matrices were notably diagonally dominant, for KS method we zeroed-out the diagonal, as suggested by authors in (Quadrianto et al. 2010). For the remaining algorithms, no post-processing was done and the original kernel matrices were used.

The results are presented in Table 1, where we report the rounded average and the best achieved (given in parentheses) number of correctly aligned documents after 10 runs. As a baseline reference, we calculated the performance of a simple method which aligns the documents according to their lengths. We can see that CKS algorithm recovered more correct matches than all other competing methods on all 9 NLP tasks. Even if we consider only the best performances achieved by the competing algorithms, our method still outperforms the competition. Considering that CKS algorithm requires only a single run, this is an impressive result. Furthermore, for most languages our method found nearly all the correct matches, except on the tasks of alignment of documents written in Finnish and Swedish languages. Although the two Nordic languages proved to be very challenging for all methods, CKS achieved around 5 times more correct alignments than the previously proposed algorithms. Interestingly, on these two tasks the baseline method is the second best, actually performing better than the existing algorithms.

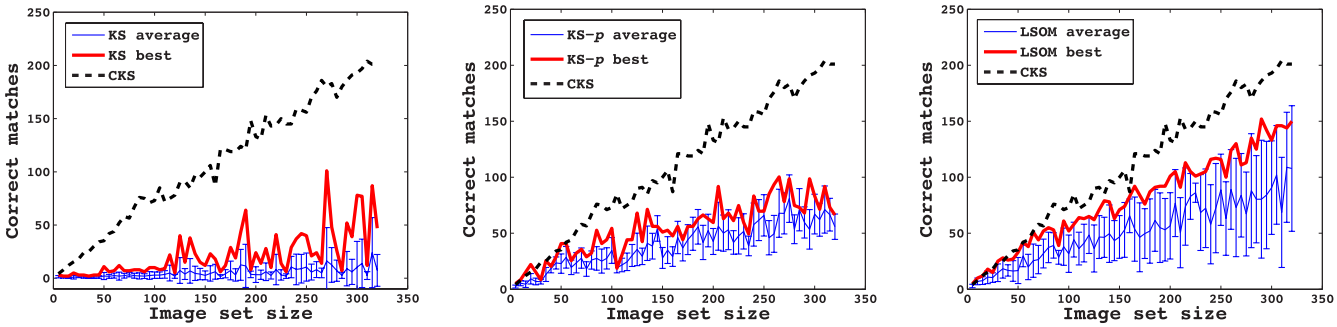
Conclusion

In this paper we presented a novel, convex formulation of the object matching problem. Unlike previous approaches, the CKS algorithm avoids the local optima issue and is guaranteed to find a globally optimal solution. The method outputs soft alignments between objects of two sets, which can be used to rank the best matches for each object, or to obtain one-to-one alignments by solving a Linear Assignment Problem. The performance of our method was evaluated on several matching tasks from image processing and natural language processing, where we observed that CKS consistently outperformed the existing approaches. All material used in the paper is available upon request.

There are several avenues which can be pursued further. The method can be easily extended to semi-supervised set-

[5] http://www.statmt.org/europarl/archives.html, version 1

[6] http://snowball.tartarus.org/



(a) Comparison of CKS and KS (b) Comparison of CKS and KS-p (c) Comparison of CKS and LSOM

Figure 2: Number of correctly recovered images for different image set sizes (error bars represent confidence intervals of one standard deviation, calculated after 10 runs)

Figure 3: Alignment results of CKS for matching of 320 left and right image halves (206 correct matches)

ting, by adding new constraints to the convex optimization problem. For instance, if we suspect that two objects should match, we can constrain the corresponding permutation matrix element to a certain value. It would be interesting to see how our method would work in this setting. Another issue is the restriction that two sets which are being aligned need to have the same number of elements. Although there are proposed solutions (e.g., padding the smaller set with dummy objects), it remains to be seen how would these ideas work with CKS algorithm. In the future work we plan to address both of these issues.

Acknowledgments

The authors thank Vladan Radosavljevic for helpful discussions, Qingqing Cai and Xin Li for their help during preprocessing of the data for the NLP task, and also acknowledge help from Jagadeesh Jagarlamudi and Hal Daumé III, as well as Novi Quadrianto, who provided material used in their respective papers.

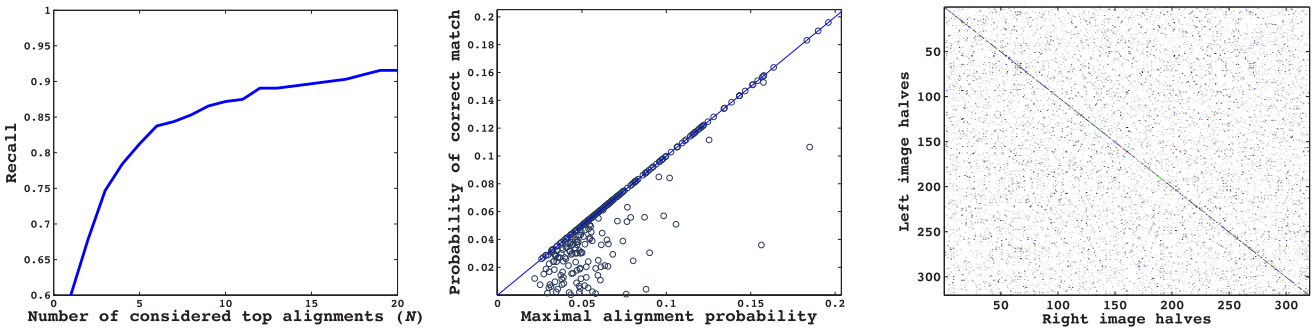
References

Bacon, D. J., and Anderson, W. F. 1986. Multiple sequence alignment. *Journal of Molecular Biology* 191(2):153–161.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. New York, NY, USA: Cambridge University Press.

Coleman, T. F., and Li, Y. 1996. An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM Journal on Optimization* 6(2):418–445.

Haghighi, A.; Liang, P.; Berg-Kirkpatrick, T.; and Klein, D. 2008. Learning Bilingual Lexicons from Monolingual Cor-



(a) Recall for top N-ranked alignments

(b) Maximal learned probability vs. correct-match probability (each circle represents a single object)

(c) Matrix π for alignment of 320 image halves

Figure 4: Analysis of CKS performance on the task of aligning 320 left and right images halves

Table 1: Average number of correctly aligned non-English documents to documents in English (numbers in the parentheses are the best obtained performances; baseline method matches the documents according to their lengths)

Language	**Corpus size**	Baseline	KS	KS-p	LSOM	CKS
Danish	387	39	261 (318)	258 (273)	159 (173)	**379**
Dutch	387	50	266 (371)	237 (317)	146 (375)	**383**
Finnish	308	54	19 (32)	22 (38)	10 (10)	**114**
French	356	64	319 (356)	320 (334)	354 (354)	**356**
German	356	50	282 (344)	258 (283)	338 (350)	**356**
Italian	387	49	341 (382)	349 (353)	378 (381)	**385**
Portuguese	356	46	308 (354)	326 (343)	342 (356)	**356**
Spanish	387	48	342 (365)	351 (364)	386 (387)	**387**
Swedish	337	76	20 (39)	20 (33)	5 (5)	**97**

pora. In *Proceedings of ACL-08: HLT*, 771–779. Association for Computational Linguistics.

Hotelling, H. 1936. Relation between two sets of variables. *Biometrica* 28:322–377.

Jagarlamudi, J.; Juarez, S.; and Daumé III, H. 2010. Kernelized Sorting for Natural Language Processing. In *AAAI Conference on Artificial Intelligence*, 1020–1025. AAAI Press.

Jonker, R., and Volgenant, A. 1987. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* 38:325–340.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Machine Translation Summit X*, 79–86.

Kuhn, H. W. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2(1-2):83–97.

Luo, B., and Hancock, E. R. 2001. Structural Graph Matching Using the EM Algorithm and Singular Value Decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23:1120–1136.

Parlett, B. N., and Landis, T. L. 1982. Methods for Scaling to Doubly Stochastic Form. *Linear Algebra and its Applications* 48:53–79.

Quadrianto, N.; Smola, A. J.; Song, L.; and Tuytelaars, T. 2010. Kernelized Sorting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32:1809–1821.

Sinkhorn, R., and Knopp, P. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics* 21(2):343–348.

Smola, A.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A Hilbert Space Embedding for Distributions. *Algorithmic Learning Theory* 4754(x):1–20.

Tripathi, A.; Klami, A.; Orešič, M.; and Kaski, S. 2011. Matching samples of multiple views. *Data Mining and Knowledge Discovery* 23:300–321.

Wang, C., and Mahadevan, S. 2009. Manifold Alignment without Correspondence. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, IJCAI'09, 1273–1278. Morgan Kaufmann Publishers Inc.

Wang, H.-Y., and Yang, Q. 2011. Transfer Learning by Structural Analogy. In Burgard, W., and Roth, D., eds., *AAAI Conference on Artificial Intelligence*. AAAI Press.

Yamada, M., and Sugiyama, M. 2011. Cross-Domain Object Matching with Model Selection. *Journal of Machine Learning Research - Proceedings Track* 15:807–815.