Using KCCA for Japanese-English cross-language information retrieval and classification

Yaoyong Li[*] and John Shawe-Taylor[†]

Abstract

Kernel Canonical Correlation Analysis (KCCA) is a method of correlating linear relationship between two multidimensional variables in feature space. We applied the KCCA to the Japanese-English cross-language information retrieval and classification. The results were encouraging.

1 Introduction

Proposed by H. Hotelling in [4], Canonical Correlation Analysis (CCA) is to find basis vectors for two sets of variables such that the correlation between the projections of the variables onto these basis vectors are mutually maximised. CCA can be seen as using complex labels as a way of guiding feature selection towards underlying semantics. CCA makes use of two views of the same semantics object to extract the representation of the semantics. In an attempt to increase the flexibility of the feature selection, kernelisation of CCA (KCCA) has been applied to map the hypotheses to a higher-dimensional feature space.

The KCCA is particularly suitable to the applications where the semantics of the object with two or more views are crucial. Two such problems are cross-language information retrieval and multimedia content based retrieval. Actually the KCCA has achieved state of the art results for the two problems, respectively (ref [3] and [7]). In this paper we present the results by using the KCCA for Japanese-English cross-language information retrieval. There were two motivations to do this kind of experiments. One was that, as the KCCA had been used successfully to infer a good semantic representation from English-French bilingual corpus (ref [7]), we wanted to check whether the KCCA could do the similar work for two big different languages like English and Japanese. Furthermore, we want to see how good of the KCCA for cross-language patent retrieval.

We also present the results for cross-language classification. The cross-language classification means that, once obtaining a classifier for some classification problem in one language, we try to project the classifier onto another language for the same classification problem .

The KCCA for cross-language application is formulated in Section 2. Section 3 and 4 present the experiments for cross-language information retrieval and classification, respectively.

2 Kernel canonical correlation analysis for cross-language text applications

The KCCA can be used to infer a semantic representation of text from bilingual corpus. This kind of semantic representation can then be used for cross-language text applications

[*] Department of Computer Science, Royal Holloway University of London

[†] ISIS Group, School of Electronics and Computer Science, University of Southampton

such as information retrieval and classification. In the following we will explain how the KCCA works for the cross-language applications.

Suppose we are given N pairs of documents in two languages, i.e. every document d_i $(i = 1, \ldots, N)$ in one language is a translation of document c_i in another language. After some preprocessing, we obtain a feature vector $x_i \in \mathcal{X}$ for every document d_i and a feature vector $y_i \in \mathcal{Y}$ for document c_i, where $\mathcal{X}$ and $\mathcal{Y}$ are the feature spaces of the two languages, respectively. By using the canonical correlation analysis (CCA), we can find directions $f_x \in \mathcal{X}$ and $f_y \in \mathcal{Y}$ in the two spaces so that the projections $\{(f_x, x_i)\}_{i=1}^N$ and $\{(f_y, y_i)\}_{i=1}^N$ of the feature vectors of documents from the two languages would be maximal correlated. The pair of directions f_x and f_y represents a good correspondence between two languages and can be used for cross-language applications.

Formally, the CCA is to find a canonical correlation ρ in the space $\mathcal{X} \times \mathcal{Y}$ which is defined as

$$\begin{aligned} \rho &= \max_{(f_x,f_y)\in\mathcal{X}\times\mathcal{Y}} \mathrm{corr}((f_x, x_i), (f_y, y_i)) \\ &= \max_{(f_x,f_y)\in\mathcal{X}\times\mathcal{Y}} \frac{\sum_{i=1}^N (f_x, x_i)(f_y, y_i)}{\sqrt{\sum_i (f_x, x_i)^2 \sum_j (f_y, y_j)^2}} \end{aligned} \tag{1}$$

We search for f_x and f_y in the space spanned by the corresponding feature vectors, i.e. $f_x = \sum_l \alpha_l x_l$ and $f_y = \sum_m \beta_m y_m$. This rewrites the numerator of (1) as

$$\sum_i (f_x, x_i), (f_y, y_i) = \sum_i \sum_{lm} \alpha_l \beta_m (x_l, x_i)(y_m, y_i) = \alpha^T K_x K_y \beta \tag{2}$$

where α is the vector with components α_l $(l = 1, ..., N)$ and β the vector with components β_m $(m = 1, ..., N)$ and K_x is the Gram matrix of $\{x_i\}_{i=1}^N$ and K_y the Gram matrix of $\{y_j\}_{j=1}^N$. The problem (1) can then be reformulated as

$$\rho = \max_{\alpha,\beta} \frac{\alpha^T K_x K_y \beta}{\sqrt{\alpha^T K_x^2 \alpha \cdot \beta^T K_y^2 \beta}} \tag{3}$$

In order to force non-trivial learning on the correlation, we introduce a regularisation parameter to penalise the norms of the associated weights. By doing so, the problem (3) becomes

$$\rho = \max_{\alpha,\beta} \frac{\alpha^T K_x K_y \beta}{\sqrt{(\alpha^T K_x^2 \alpha + \kappa \alpha^T \alpha) \cdot (\beta^T K_y^2 \beta + \kappa \beta^T \beta)}} \tag{4}$$

Note that the new regularised equation is not affected by re-scaling of α or β, hence the optimisation problem is subject to the two constraints

$$\alpha^T K_x^2 \alpha + \kappa \alpha^T \alpha = 1 \tag{5}$$

$$\beta^T K_y^2 \beta + \kappa \beta^T \beta = 1 \tag{6}$$

The corresponding Lagrangian is

$$L(\lambda_\alpha, \lambda_\beta, \alpha, \beta) = \alpha^T K_x K_y \beta - \frac{\lambda_\alpha}{2}(\alpha^T K_x^2 \alpha + \kappa \alpha^T \alpha - 1) - \frac{\lambda_\beta}{2}(\beta^T K_y^2 \beta + \kappa \beta^T \beta - 1)$$

Taking derivatives of the Lagrangian with respect to α and β and setting them to be zero, respectively, we have the equations

$$K_x K_y \beta - \lambda_\alpha (K_x^2 + \kappa I)\alpha = 0 \tag{7}$$

$$K_y K_x \alpha - \lambda_\beta (K_y^2 + \kappa I)\beta = 0 \tag{8}$$

According to the *Kuhn-Tucker* Theorem, the solution (α^*, β^*) of the equations (7) and (8) is the solution of the optimisation problem (4) with the constraints (5) and (6). Let α^T times the equation (7) we have

$$\alpha^T K_x K_y \beta - \lambda_\alpha \alpha^T (K_x^2 + \kappa I)\alpha = 0$$

which together with (5) implies that

$$\lambda_\alpha = \alpha^T K_x K_y \beta$$

Similarly let β^T times the equation (8) and together with the constraint (6) we have

$$\lambda_\beta = \beta^T K_y K_x \alpha$$

The above two equations implies that

$$\lambda_\alpha = \lambda_\beta = \alpha^T K_x K_y \beta \tag{9}$$

Let $\lambda = \lambda_\alpha = \lambda_\beta$, we can rewrite the equations (7) and (8) as a generalised eigenvalue problem

$$B\xi = \lambda D\xi \tag{10}$$

where λ is the canonical correlation ρ between projections (f_x^*, x_i) and (f_y^*, y_i) $(i = 1, ..., N)$, and

$$B = \begin{pmatrix} 0 & K_x K_y \\ K_y K_x & 0 \end{pmatrix}, \; D = \begin{pmatrix} K_x^2 + \kappa I & 0 \\ 0 & K_y^2 + \kappa I \end{pmatrix}, \; \xi = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \tag{11}$$

So, the optimisation problem of the CCA was transformed into a generalised eigenvalue problem (10), where the eigenvectors with the largest eigenvalues represent the maximally correlated directions in feature space.

We can see that, either in the optimisation problem (4), (5) and (6) or in the eigenproblem (10), the training points $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ are involved only through the Gram matrix K_x and K_y. Therefore, the so-called "kernel-trick" can be used to introduce extra flexibility into CCA. Kernelisation of CCA means that the training points $\{x_i\}_{i=1}^N$ and $\{y_i\}_{i=1}^N$ are mapped to another (some high-dimensional) feature space by a kernel function (ref [2]) and the canonical correlation is then computed in the new feature space. This can be done easily by replacing the Gram matrixes with the corresponding Kernel matrixes in the optimisation formulation (4), (5) and (6) and the eigenproblem (10). However, the experiments in [7] showed that the linear kernel was quite good for cross-language applications of the KCCA. Hence, only the linear kernel was used in our experiments. Also, we used the same value of regularisation parameter as in [7], i.e. $\kappa = 1.5$.

3 Using KCCA for Japanese-English cross-language information retrieval

Cross-language information retrieval with KCCA. The KCCA leads to a generalised eigenvalue problem. The eigenvectors with the largest eigenvalues correspond to the maximally correlating directions in the feature spaces of two languages. These eigenvectors constitute some kind of semantic correspondence between the training documents of two languages, which provides a framework for performing cross-language information retrieval where, given a query in one language, we try to find out the relevant documents in another language. We first select a number d of eigenvectors with largest eigenvalues from the solution of (10), which form a common semantic space for the two language. To process a query q we represent q as a feature vector $\tilde{q}$ and project it onto the d canonical correlation directions in feature space

$$\tilde{q}_d = A^T Z^T \tilde{q} \tag{12}$$

where A is $N \times d$ matrix whose columns are the first or the second half (depending on which language was used in query) of eigenvectors of (10) with the largest d eigenvalues, and each column of Z is a training vector in the same language as query. Similarly, we represent every documents in another language for retrieval as d-dimensional vectors by projecting them onto the d-dimensional canonical correlation directions. Then the documents with the shortest distances to the query in the d-dimensional space are regarded as being relevant to the query.

The dataset for experiment. The dataset we used was from the NTCIR-3 patent retrieval test collection[1]. The collection includes about 1.7 millions of Japanese patent abstracts and their English translations, spanned over five years (1995–99). Only the 336,929 documents in 1995 (referred as the 1995 collection thereafter) was used in the experiments we did. First of all, we collected the terms and computed the idf (inverse document frequency) for every term from the 1995 collection. The English terms were collected in the usual way, i.e. down-casing the alphabetic characters, removing the stop words, replacing every no-alphabetic character with a blank, stemming words by the Porter stemmer, and finally removing the terms which appears less than 3 times in the corpus. We preprocessed the Japanese documents using a Japanese morphological analysis software Chasen version 2.3.3 [2]. From the documents processed by the Chasen, we picked up as our terms those words the part of speech tags of which are noun (but not dependent noun, proper noun or number noun), independent verb, independent adjective, or unknown. We also removed the Japanese terms appearing less three times in the documents of the 1995 collection. By doing so, 61583 English terms and 90055 Japanese terms were obtained, respectively. Then we computed the tf*idf feature vectors for the Japanese patent abstracts and the corresponding English translations in the usual way.

Mate retrieval. We first conducted experiments for mate retrieval. In mate retrieval a document in one language was treated as a query and only the mate document in another language was considered as relevant. A mate document was considered to be retrieved if it is most close to the query document in the semantic space. We applied the KCCA to the first 1000 Japanese documents and the English translations of the 1995 collection. For comparison, we also implemented the LSI for cross-language information retrieval (see [6]) under the same experimental settings .

The results presented in the first part of Table 1 is for 1000 training documents as queries. These results are consistent with those on the English-French documents (see [7]). That is, the KCCA can achieve quite good performance using a fraction of eigenvectors (say 200) while the LSI achieved same results only when using full 1000 eigenvectors. The second part of Table 1 shows the results for the queries from other 2000 test documents. The results looks good, though still could be improved. And the KCCA outperformed the LSI on test documents significantly.

Pseudo query retrieval. We also did experiments for pseudo query retrieval. We generated a short query consisting of the five most probable words for each test document. And the relevant document is the mate of the document in another language. Table 2 shows the relative number of correctly retrieved documents in each experimental setting. Once again, we present the results for the queries from the 1000 training documents and the 2000 test documents, respectively. The retrieval accuracy of KCCA is high and is better than those from the LSI when a short query was generated from training document. The accuracy dropped when the short query was from test document (not in training set).

[1] See http://research.nii.ac.jp/ntcir/permission/perm-en.html

[2] See http://chasen.aist-nara.ac.jp/

Table 1: Mate retrieval (1000 training documents): the accuracy rates averaged over all the training documents and over other 2000 test documents, respectively. Different numbers of the eigenvectors were used and the KCCA was compared with the LSI.

#Eigenvectors	5	10	50	100	200	300	400	500	1000
Training docs as queries									
KCCA(E-J)	0.678	0.896	0.978	0.988	0.993	0.994	0.993	0.991	0.996
KCCA(J-E)	0.664	0.878	0.973	0.981	0.988	0.988	0.989	0.986	0.997
LSI(E-J)	0.093	0.328	0.769	0.898	0.949	0.96	0.965	0.966	0.996
LSI(J-E)	0.091	0.264	0.652	0.827	0.923	0.946	0.952	0.959	0.996
Test docs as queries									
KCCA(E-J)	0.050	0.154	0.401	0.471	0.534	0.528	0.506	0.486	0.377
KCCA(J-E)	0.084	0.173	0.369	0.448	0.461	0.430	0.398	0.369	0.272
LSI(E-J)	0.037	0.095	0.296	0.376	0.431	0.431	0.417	0.393	0.247
LSI(J-E)	0.029	0.079	0.212	0.294	0.362	0.355	0.329	0.304	0.170

Table 2: Pseudo query retrieval (1000 training documents): the accuracy rates averaged over all the training documents and over other 2000 test documents, respectively. Different numbers of the eigenvectors were used and the KCCA was compared with the LSI.

#Eigenvectors	5	10	50	100	200	300	400	500	1000
Training docs as queries									
KCCA(E-J)	0.114	0.323	0.688	0.819	0.915	0.935	0.943	0.942	0.961
KCCA(J-E)	0.129	0.319	0.615	0.762	0.876	0.901	0.910	0.915	0.933
LSI(E-J)	0.062	0.170	0.415	0.561	0.734	0.785	0.829	0.862	0.911
LSI(J-E)	0.048	0.128	0.244	0.317	0.433	0.495	0.528	0.539	0.548
Test docs as queries									
KCCA(E-J)	0.024	0.068	0.168	0.198	0.219	0.229	0.230	0.230	0.205
KCCA(J-E)	0.029	0.060	0.134	0.159	0.164	0.157	0.152	0.143	0.116
LSI(E-J)	0.028	0.077	0.152	0.186	0.203	0.212	0.220	0.211	0.172
LSI(J-E)	0.023	0.061	0.114	0.137	0.140	0.140	0.133	0.126	0.093

The experimental results have shown that the KCCA outperformed the LSI consistently and significantly for cross-language information retrieval. We can also see that the similar results were obtained for the English-Japanese bilingual corpus as for English-French. However, comparing with the high retrieval accuracy when training documents as queries, the retrieval accuracy is low when the documents not used in training as queries. This may be due to a small number of training documents we used. By the KCCA we extracted a semantic correspondence between two languages from the training documents. If the training documents is too small to be representative, then the semantic correspondence is not good in general.

More training documents. We expected to have better generalisation performance when the training set become larger. So we did additional experiments by enlarging the training set to 2000 documents. In the case of training documents as queries, the results for 2000 training documents were similar to those for 1000 training documents. The results for the 2000 test documents as queries are presented in Table 3. Comparing with the corresponding results in Table 1 and Table 2, we can see from Table 3 that the generalisation performance was improved indeed when using more training documents, though the accuracy is still much less than those for training documents as queries.

Discussions. It is possible that the generalisation performance of the KCCA will become

Table 3: Results of experiments with the 2000 training documents: the accuracy rates averaged over 2000 test documents. Different numbers of the eigenvectors were used and the KCCA was compared with the LSI.

#Eigenvectors	**5**	**10**	**50**	**100**	**200**	**300**	**400**	**500**	**1000**
Mate retrieval									
KCCA(E-J)	0.135	0.244	0.567	0.611	0.641	0.646	0.633	0.614	0.489
KCCA(J-E)	0.161	0.287	0.525	0.573	0.591	0.572	0.542	0.506	0.369
Pseudo query retrieval									
KCCA(E-J)	0.063	0.102	0.211	0.249	0.274	0.286	0.298	0.302	0.278
KCCA(J-E)	0.054	0.090	0.170	0.194	0.205	0.218	0.215	0.206	0.171

further better if we use more training documents. However, we are unable to use very large training set for the KCCA because the time would become very long when using for example 50,000 documents for training. Two approximation approaches, the incomplete Cholesky decomposition and the partial Gram-Schmidt othogonolisation, were proposed (in [1] and [3], respectively) to tackle the problem of computational complexity.

Another possible way for dealing with large training set is to split training set into some relatively small subsets and apply the KCCA to each subset independently and then integrate the solutions of the KCCA from the subsets into a general semantic correspondence between two languages. We think that clustering a large training set may be better for the KCCA to handle large dataset than splitting it randomly into small groups. Clustering a large dataset not only results in some relatively small training sets for the KCCA. It also can put together the documents with similar contents. Hence, the semantic correspondence extracted by the KCCA from the cluster could be a good semantic representation of the cluster. In addition, the semantic correspondences extracted from different clusters are expected to compensate for each other so that it may be fruitful to integrate the semantic representations from different clusters into a general one. We know that clustering itself is a current research topic. However, we do not think the performance is much dependent on the clustering algorithm, as the application of KCCA to each cluster would be noise tolerable. Unfortunately, we have not done experiment to compare clustering with other approaches such as partial Gram-Schmidt othogonolisation for the KCCA to dealing with very large dataset.

Another problem we can see from the results of our experiments is that the accuracy of retrieving English documents by Japanese query (from Japanese to English) is lower than the one from English to Japanese in almost all cases . This was probably because the quality of the Japanese terms we collected was not as good as the quality of English terms. Hence, one work we could do is to improve the procedure of Japanese term collection.

4 Cross-language document classification

The NTCIR-3 patent retrieval test collection includes 31 topics as well as some documents annotated for each topic. The annotated documents for a topic form a classification problem, where the relevant documents are the positive examples and the unrelevant documents are the negative examples. By using the documents, we have done the experiments for the cross-language classification where a classifier was learned from documents in one language and then was used to classify documents in another language.

We proposed two methods to induce an SVM classifier for cross-language classification. One was to directly use pairs of training documents between two languages, i.e. $\{(x_i, y_i) : i = 1, ..., N\}$. An SVM was first learned from the training documents $\{x_i :$

$i = 1, ..., N\}$ in one language, which can be represented in dual form as

$$h_x(\cdot) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i K(\cdot, x_i)\right) \tag{13}$$

we can then obtain an SVM classifier in another language

$$h_y(\cdot) = \text{sgn}\left(\sum_{i=1}^{N} \alpha_i K(\cdot, y_i)\right) \tag{14}$$

We call the new SVM classifier (14) as *pSVM* since it was deduced by using the pairness of the training document.

Another method involved the KCCA. We first obtained a semantic correspondence between two languages using the KCCA. Given a training set containing pair of documents in both languages, projecting the training documents onto the semantic space resulted in pairs of semantic feature vectors. These pairs of semantic vectors were used to project an SVM classifier in one language into another language, just as what was done for *pSVM*. We call this kind of classifier as *kcca_SVM*. Note the training set for KCCA may be different from that for learning the SVM. This implies that a large (unlabelled) training set can be used for KCCA to deduce a good semantic correspondence between two languages and another labelled document set would be employed for the SVM training. However, in the experiments described below, only one training set was used for both the KCCA and the SVM.

In our experiments, pairs of documents in Japanese and English annotated for a topic in the NTCIR-3 patent retrieval test collection form the dataset for a cross-langauge classification. we randomly splitted the dataset into two parts with the same number of documents — one for training and another for test. We used the English part of training documents to train an SVM classifier, which then induced *pSVM* and *kcca_SVM* classifiers in Japanese documents. Averaged precision was used to evaluate the performances of the SVM classifiers[3]. Table 4 show the results for the cross-language classification from two topics, Topic 01 and Topic 07 in the NTCIR collection. Topic 01 has 837 annotated documents, 26 of which are relevant. In contrast, Topic 07 has 366 annotated documents and 102 relevant documents. So Topic 01 and Topic 07 represent small and large topics, respectively. For the *kcca_SVM* we present the results with different number of eigenvectors from KCCA. We can see that the induced SVM classifier can achieve similar performance in the Japanese documents with the original SVM classifier in English documents for both topics. Hence the SVM had quite good behaviour for the cross-language classification.

5 Conclusions

We used the KCCA for the cross-language Japanese-English information retrieval. The experimental results are quite encouraging. We also presented two methods for cross-language classification. One was to project the SVM classifier in one language onto another language directly through the pairs of training documents in two langauge. Another was to induce an SVM classifier in another language by the semantic correspondence of the KCCA. Both methods achieved good results.

[3] We did not use the F_1, a commonly used measure in formation retrieval research, to measure the performance. The F_1 is dependent on the bias b of the SVM solution but the average precision is not. It is known that the SVM would learn a poor bias if the number of positive training pattern is very small and the bias can be improved by some algorithms (ref [5]). Hence, averaged precision is a better measure than F_1 for the SVM.

Table 4: Averaged precisions: results of cross-language classification for two problems. The SVM classifiers were learned from English training set and induced *pSVM* and *kcca_SVM* classifiers in Japanese. We present the averaged precisions of the SVM on English training and test sets as well as the results of the induced classifiers on Japanese training and test sets. For the *kcca_SVM*, results were presented for different numbers of eigenvectors from KCCA.

	pSVM	**KCCA_50**	**KCCA_100**	**KCCA_150**	**KCCA_full**
Topic 01					
E_tr	1.000	0.781	0.977	0.992	1.000
E_te	0.453	0.360	0.410	0.444	0.469
J_tr	0.941	0.794	0.925	0.984	0.992
J_te	0.499	0.411	0.424	0.489	0.491
Topic 07					
E_tr	0.956	0.876	0.939	0.958	0.971
E_te	0.827	0.851	0.874	0.870	0.879
J_tr	0.936	0.874	0.929	0.954	0.968
J_te	0.769	0.772	0.777	0.773	0.784

References

[1] F. R. Bach and M. I. Jordan. Kernel indepedendent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

[2] Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[3] D. R. Hardon, S. Szedmark, and J. Shawe-Taylor. Canonical correlation analysis: an overview with application to learning methods. Technical Report CSD-TR-03-02, Department of Computer Science, Royal Holloway, University of London, 2003.

[4] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:312–377, 1936.

[5] Yaoyong Li and John Shawe-Taylor. he svm with uneven margins and chinese document categorization. In *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*, pages 216–227, Singapore, Oct. 2003.

[6] M. L. Littman, S. T. Dumais, and T. K. Landauer. Automatic cross-language information retrieval using latent semantic indexing. In G. Grefenstette, editor, *Cross language information retrieval*. Kluwer, 1998.

[7] Alexei Vinokourov, John Shawe-Taylor, and Nello Cristianini. Inferring a semantic representation of text via cross-language correlation analysis. In *Advances of Neural Information Processing Systems 15*, 2002.