THE SIXTH FRAMEWORK PROGRAMME
The sixth framework programme covers Community activities in the field of research, technological development and demonstration (RTD) for the period 2002 to 2006

SMART

Statistical Multilingual Analysis for Retrieval and Translation

D 5.2 MULTILINGUAL LATENT LANGUAGE-INDEPENDENT ANALYSIS METHODS APPLIED TO CLTIA TASKS

Version 1.0
24/March/2008

Executive Summary

VERSION	DATE	AUTHOR
01	24/03/2008	John Shawe-Taylor

TITLE	D 5.2 Multilingual Latent language-independent Analysis methods applied to CLTIA tasks
STATE	Final
CONFIDENTIALITY	PU
AUTHOR(S)	J.S. Taylor, Blaz Fortuna, David Hardoon, Jan Rupnik, Zakria Hussain, Jean-Michel Renders
PARTICIPANT PARTNERS	UCL, University of Southampton, JSI, Helsinki, XRCE
WORKPACKAGE	WP5
ABSTRACT	The Deliverable considers methods of inferring Latent Language-independent representations as the basis for improved cross-lingual text information access (CLTIA) and processing. The representations are derived using extensions of a classical correlation technique known as Canonical Correlation Analysis (CCA) that has established itself as a well-founded method for cross-lingual information retrieval. The extensions aim to achieve three main goals: 1) to extend the approach beyond paired corpora to multi-lingual aligned datasets, 2) to improve the scalability of the algorithms beyond current limits of tens of thousands of documents, and 3) to tune the algorithms and representations to the specific statistics of text data. The extensions to multi-lingual aligned corpora are presented in Chapter 3, where an algorithm is developed that can also scale to very much larger numbers of training documents. Chapter 4 presents methods that seek sparse solutions better tuned to the statistics of text data and which also scale well beyond standard CCA. Finally, the developed techniques are tested on CLTIA tasks to test their efficiency and efficacy in real-world applications. The achieved levels of accuracy match and in many cases extend the state of the art, while the scaling characteristics are in all cases an order of magnitude better than CCA, and in some cases even two or more orders of magnitude.
KEYWORDS	Latent Semantic Analysis, Sparse Kernel Methods, Cross-language Information Retrieval, N-way Canonical Correlation Analysis
REVIEWERS	Jean-Michel Renders, Nicola Cancedda

Contents

Chapter 1

Introduction

The SMART project sets as one of its goals the extension of Canonical Correlation Analysis inspired approaches to CLTIA. The deliverable has focussed on three key extensions:

1. the consideration of aligned corpora involving more than two languages;
2. the scalability of the algorithms beyond current limits of tens of thousands of documents;
3. the tuning of the algorithms and representations to the specific statistics of text data.

The motivation for 2. is clear since increasingly datasets are scaling well beyond the sizes that admit CCA analysis. Point 1 has the potential to create richer and more widely applicable representations, but also to combine subsets of aligned corpora in different languages to infer a unified representation. Such representations could be used in many different processing tasks such as cross-lingual topic classification and clustering of multi-lingual corpora. The final point is concerned with the possible mismatch between many standard statistical methods that rely on the assumption that Gaussian distributions are good approximators of the underlying term frequency distributions. This is known not to be the case with term frequencies widely agreed to follow a power law. TFIDF weightings can be seen as a way of overcoming this difficulty by downweighting the high frequency terms, but it would be preferable to develop more principled methods that better model the underlying distributions of text production. One approach relies on using sparse representations to capture the low-frequency phenomena, and this is a key point of this deliverable.

But let us recall first the philosophy behind using Latent Analysis. Latent Analysis, usually based on co-occurrence analysis, is an important component of modern information retrieval system. It can be used to cope with synonyms, related words and disambiguation or in other words, to reduce the problem of sparsity in the information retrieval – making the retrieval system capable of retrieving documents which on a semantic level match the query, but have no words in common with it.

When used on aligned corpora, that is a collection of documents where for each document we have its translation in two or more languages, it can also be used to bridge the language gap. To this end, information of the word co-occurrence across the document translations is used (e.g. that word in one language frequently co-occurs with a set of words in corresponding documents in some other language). Most commonly used method for multilingual latent analysis is Canonical Correlation Analysis (CCA).

Traditionally, bilingual lexicons are used for the task of cross-lingual information retrieval (CLIR). However, multilingual latent analysis can be used either to enrich the lexicons or to provide support for matching query to the documents which share no or little words with the query but are semantically similar to it.

In this deliverable we address several aspects of Latent Analysis in multilingual setting. First, we introduce a generalization of CCA to more than two languages and give an efficient algorithm which

can scale linearly both with the number of languages and the number of documents. The second aspect, sparse representation, is motivated by the fact that such representations are better able to capture low freqency phenomena as one approach to this problem. The sparsity is typically introduced using a 1-norm constraint either on the features (words) involved in a topic or its dual representation in terms of a small subset of documents. The primal-dual sparsity methods introduced in Section 4.3 combines these two approaches while sparse KCCA (SKCCA) introduced in section 4.2 focuses on dual sparsity. Both methods are developed with an eye on efficiency considerations ensuring that either the optimisation itself or well-motivated approximations to it can be implemented in algorithms that scale to very large dataset sizes.

Chapter 2

A short Reminder on CCA

2.1 Vector space model

The classic representation of a text document in Information Retrieval [17] is as Bag of Words (a bag is a set where repetitions are allowed), also known as Vector Space Model, since a bag can be represented as a (column) vector recording the number of occurrences of each word of the dictionary in the document at hand. A document is represented, in the vector-space model, by a vertical vector $\mathbf{d}$ indexed by all the elements of the dictionary (i-th element from the vector is the frequency of i-th term in the document TF_i). A corpus is represented by a matrix D, whose columns are indexed by the documents and whose rows are indexed by the terms, $D = (\mathbf{d}_1, \ldots, \mathbf{d}_\ell)$. We also call the data matrix D the **term-document** matrix.

Example 1. *Suppose that our corpus is comprised of the following Slovene documents:*

1. TAJNA OPORISCA CIE PO VSEM SVETU,
2. CASTRO POSMEHLJIVO ZAVRNIL CIA ,
3. FRATTINI: DOKAZOV O ZAPORIH CIE NI,
4. CIA IZKORISCALA SPANSKA LETALISCA.

First we enumerate the words that appear in the documents. The numbers assigned to a word will correspond to the row with the counts of this word in the term-document matrix:

t_1 = *tajna*	t_2 = *oporisca*	t_3 = *cie*	t_4 = *po*	t_5 = *vsem*
t_6 = *svetu*	t_7 = *castro*	t_8 = *posmehljvo*	t_9 = *zavrnil*	t_{10} = *cia*
t_{11} = *frattini*	t_{12} = *dokazov*	t_{13} = *o*	t_{14} = *zaporih*	t_{15} = *ni*
t_{16} = *izkoriscala*	t_{17} = *spanska*	t_{18} = *letalisca*		

Forming the term-document matrix we get:

$$D' = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

■

The classic similarity measure in used in Information Retrieval is **cosine similarity** [17]:

$$sim(\mathbf{d}_i, \mathbf{d}_j) = \frac{\mathbf{d}_i' \mathbf{d}_j}{\|\mathbf{d}_i\| \|\mathbf{d}_j\|} = \cos\alpha, \tag{2.1}$$

where α is the angle between the two vectors.

Since not all terms are of the same importance for determining similarity between the documents we introduce term weights. A term weight corresponds to the importance of the term for the given corpus and each element from the document vector is multiplied with the respective term weight. The most widely used weighting is called TFIDF weighting. A **IDF weight** for term i from the dictionary is defined as $\log(\ell/\mathrm{DF}_i)$ where DF_i is the number of documents from the corpora which contain word i. A document's **TFIDF vector** is a vector with elements: $\mathrm{TF}_i \log(\ell/\mathrm{DF}_i)$. The acronyms IDF and TFIDF stand for "Inverse Document Frequency" and "Term Frequency Inverse Document Frequency" respectively.

We will consider linear combinations of terms as "concepts". Concepts offer a for of abstraction, independent of the problem with surface form of words, such as polysemy and synonymy. The magnitude of the coefficients of a term in a given combination could be interpreted as the level of membership of that given term to the concept. For the application point of view, these could be interpreted as generalized versions of sets of terms. Geometrically, we will interpret them as directions in the term-space.

Example 2. *Here are a couple of examples of concepts, as extracted using Latent Semantic Indexing (LSI) method from the GIRT corpora; for each concept, a list of highest weighted words together with their weights is given.*

- [WOMAN:0.36] [STUDY:0.27] [GENDER:0.18] [EMPLOYMENT:0.12] [FEMINIST:0.12] [EDUCATION:0.12] [OCCUPATIONAL:0.11] [LABOR:0.10] [FAMILY:0.10] [WORK:0.09]
- [TECHNOLOGY:0.31] [INDUSTRIAL:0.24] [SCIENCE:0.22] [SOCIOLOGY:0.13] [PHILOSOPHY:0.13] [WORK:0.12] [ETHICS:0.11] [ENGINEER:0.10] [CULTURAL:0.10] [MANAGEMENT:0.09]
- [POPULATION:0.24] [AFRICA:0.20] [AMERICA:0.18] [COUNTRY:0.16] [DEVELOP:0.14] [TECHNOLOGY:0.12] [FAMILY:0.11] [ASIA:0.10] [MIGRATION:0.09] [SOUTH:0.08]

In order to introduce some semantic information to the document representation, we can consider linear transformations (feature mapping) of the document vectors of the type

$$\phi(\mathbf{d}) = A\mathbf{d},$$

where A is any matrix. If we view the rows of the matrix A as "concepts", than the elements of the new document vector $\phi(\mathbf{d})$ are similarities to these concepts. We can define a kernel using feature mapping ϕ as

$$K(\mathbf{d}_i, \mathbf{d}_j) = \mathbf{d}_i A A' \mathbf{d}'_j.$$

Recall that, for two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$, and for any linear mapping denoted by the matrix A, the function $K_A(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i A A' \mathbf{x}'_j$ is always a kernel. The feature space defined by the mapping $\phi(\mathbf{d}) = A\mathbf{d}$ is called **semantic space**.

Given two sets of documents, in two different languages, we will see them as sets of points in two different spaces, provided with a one-to-one relation. We will exploit correlations between the two sets in order to learn a better embedding of the documents into a space, where semantic relations are easier to detect. By detecting correlated directions between the two spaces, we will detect groups of terms (concepts) that have a similar pattern of occurrence across the two corpora, English and French, and hence can be considered as semantically related.

2.2 Definition of CCA

Canonical Correlation Analysis (CCA) [5] is a statistical method for extracting correlations between a pair of random variables. For example, given a paired bilingual corpus (a set of pairs of documents, each

pair being formed by two versions of the same text in two different languages), this method defines two embedding spaces for the documents of the corpus, one for each language, and an obvious one-to-one correspondence between points in the two spaces. KCCA then finds projections in the two embedding spaces for which the resulting projected values are highly correlated. In other words, it looks for particular combinations of words that appear to have the same co-occurrence patterns in the two languages. Our hypothesis is that finding such correlations across a paired cross-lingual corpus will locate the underlying semantics, since we assume that the two languages are 'conditionally independent', or that the only thing they have in common is their meaning. The directions would carry information about the concepts that stood behind the process of generation of the text and, although expressed differently in different languages, are, nevertheless, semantically equivalent.

The following list gives a series of example of extracted concepts, which appear semantically equivalent based on the provided aligned corpus. The corpus used in this example was the corpus of Canadian parliament proceedings called Hansard.

ENGLISH	health, minister, quebec, prime, he, we, federal, hepatitis, provinces, not, victims, they, care, are, people, resources
FRENCH	sante, ministre, quebec, nous, premier, federal, provinces, hepatite, ils, pas, developpement, il, canadiens, dollars, leur
ENGLISH	motion, house, agreed, standing, bill, adams, members, commons, order, parliamentary, leader, peter, consent, petitions
FRENCH	motion, chambre, leader, voix, adams, communes, loi, accord, parlementaire, peer, reglement, conformement, secretaire
ENGLISH	motion, bill, division, members, act, declare, paired, nays, deferred, no, yeas, vote, stage, reading, c, carried, proceed
FRENCH	motion, loi, vote, deputes, projet, no, paires, consentement, adoptee, o, voix, unanime, votent, motions, rejetee, lecture
ENGLISH	petitions, standing, response, parliamentary, languages, honour, secretary, pursuant, adams, official, table, report, both
FRENCH	petitions, honneur, reponse, langues, officielles, parlementaire, reglement, deposer, secretaire, paragraphe, adams

Such directions can then be used to calculate the coordinates of the documents in a 'language independent' way, and this representation can be used for retrieval tasks or for other purposes such as clustering or categorization. Of course, particular statistical care is needed for excluding 'spurious' correlations which is done by introducing regularization parameters.

More formally, in standard CCA we assume that we are given sample data

$$S = \left((X_1^1, X_1^2), \ldots, (X_n^1, X_n^2)\right) \subset \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$$

that is drawn independently at random according to some underlying distribution. The two parts of every observation correspond to two representations of the same underlying object. We can think of them as two multivariate random variables $\mathcal{X}_1$ and $\mathcal{X}_2$ with some mutual information. The idea is to find patterns in both views that represent this mutual information. One way of doing this is to find two functions

$$\phi_1 : \mathbb{R}^{n_1} \to \mathbb{R}, \quad \phi_2 : \mathbb{R}^{n_2} \to \mathbb{R},$$

so that the univariate random variables $\phi_1(\mathcal{X}_1)$ and $\phi_2(\mathcal{X}_2)$ are maximally correlated. If $\phi_i(\mathcal{X}_i)$ has zero mean, then we are looking for ϕ_1 and ϕ_2 that maximize the empirical correlation:

$$corr(\phi_1(\mathcal{X}_1), \phi_2(\mathcal{X}_2)) = \frac{\sum_i \phi_1(X_i^1)\phi_2(X_i^2)}{\sqrt{\sum_i \phi_1(X_i^1)^2 \sum_i \phi_2(X_i^2)^2}}.$$

The class of candidate mappings proposed by Hotelling [5] is the class of linear mappings: $\phi(x) = w'x$, where w_i is a vector and we interpret w' as a linear functional codomain of random variable X. High correlation values between $\phi_1(\mathcal{X}_1)$ and $\phi_2(\mathcal{X}_2)$ indicate strong linear dependence. The resulting optimization problem

$$\max_{w_1, w_2} \frac{w_1' X^1 X^{2\prime} w_2}{\sqrt{w_1' X^1 X^{1\prime} w_1 w_2' X^2 X^{2\prime} w_2}}$$

can be solved as a generalized eigenvalue problem. If we rescale the vector $X^{1'}w_1$ to $cX^{1'}w_1$ we do not change the objective, so we can fix it to have unit norm: $w_1'X^1X^{1'}w_1 = 1$ and the same holds for w_2. We arrive at the constrained optimisation problem:

$$\max_{w_1,w_2} w_1'X^1X^{2'}w_2$$

s.t.

$$w_1'X^1X^{1'}w_1 = 1, \quad w_2'X^2X^{2'}w_2 = 1.$$

Writing the Lagrangian and setting the derivatives to be zero yields:

$$X^1X^{2'}w_2 = 2\lambda_1 X^1X^{1'}w_1,$$

$$X^2X^{1'}w_1 = 2\lambda_1 X^2X^{2'}w_2,$$

where λ_1 and λ_2 correspond to the multipliers of two constraints. Note that we can substitute $2\lambda_i$ with λ_i. Multiplying the first equation with w_1 and the second w_2 and using the original constraints shows that $\lambda_1 = \lambda_2$ and that the problem can be solved as a generalized eigenvalue problem:

$$\begin{bmatrix} 0 & X^1X^{2'} \\ X^2X^{1'} & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \lambda \begin{bmatrix} X^1X^{1'} & 0 \\ 0 & X^2X^{2'} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$$

Traditionally [4]), CCA was solved by first transforming it into the dual formulation where it was equivalent to solving a symmetric eigenvalue problem in the dual space. Since dimensionality of the dual space correspond to the number of the input data points and solving eigenvalue problems is of cube complexity with respect to the dimensionality, the complexity of the algorithm for solving CCA is $O(n^3)$, where n corresponds to the number of data points. This approach made running CCA over more than 10.000 data points practically infeasible on standard desktop computers. In this deliverable we introduce an improved version of CCA in Chapter 3 which scales linearly with respect to the number of data points, provided that the vectors corresponding to the data points have bounded number of dimensions.

Chapter 3

Short Synthesis of the State-of-the-Art

The use of latent models for cross-language retrieval was first proposed by [12]. Latent Semantic Indexing (LSI) uses a method from linear algebra, singular value decomposition, to discover important associative relationships. An initial sample of documents is translated by human or, perhaps, by machine, to create a set of dual-language training documents $D_x = \{\mathbf{x}_i\}_{i=1}^{\ell}$ and $D_y = \{\mathbf{y}_i\}_{i=1}^{\ell}$. After preprocessing the documents a common vector-space, including words from both languages, is created and the training set is analyzed in this space using SVD:

$$D = \begin{pmatrix} D_x \\ D_y \end{pmatrix} = U\Sigma V', \tag{3.1}$$

where the i-th column of D corresponds to the i-th document with its first set of coordinates giving the first language features and the second set giving the second language features. To translate a new document (query) $\mathbf{d}$ to a language-independent representation one projects (*folds-in*), its expanded (components related to another language are filled up with zero) vector representation $\mathbf{d}$ into the space spanned by the k first singular vectors U_k : $\phi(\mathbf{d}) = U_k'\mathbf{d}$. The similarity between two documents is measured as the inner product between their projections. The documents that are the most similar to the query are considered to be relevant. The described use of LSI for cross-lingual information retrieval is also know as Cross-Lingual Latent Semantic Indexing (CL-LSI);

In [15], Tatsunori proposed segmented CL-LSI, to increase the number of documents on which the method can be applied. The crucial step of the method was first splitting the aligned dataset into several areas, performing CL-LSI on each of the areas, and finally merging the discovered subspaces. This allowed application of CL-LSI approach to a corpus of 180.000 aligned Japanese-English documents.

Chapter 4

Multiview CCA

4.1 Motivations

As described in the second chapter, the standard CCA was already applied in cross-lingual information retrieval. In this chapter we focus on two of its weak points, namely the number of languages it can handle, and the size of the aligned training corpora it can handle. An approach for solving the multivariate eigenvalue problem (Horst algorithm) is used and adapted to solve large scale problems. The complexity of CCA was effectively reduced from cube to linear in the size of the training corpora, making the method much more appropriate for the real-world scenarios. The effectiveness of this approach is demonstrated in the experiments section, where we apply CCA to several large-scale datasets.

In order to do cross-lingual information retrieval using CCA one has to project the query and the whole multilingual corpus into the same space. In the classical CCA, one has to run CCA for each language pair resulting in several rankings (one for each language) which need to be merged. Advantage of multiview CCA (MCCA) is that it runs on all of the languages simultaneously, enabling us to rank the whole corpus of documents by their similarity to the query in the same space. There are other possible applications of MMCA - for example using the semantic space for cross-lingual topic classification. If the classification training data included a number of examples from each language, we could train a classifier on the complete set by projecting them all into the common semantic space learnt from a separate multi-lingual corpus. The resulting classifier could be used to classify documents from any of the languages. Using separate semantic spaces for each pair of languages would mean that a separate classifier would need to be found for each semantic space and only very few training examples would be available for training many of these classifiers. A similar disadvantage could apply to other tasks such as finding a common clustering of documents from a range of languages.

4.2 Problem Formulation

Defining a measure of cross-correlation for more than two random variables is less straightforward and many possible measures have been proposed [8]. There are some reasonable conditions that such a measure must fulfill. For instance for two views it should be equal to the correlation and should be maximised when there is perfect linear dependence amongst all random variables involved. Typical approaches define cross-correlation as a function of pairwise correlations between variables. The sum of correlations, sum of squared correlations, and product of correlations over all pairwise correlations are some examples that satisfy those conditions. Sum of correlations problem formulation, SUMCOR, was first studied by Horst [16]. We will adopt this approach. He formulated the optimisation problem and proposed a method to solve

it (a generalisation of the power method for standard eigenvalue problem which has been proved to converge by Chu and Watterson [14]). His method was designed to find a one-dimensional common representation. We will describe the original formulation, rewrite it in dual form and include a regularisation. After we present the algorithm (Horst) to solve the problem we will deal with how to derive a higher dimensional representation.

4.3 Primal formulation

Consider a set of vectors $w_i \in \mathbb{R}^{n_i}$, $i = 1...m$. For each random vector $\mathcal{X}_i$, with dimension n_i, we can define a univariate random variable $\mathcal{Z}_i$ as a linear combination of its features: $\mathcal{Z}_i = w_i' \mathcal{X}_i$. We can now compute pairwise correlation coefficients for each pair of the variables $\mathcal{Z}_i$. The goal is to find the vectors w_i so that the sum of all pairwise correlations is the highest. With the same re-scaling argument as in standard CCA the optimisation can be written as:

$$\max_{w_1,\ldots,w_m} \sum_{i<j} w_i' X^i X^{j'} w_j,$$

s.t.

$$w_i' X^i X^{i'} w_i = 1, \quad \forall i,$$

where $X^i \in \mathbb{R}^{n_i \times n}$ are centered matrices of observations of random vectors $\mathcal{X}_i$.

4.4 Dual formulation

We will reformulate the problem in dual form to make the problem feasible in the case of high dimensional data (e.g. text mining, where the number of features is the number of words encountered in the corpus) and with the use of the kernel trick make the solution more flexible than the linear model [18]. To express the problem in dual form we intruduce new variables (we will also refer to them as dual variables), $\beta_i \in \mathbb{R}^n$, so that $w_i = X^i \beta_i$. Let K_i be the kernel matrix computed on data X^i, which means that the element in the k-th row and l-th column of K_i is equal to $\langle \phi_i(X_k^i), \phi_i(X_l^i) \rangle$ for some mapping $\phi_i : \mathbb{R}^{n_i} \to \mathcal{H}_i$ to some Hilbert space $\mathcal{H}_i$. The kernelised dual formulation of the problem is then:

$$\max_{\beta_1,\ldots,\beta_m} \sum_{i<j} \beta_i' K_i K_j \beta_j,$$

s.t.

$$\beta_i' K_i K_i \beta_i = 1, \quad \forall i.$$

4.5 Regularisation

On the domain of text mining it is very common that the dimensionality of the vector space model is much higher than the number of documents. This can make the problem of finding correlations underdetermined and bound to overfit (extract spurious patterns). Regularisation is a way of restricting the flexibility and helps us avoid this problem to some extent.

4.5.1 Overfitting

We will show that if the kernels K_i are invertible, a direct implementation of SUMCOR leads to trivial, useless solutions (i.e. starting with an arbitrary projection vector in one space one can compute the other vectors so that all the pairs of projected spaces will be perfectly correlated).

Note that the SUMCOR optimisation problem is equivalent to the following problem:

$$\min_{\beta_1,\ldots,\beta_m} \sum_{i<j} \|K_i\beta_i - K_j\beta_j\|^2$$

s.t.

$$\beta_i' K_i^2 \beta_i = 1, \quad \forall i.$$

Indeed, if we expand the formula and use the unit variance conditions:

$$\min_{\beta_1,\ldots,\beta_m} \sum_{i<j} \beta_i' K_i^2 \beta_i - 2\beta_i K_i K_j \beta_j + \beta_j K_j^2 \beta_j =$$

$$= \max_{\beta_1,\ldots,\beta_m} \sum_{i<j} (-1 + 2\beta_i K_i K_j \beta_j - 1),$$

and this is equivalent to optimising the SUMCOR objective function. An upper bound for the SUMCOR objective function is $\binom{m}{2}$, which is the case when all the pairs of projected spaces are maximally correlated. In that case the alternative objective function is equal to zero. Let β_1 be an arbitrary vector which satisfies $\beta_1 K_1^2 \beta_1 = 1$. We set

$$\beta_i := K_i^{-1} K_1 \beta_1,$$

and show that they have unit variance and that they maximise the SUMCOR criterion function.

Unit variance:

$$Var(\beta_i) = \beta_i' K_i K_i \beta_i =$$

$$= \beta_1' K_1 K_i^{-1} K_i K_i K_i^{-1} K_1 \beta_1 = \beta_1' K_1 K_1 \beta_1 = 1.$$

Maximal correlation:

$$\sum_{i<j} (\beta_i' K_i K_j \beta_j) = \sum_{i<j} (\beta_1' K_1 K_i^{-1} K_i K_j K_j^{-1} K_1 \beta_1) =$$

$$= \sum_{i<j} \beta_1' K_1 K_1 \beta_1 = \sum_{i<j} 1 = \binom{m}{2}.$$

This means that there is a continuum $\mathbb{R}^n$ of optimal solutions. Not that most of them yields low correlation on a test set and probably does not correspond to a language concept.

4.5.2 Regularisation

One way of regularising is to use a parameter $\kappa \in [0, 1]$ and use regularised kernel matrices

$$\tilde{K}_i := (1 - \kappa) K_i + \kappa I$$

with the optimisation:

$$\max_{\beta_1,\ldots,\beta_m} \sum_{i<j} \beta_i' K_i K_j \beta_j, \tag{4.1}$$

s.t.

$$\beta_i' \tilde{K}_i \tilde{K}_i \beta_i = 1, \quad \forall i.$$

Every unit variance constraint $\beta' K^2 \beta = 1$ is replaced by $\beta' \big((1-\kappa)^2 K^2 + 2\kappa(1-\kappa)K + \kappa^2 I\big)\beta = 1$. This bounds the quantities $\beta' K^2 \beta$, $\beta' K \beta$ and $\beta' \beta$ which respectively correspond to the variance and the norm of primal variables and the norm of the dual variables. The main benefit of this form of regularisation is that dual regularised variance matrix is already decomposed into Cholesky factors which saves a great deal of computations. For further details refer to [1].

The regularisation in [1] paper is realised by taking $\hat{K}_i = K_i + \tau I$ for $\tau \in [0, \infty)$. This is equivalent to the $\tilde{K}_i = (1-\kappa)K_i + \kappa I$ regularisation for $\kappa \in [0, 1)$. We can prove that both regularisation schemes yield the same set of optimisation problems by looking at the regularised variance equation in (4.1) and divide both sides by $(1-\kappa)^2$. We get to the second form of regularisation by setting $\tau := \frac{\kappa}{1-\kappa}$.

Solutions of this problem do not force $\mathcal{Z}_i$ (introduced in section 4.3) to have unit variance and they must be rescaled after the optimisation. In addition to restricting the function space to prevent overfitting the regularisation also improves the scalability (the conditional number of the regularised variance matrix is lower, thus the convergence rate of conjugate gradient method for computing the inverse is faster) and numerical stability of the algorithm that solves the problem.

4.6 Multivariate eigenproblem

We will arrive at the generalised multivariate eigenproblem by using the Lagrangian techniques on the optimisation problem (4.1). Since $\beta_i \tilde{K}_i^2 \beta_i$ is a constant, we do not change the optimisation problem if we add $\frac{1}{2(1-\kappa)^2} \sum_i \beta_i \tilde{K}_i^2 \beta_i$ to the objective. The last technical detail will ensure the convergence of the algorithm that will solve the problem. The Lagrangian of this problem is equal to:

$$\sum_{i<j} \beta_i' K_i K_j \beta_j - \sum_i \lambda_i (\beta_i' \tilde{K}_i \tilde{K}_i \beta_i - 1) +$$

$$+ \frac{1}{2(1-\kappa)^2} \sum_i \beta_i' \tilde{K}_i^2 \beta_i.$$

We can substitute every β_i with $\tilde{K}_i^{-1} \alpha_i$ in the Lagrangian since $\tilde{K}_i$ is invertible:

$$\sum_{i<j} \alpha_i' \tilde{K}_i^{-1} K_i K_j \tilde{K}_j^{-1} \alpha_j - \sum_i \lambda_i (\alpha_i' \alpha_i - 1) + \frac{1}{2(1-\kappa)^2} \sum_i \alpha_i' \tilde{K}_i^{-1} \tilde{K}_i \tilde{K}_i \tilde{K}_i^{-1} \alpha_i. \tag{4.2}$$

The matrix $\tilde{K}_i$ is canceled out in the last summand:

$$\sum_{i<j} \alpha_i' \tilde{K}_i^{-1} K_i K_j \tilde{K}_j^{-1} \alpha_j - \sum_i \lambda_i (\alpha_i' \alpha_i - 1) + \frac{1}{2(1-\kappa)^2} \sum_i \alpha_i' I \alpha_i. \tag{4.3}$$

Taking the derivative of the Lagrangian in (4.3) with respect to α_i and setting equal to zero gives us n equations:

$$\sum_{j, j \neq i} \tilde{K}_i^{-1} K_i K_j \tilde{K}_j^{-1} \alpha_j - \lambda_i \alpha_i + \frac{1}{(1-\kappa)^2} \sum_i \alpha_i = 0, \quad \forall i, \tag{4.4}$$

The equations can be written in block matrix notation:

$$A\alpha = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,m} \\ A_{2,1} & \ddots & & \vdots \\ \vdots & & & A_{(m-1),m} \\ A_{m,1} & A_{m,2} & \cdots & A_{m,m} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \lambda_1\alpha_1 \\ \lambda_2\alpha_2 \\ \vdots \\ \lambda_m\alpha_m \end{bmatrix}, \tag{4.5}$$

where $A_{i,i} = \frac{1}{(1-\kappa)^2} I$ and $A_{i,j} = \tilde{K}_i^{-1} K_i K_j \tilde{K}_j^{-1}$ for $i \neq j$. Note that we need to project the solutions of the problem to the original space with $\tilde{K}_i^{-1}$, due to the substitution. A is obviously symmetric. We will now show that the matrix A in the above MEP is positive definite which then guarantees convergence of the algorithm to solve the problem. By using the definition of $\tilde{K}_i$ with equation (4.2) the Lagrangian can be written as:

$$\sum_{i<j} \alpha_i' \tilde{K}_i^{-1} K_i K_j \tilde{K}_j^{-1} \alpha_j - \sum_i \lambda_i(\alpha_i'\alpha_i - 1) + \tag{4.6}$$
$$+ \frac{1}{2(1-\kappa)^2} \sum_i \alpha_i' \tilde{K}_i^{-1} \big((1-\kappa)^2 K_i^2 + 2\kappa(1-\kappa)K_i + \kappa^2 I\big) \tilde{K}_i^{-1} \alpha_i.$$

Taking the derivative of the Lagrangian in (4.2) with respect to α_i and setting equal to zero gives us n equations:

$$\sum_j \tilde{K}_i^{-1} K_i K_j \tilde{K}_j^{-1} \alpha_j - \lambda_i \alpha_i +$$
$$+ \left(\frac{2\kappa}{1-\kappa} \tilde{K}_i^{-1} K_i \tilde{K}_i^{-1} + \frac{\kappa^2}{(1-\kappa)^2} \tilde{K}_i^{-2} \right) \alpha_i = 0, \quad \forall i, \tag{4.7}$$

after some rearranging of summands. The next step is to write the equation 4.7 in matrix form:

$$A\alpha = \left(\begin{bmatrix} B_{1,1} & \cdots & B_{1,m} \\ \vdots & & \vdots \\ B_{m,1} & \cdots & B_{m,m} \end{bmatrix} + \begin{bmatrix} C_{1,1} & \cdots & C_{1,m} \\ \vdots & & \vdots \\ C_{m,1} & \cdots & C_{m,m} \end{bmatrix} + \begin{bmatrix} D_{1,1} & \cdots & D_{1,m} \\ \vdots & & \vdots \\ D_{m,1} & \cdots & D_{m,m} \end{bmatrix} \right) \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} = \begin{bmatrix} \lambda_1\alpha_1 \\ \vdots \\ \lambda_m\alpha_m \end{bmatrix}, \tag{4.8}$$

where A was decomposed to three block matrices B, C and D whose blocks are defined by:

$$B_{i,j} = K_i^{-1} K_i K_j \tilde{K}_j^{-1},$$
$$C_{i,i} = \frac{2\kappa}{1-\kappa} \tilde{K}_i^{-1} K_i \tilde{K}_i^{-1}, \qquad C_{i,j} = 0; i \neq j,$$
$$D_{i,i} = \frac{\kappa^2}{(1-\kappa)^2} \tilde{K}_i^{-2}, \qquad D_{i,j} = 0; i \neq j.$$

Matrix D is strictly positive definite, because it is a block diagonal matrix and all its blocks are strictly positive (since $\tilde{K}_i$ is positive-definite, so is $\tilde{K}_i^2$ and so is its inverse $\tilde{K}_i^{-2}$ and multiplying with a positive scalar $\frac{\kappa^2}{(1-\kappa)^2}$ keeps it positive). Define matrices G_i as Cholesky factors of matrices K_i, e.g. $K_i = G_i G_i'$. Block matrices B and C are positive-semidefinite, since they have a Cholesky decompositions: $B = EE'$ and $C = FF'$ for matrices E and F defined as (E and F are block column matrices):

$$E = \begin{bmatrix} \tilde{K}_1^{-1} K_1 \\ \vdots \\ \tilde{K}_m^{-1} K_m \end{bmatrix}, \qquad F = \sqrt{\frac{2\kappa}{1-\kappa}} \begin{bmatrix} \tilde{K_1}^{-1} G_1 \\ \vdots \\ \tilde{K_m}^{-1} G_m \end{bmatrix}. \tag{4.9}$$

One can prove that the solutions $\alpha_1, \ldots, \alpha_m$ to the equation (4.5) that maximise the $\sum_i \lambda_i$ are the solutions of the optimisation problem (4.1) (after they are transformed to corresponding $\beta_1, \ldots, \beta_m$).

4.7 Horst algorithm

The algorithm for solving a multivariate eigenvalue problem (4.5) converges if A is symmetric and positive-definite. The general iterative procedure is given as Algorithm 1:

Algorithm 1 Horst algorithm

Input: matrices $A_{i,j}$, initial vectors α_i^0 ;$i, j = 1, \ldots, m$
Output: $\alpha_1^{maxiter}, \ldots, \alpha_m^{maxiter}$

```
for i = 1 to maxiter do
  for j = 1 to m do
    α_j^i ← Σ_k A_{j,k} α_k^{i-1}
    α_j^i ← α_j^i / sqrt(α_j^{i'} α_j^i)
  end for
end for
```

Every step of the main loop in the algorithm 1 involves m^2 matrix vector multiplications. We can exploit the structure of the problem, namely the fact that the MEP matrix is a sum of block diagonal matrices and a low rank block matrix (with block column matrix as its Cholesky factor). This gives us a speed up by a factor of m and leads us to Algorithm 4.7 (the sum in the innermost "j" loop involves adding $m-1$ vecotrs which is inexpensive and multiplying the result with only one matrix matrix, that is $\tilde{K}_j^{-1} K_j$).

Algorithm 2 Horst algorithm for computing a one-dimensional representation

Input: $K_1, \ldots, K_m$, κ, $maxiter$
Output: $\beta_1, \ldots, \beta_m$

```
K̃_i = (1 − κ)K_i + κI, ∀i
Choose random vectors α_1^0, ..., α_m^0
u_i^0 = K_i K̃_i^{-1} α_i^0, ∀i
for i = 1 to maxiter do
  for j = 1 to m do
    α_j^i ← K̃_j^{-1} K_j Σ_{k≠j} u_k^{i-1} + 1/(1−κ)^2 α_j^{i-1}
    α_j^i ← α_j^i / sqrt(α_j^{i'} α_j^i)
    u_j^i = K_j K̃_j^{-1} α_j^i
  end for
end for
for i = 1 to m do
  β_i = K̃_i^{-1} α_i^{maxiter}
end for
```

4.8 Computing more than one set of projections

Usually a one-dimensional representation (see algorithm 4.7) does not sufficiently capture all the information in the data and higher dimensional subspaces are needed. After computing the first set of primal canonical vectors $w_1^1, \ldots, w_m^1$ we proceed to computing the next set. We search for the next set of vectors $w_1^2, \ldots, w_m^2$ that is maximally correlated and essentially different from the first set, e.g. $w_i^{1\prime} \mathcal{X}_i$ and $w_i^{2\prime} \mathcal{X}_i$

are uncorrelated random variables for every i. We have mentioned that CCA for two views can be solved as a generalised eigenvalue problem (so $\lambda_1 = \lambda_2$). We have already introduced the dual view, kernels and regularisation. Using the Cholesky decomposition of the regularised dual variance and its inverse one transforms the problem to a symmetric standard eigenvalue problem. In addition, for every eigenvalue-eigenvector pair, $\left(\lambda, (\beta_1', \beta_2')'\right)$, the pair $\left(-\lambda, (\beta_1', -\beta_2')'\right)$ is also an eigen pair. Consequently one can retrieve all the eigenvectors $\left(\beta_1^{1\prime}, \beta_2^{1\prime}\right)', \ldots \left(\beta_1^{n\prime}, \beta_2^{n\prime}\right)'$ corresponding to positive eigenvalues so that the component vectors will be uncorrelated with respect to the regularised kernel: $\beta_1^i \tilde{K}_1^2 \beta_1^j = 0, i \neq j$ and the same for the second view. Ideally they should be uncorrelated with respect to the original kernels.

In CCA for more than two views the situation is different. For m views the size of the set of multivariate eigenvalue solutions is $(2n)^m$ [14], so there is a massive redundancy for every view. There is no obvious way to see that there exists a subset of the set of all multivariate eigenvector solutions that would be uncorrelated in each component. We will present an extension to Horst algorithm that keeps orthogonalising the component vectors.

We will force the projection vectors to be uncorrelated with respect to $\tilde{K}_i^2$ (same as in two view KCCA) because we can prove that the resulting optimisation problem is a multivariate eigenproblem and Horst algorithm can be applied. Small values of the regularisation parameter will result in approximately uncorrelated vectors. Let us assume that $W_i^k = (\beta_i^1, \ldots, \beta_i^k)$ is the matrix of k uncorrelated vectors with respect to $\tilde{K}_i$, for every view i. We are searching for the set of vectors $\beta_1^{k+1}, \ldots, \beta_m^{k+1}$ with unit regularised variance that maximise the SUMCOR objective and are uncorrelated with the first k solutions:

$$\beta_i^{k+1\prime} \tilde{K}_i^2 \beta_i^j, \forall j < k+1, \forall i,$$

which can be written as:

$$\beta_i^{k+1\prime} \tilde{K}_i^2 W_i = 0, \forall i.$$

In order to find vectors β_i^{k+1} that satisfy the condition above we define projection operators

$$P_i^{k+1} = I - \tilde{K}_i W_i^k W_i^{k\prime} \tilde{K}_i,$$

which map to the space orthogonal to the columns of $\tilde{K}_i W_i$. When proving that P_i is a projection operator (P must satisfy $P^2 = P$) it is crucial that the columns of W_i are uncorrelated:

$$P_i^2 = (I - \tilde{K}_i W_i^k W_i^{k\prime} \tilde{K}_i)(I - \tilde{K}_i W_i^k W_i^{k\prime} \tilde{K}_i) =$$

$$= I - 2\tilde{K}_i W_i^k W_i^{k\prime} \tilde{K}_i + \tilde{K}_i W_i^k W_i^{k\prime} \tilde{K}_i = P_i.$$

We will now extend the set of constraints in the SUMCOR optimisation (4.1) to enforce the orthogonality.

$$\max_{\beta_1,\ldots,\beta_m} \sum_{i<j} \beta_i' K_i K_j \beta_j + \frac{1}{2(1-\kappa)^2} \sum_i \beta_i' \tilde{K}_i^2 \beta_i, \tag{4.10}$$

s.t.

$$\beta_i' \tilde{K}_i \tilde{K}_i \beta_i = 1, \quad \forall i$$

$$W_i^{k\prime} \tilde{K}_i^2 \beta_i = 0, \forall i. \tag{4.11}$$

Substituting β_i with $\tilde{K}_i^{-1} \alpha_i$ yields the optimisation:

$$\max_{\alpha_1,\ldots,\alpha_m} \sum_{i<j} \alpha_i' \tilde{K}_i^{-1} K_i K_j \tilde{K}_j^{-1} \alpha_j + \frac{1}{2(1-\kappa)^2} \sum_i \alpha_i' \alpha_i, \tag{4.12}$$

s.t.

$$\alpha_i' \alpha_i = 1, \quad \forall i$$

$$W_i^{k'}\tilde{K}_i\alpha_i = 0, \quad \forall i. \tag{4.13}$$

Taking the derivative of the Lagrangian of the optimisation gives:

$$\sum_{j\neq i}\tilde{K}_i^{-1}K_iK_j\tilde{K}_j^{-1}\alpha_j + \frac{1}{(1-\kappa)^2}\alpha_i + \lambda_i\alpha_i + \tilde{K}_iW_i^k\mu_i = 0$$

for all i, where $\mu_i \in \mathbb{R}^k$ is the dual vector corresponding to constraint matrix $W_i^{k'}\tilde{K}_i$. We can eliminate each vector μ_i by multiplying each corresponding the equation by P_i. The problem is transformed to:

$$\sum_{j\neq i}P_i\tilde{K}_i^{-1}K_iK_j\tilde{K}_j^{-1}\alpha_j + \frac{1}{(1-\kappa)^2}P_i\alpha_i + \lambda_iP_i\alpha_i = 0, \forall i.$$

From the orthogonality constraints it follows that $P_i\alpha_i = \alpha_i$ which we use in every summand of the above sum. This enables us to solve the problem as a standard multivariate eigenproblem with a symmetric and positive-semidefinite matrix:

$$\sum_{j\neq i}P_i\tilde{K}_i^{-1}K_iK_j\tilde{K}_j^{-1}P_j\alpha_j + \frac{1}{(1-\kappa)^2}\alpha_i + \lambda_i\alpha_i = 0, \forall i.$$

Note that we will omit the inner products with P_j in the algorithm, since they only served the purpose of seeing that the multivariate eigenvalue problem (MEP) is symmetric.

Matrix of this MEP is positive-semidefinite because it can be written as PAP, where A is the MEP matrix for computing the first set of projection vectors and P is the symmetric block diagonal matrix with blocks P_i. We can apply shifting to ensure strict positive-definiteness. By adding a small positive number $\epsilon > 0$ to the diagonal of the MEP matrix with we do not change the optimal projection vectors, we just shift their corresponding λ multipliers. One can easily prove that if $\lambda_1, ..., \lambda_m$ and $\alpha_1, \ldots, \alpha_m$ are solutions to a MEP with matrix A, then $\lambda_1 + \epsilon, ..., \lambda_m + \epsilon$ and $\alpha_1, \ldots, \alpha_m$ are the solutions to MEP with matrix $A + \epsilon I$ (all we do is add $\epsilon\alpha$ on both sides of the MEP equation, where α a concatenation of all α_i vectors). The algorithm is shown in Algorithm 3.

REMARK: In step i of the maxiter loop the computation of α_j^i involves vectors $\alpha_1^{i-1}, \ldots, \alpha_m^{i-1}$, so we ignore the fact that we have already computed vectors $\alpha_1^i, \ldots, \alpha_{j-1}^i$. Using the latest updates can speed up convergence.

The choice of ϵ is arbitrary, it guarantees convergence as long as it is positive. Large ϵ values slow down convergence because they increase the conditional number of the MEP matrix. In our experience the algorithm converged with $\epsilon = 0$. The κ parameter is usually found by cross validation. In our experience setting κ close too to zero considerably decreases the performance (otherwise the κ coefficients produce models with comparable performance).

4.9 Implementation for linear kernel and sparse data

The algorithm involves matrix vector multiplications and inverted matrix vector multiplications. If kernel matrices are products of sparse matrices: $K_i = X_i'X_i$ with X_i having sn elements where $s << n$, then kernel matrix vector multiplications cost $2ns$ instead of n^2. We omit computing the full inverses and rather solve the system $K_ix = y$ for x, every time $K_i^{-1}y$ is needed. Since regularised kernels are symmetric and multiplying them with vectors is fast (roughly four times slower as multiplying with original sparse matrices X_i), an iterative method like conjugate gradient (CG) is suitable. Higher regularisation parameters increase the condition number of each $\tilde{K}_i$ which speeds up CG convergence.

Algorithm 3 Horst algorithm for computing a k-dimensional representation

Input: $K_1, \ldots, K_m$, κ, $maxiter$, k,
Output: $W_1^k, \ldots, W_m^k$

$\tilde{K}_i = (1-\kappa)K_i + \kappa I, \forall i$
for $d = 1$ to k **do**
  Choose random vectors $\alpha_1^0, \ldots, \alpha_m^0$
  if $d > 1$ **then**
    $P_i^d = I - \tilde{K}_i W_i^{d-1} W_i^{d-1\prime} \tilde{K}_i$
    Set $\alpha_i^0 \leftarrow P_i^d \alpha_i^0, \quad \forall i$
  else
    $P_i^d = I \quad \forall i$
  end if
  $u_i^0 = K_i \tilde{K}_i^{-1} \alpha_i^0, \forall i$
  for $i = 1$ to $maxiter$ **do**
    for $j = 1$ to m **do**
      $\alpha_j^i \leftarrow P_j^d \tilde{K}_j^{-1} K_j \sum_{k \neq j} u_k^{i-1} + \left(\frac{1}{(1-\kappa)^2} + \epsilon\right) \alpha_j^{i-1}$
      $\alpha_j^i \leftarrow \frac{\alpha_j^i}{\sqrt{\alpha_j^{i\prime} \alpha_j^i}}$
      $u_j^i \leftarrow K_k \tilde{K}_k^{-1} \alpha_k^i$
    end for
  end for
  for $l = 1$ to m **do**
    $\beta_l^d = \tilde{K}_l^{-1} \alpha_l^{maxiter}$
    $W_l^d = [W_l^d, beta_l^d]$ if $d > 1$
    $W_l^d = [beta_l^d]$ if $d = 1$
  end for
end for

If we empirically fix the number of iterations, $maxiter$, and number of CG steps, C, the computational cost of computing a k-dimensional representation is upper bounded by:

$$O\big(C \cdot maxiter \cdot k^2 \cdot m \cdot n \cdot s\big),$$

where m is the number of views, n the number of observations and s average number of nonzero features of each observation.

For example: first line of the innermost loop of Algorithm 3 involves computing $P_j^d \tilde{K}_j^{-1} K_j \sum_{k \neq j} u_k^{i-1}$, for given vectors u_k^{i-1}. Summing the vectors is a linear complexity operation and we can efficently compute $u = \sum_{k \neq j} u_k^{i-1}$. Multiplying u with K_j can be realised by first computing $u2 = X'u$ (fast) and $u3 = Xu2$ (fast). Computing $u4 = \tilde{K}_j^{-1} u3$ is realised by approximately solving the system $u3 = \tilde{K}_j u4$. Since $\tilde{K}_j$ is a symmetric positve definite matrix and multiplying it with vectors is fast we can use the conjugate gradient method to solve the system (majority of computations in each step of the CG method involve multiplying some vector with the matrix of the linear system). Finally, we compute $P_j^d u4$, which involves d inner products between full vectors. Note that the squared k factor is a consequence of multiplying with projection matrices, since they are not sparse and keep growing in every step of the algorithm.

An extra speed-up can be achieved by adapting the algorithm to run in parallel. This is simple, since the majority of computations is focused on sparse matrix-vector multiplications. Since every sparse matrix involved is fixed, we can partition it into l blocks of rows so that the number of nonzero elements in each block is the close to equal for all blocks.

4.10 Centering

So far we have assumed that the data is centered. If that is not the case we can implement centering in two possible ways. Let $\vec{1}$ be a column vector of length n with all entries equal to 1. If we are working with full kernel matrices, then we can center each one at the beginning:

$$K_{center} := K - \frac{1}{n}\vec{1}\vec{1}'K - \frac{1}{n}K\vec{1}\vec{1}' + \frac{1}{n^2}(\vec{1}'K\vec{1})\vec{1}\vec{1}'.$$

We can center vectors on the fly when implementing matrix vector multiplication when using a linear kernel and sparse data:

$$K_{center}v := Kv - \frac{1}{n}\vec{1}\vec{1}'Kv - \frac{1}{n}K\vec{1}\vec{1}'v + \frac{1}{n^2}(\vec{1}'K\vec{1})\vec{1}\vec{1}'v,$$

where we can compute $K\vec{1}$, $\vec{1}'K\vec{1}$ at the start of the algorithm. The cost of extra computation of $\vec{1}'v$ and $\vec{1}'(Kv)$ is insignificant when compared to the cost of Kv.

4.11 Projecting a new example

Let us assume that we have computed the matrices W_i which map into the common space of dimension k. Let t be a new example in view j. If data is centered, then we can compute the representation of t in the common space:

$$t \to W_j' X^{j\prime} t.$$

If data is not centered and we computed the projections with centering at each step, then t is mapped by

$$t \to W_j'\big(X^{j\prime}t - \frac{1}{n}X^{j\prime}X^j\vec{1_j} - \frac{1}{n}\vec{1_j}\vec{1_j}'X^{j\prime}t +$$

$$+\frac{1}{n^2}\vec{1_j}\vec{1_j}'X^{j\prime}X^j\vec{1_j}).$$

This can be generalised to nonlinear kernels by replacing $X^{j\prime}X^j$ by the kernel matrix K_j and replacing $X^{j\prime}t$ with a vector whose i-th element is equal to $ker_j(X^j_{(i)}, t)$,where $ker_j(\cdot,\cdot)$ is the kernel function for the j-th view and $X^j_{(i)}$ is the i-th column of matrix X^j.

4.12 Experiments on EuroParl

The following section includes information retrieval experiments on the European Parliament corpus. We computed the semantic space for documents from ten different languages and compared the retrieval performance with two alternative approaches, namely Cross-lingual LSI and k-means clustering.

4.12.1 Data set and preprocessing

Experiments were conducted on the EuroParl, Release v3, [9] data set and include Danish, German, English, Spanish, Italian, Dutch, Portuguese, Swedish, Finish and French language. We first removed all documents that had one translation or more missing. Documents (each document is a day of sessions of the parliament) were then arranged alphabetically and split into smaller documents, so that each speaker intervention represented a separate document. We removed trivial entries (missing translation) and after that removed all documents that were not present in all ten languages. Thus we ended up with 107,873 documents per language. We kept the first 100,000 for training and remaining 7873 for testing or for testing and validation. They roughly correspond to all talks between 2.25.1999 and 12.17.1999. We then extracted the bag of words model for each language, where we kept all unigrams, bigrams and trigrams that occurred more than thirty times. For example: "Mr", "President" and "Mr_President" all occurred more than thirty times in the English part of the corpus and they each represent a dimension in the bag of words space. This resulted in roughly 200,000-dimensional feature spaces for each language. Finally we computed the tf-idf weighting and normalised every document.

4.12.2 Mate retrieval, pseudo-query retrieval

Given a source language, e.g. English, and a target language, e.g. French, we can test the quality of the computed common space in the following way. We use the fact that the test set is aligned, so for every source document, or query, q we know which document in the target language is its translation (mate document) q'. We map the query and whole target corpus into the common space and sort all documents of the target language according to their similarity to the query. We then assign a score to the query that is some function of the rank of its mate document $rank(q')$. Common choices for the function include $rank(\cdot)$, $\frac{1}{rank(\cdot)}$, $\frac{n-rank(\cdot)}{n}$, where n is the number documents in the target test. We will be using $window10$ score, which assigns a value of 1 if rank is less or equal to 10 and value 0 otherwise.

Pseudo-query retrieval [4] is a variation of mate retrieval that is closer to realistic information retrieval scenarios. For a given query q we extract the top 5 or 10 words according to their tf-idf weights. We then use those words to form a new query r, so that the i-th element of the term frequency vector of $r(i) = 1$ if the i-th word was in the top 5 (or 10) and $r(i) = 0$ otherwise. We then recompute it's tf-idf score and normalise it. We then map it in the common space and proceed as in mate retrieval.

4.12.3 Comparing to CL-LSI and k-means clustering

We will compare our method with Cross-Lingual Latent Semantic Indexing and k-means clustering [7]. CL-LSI is an adaptation of LSI [3] for more than one view. The idea is to merge all document matrices

Table 4.1: k-means clustering

LANGUAGE	MATE	PQ10	PQ5
EN	0.7486	0.1611	0.1319
SP	0.7450	0.1553	0.1258
GE	0.5927	0.1658	0.1333
IT	0.7448	0.1596	0.1330
DU	0.7136	0.1622	0.1339
DA	0.5357	0.1685	0.1376
SW	0.5312	0.1717	0.1376
PT	0.7511	0.1524	0.1274
FR	0.7334	0.1562	0.1300
FI	0.4402	0.1690	0.1340

$X^1, \ldots, X^m$ into a single matrix Y by concatenating the aligned feature vectors:

$$y_i = (x_i^{1\prime}, \ldots, x_i^{m\prime})'.$$

We can now use the matrix Y with any unsupervised algorithm that returns a set of basis vectors, or concept vectors (as did MCCA). The final step when comparing to MCCA is to split the concept vectors into shorter concept vectors for each view in concordance with how the views were merged. LSI computes a singular value decomposition of Y and we can use an iterative method like Lanczos [2] algorithm to find the left singular vectors, corresponding to the highest singular values. In the k-means clustering setting we use the resulting cluster centroid vectors in the same way as singular vectors (of the full Y matrix) in CL-LSI are used for the semantic space.

We tested the preformance of the three methods on mate retrieval and pseudo-query retrieval for top5 and top10 tf-idf words with $window10$ on the 100-dimensional subspaces that the methods produced. For each source language we used all remaining nine languages, and averaged each score over all languages. For example: the upper-left entry in Table 4.1 is equal to 0.7486. This entry is the average $window10$ score of retrieving mates from English to French, English to German, and so on. Results imply that the concepts detected by MCCA in Table 4.3 are of higher quality than that of LSI in Table 4.2 and clustering in Table 4.1. One way to explain this superiority is that MCCA takes into account that data come from several sources that share some mutual information, whereas the clustering and LSI approaches discarded that information (after the views are concatenated we perform standard LSI which is "unaware" that features come from different views). LSI and CCA both find new features more informative features (can detect synonyms), whereas the clustering approach uses the original features and thus performs worse than the other two methods.

It is interesting that when Finnish language is the source language, the mate retrieval yields significantly lower results than other languages with LSI and clustering semantic spaces and that the results were comparable to other languages in MCCA. Finnish language is, linguistically speaking, glutinative and is usually preprocessed in way that takes this fact into account, but we used the same preprocessing for all languages. This indicates that MCCA can perform well with less preprocessing and can thus be applied more generally. This phenomenon was not observed in pseudo-query experiments.

4.13 Experiments on CLEF - Domain-Specific Track

In the second part of experiments we evaluated MCCA on the Domain-specific Track from CLEF. The track consists of German, English and Russian documents but in the experiments we focused only on

Table 4.2: CL-LSI

LANGUAGE	MATE	PQ10	PQ5
EN	0.9129	0.3147	0.2348
SP	0.9132	0.2907	0.2226
GE	0.8545	0.3296	0.2492
IT	0.9022	0.3101	0.2343
DU	0.9021	0.3193	0.2408
DA	0.8540	0.3322	0.2517
SW	0.8623	0.3303	0.2499
PT	0.9000	0.2840	0.2187
FR	0.9116	0.2881	0.2262
FI	0.7737	0.3230	0.2490

Table 4.3: MCCA

LANGUAGE	MATE	PQ10	PQ5
EN	0.9883	0.5577	0.4413
SP	0.9855	0.5289	0.4109
GE	0.9778	0.5318	0.4158
IT	0.9836	0.5552	0.4373
DU	0.9835	0.5545	0.4369
DA	0.9874	0.5415	0.4232
SW	0.9880	0.5157	0.4038
PT	0.9874	0.5253	0.4075
FR	0.9888	0.5000	0.3931
FI	0.9830	0.5304	0.4179

German and English parts. As part of the task we were given bilingual parallel resources for German and English, pseudo-aligned corpus GIRT for German and English of approximately 150.000 documents and the Cambridge Scientific Abstracts (CSA) corpus for English. We were also provided a set of 100 queries for each language, where 75 of them were used in tracks before 2007 and 25 were used in 2007. Note that the CSA corpus was used only in the 2007 track, this is why we give separate results for previous years and 2007.

For each of the query we were provided with a set of relevant documents. Experiment was conducted as follows. We selected a random subset of 30,000 training documents and computed a 2000 dimensional semantic space. We tested three subsets of computed canonical vectors (ordered by their corresponding correlation coefficients): the first one hundred, first one thousand and the full two thousand-dimensional semantic space. We took a query, retrieved top 1000 most relevant documents from opposite language and calculated MAP (mean average precision) for the set of retrieved documents. Table (4.4) demonstrates how the MAP score increases with the dimensionality of the semantic space. Observe that adding projections with corresponding lower correlation coefficients (computed on the training set) can sometimes decrease retrieval performance. Canonical vectors with lower correlation coefficients are likely to be spurious patterns and can decrease the focus of the search.

Please note that we only used the aligned corpus for training the MCCA and did not use the thesaurus which was also provided by the organizers of the Domain-specific Track. This reflects in the lower MAP than the one achieved by the best system in CLEF 2007, which was 45.68 on German corpus and 33.41 on English corpus.

Table 4.4: CLEF

QUERY SET	CORPUS	MAP:100	MAP:1000	MAP:2000
GERMAN 2006	ENGLISH GIRT	19.55	20.53	20.92 %
ENGLISH 2006	GERMAN GIRT	20.41	21.23	21.50 %
GERMAN 2007	ENGLISH GIRT AND CSA	17.08	17.84	17.44 %
ENGLISH 2007	GERMAN GIRT	24.00	24.72	24.82 %

There were only two languages used in this experiment, so the multi-way part of it did not play any role in this experiment. However, the scale was quite crucial, since the training set consisted of 30.000 documents.

4.14 Concept vectors

The multivariate random variables from MCCA in our experiments correspond to document-vectors (in the bag of words representation) in different languages. We will now consider sets of words that are correlated between the two or more languages (sets of words that have a correlated pattern of appearance across the aligned corpus). In the algorithmic part we refered to the concept vectors as w_i (primal view). We will assume that such sets approximate the notion of 'concepts' in each language, and that such concepts are the translation of each other. To illustrate the conceptual representation we have printed few of the most probable (most typical) words in each language for the first few components found from the EuroParl (Table 4.5) corpus and from the CLEF corpus (Table 4.6). The words are sorted by their weights in the concept vectors.

Table 4.5: Concept vectors for the EUROPARL experiment

DA	menneskerettighederne, menneskerettigheder, forretningsordenen, rusland, ndringsforslag, ã ndringsforslag
DE	menschenrechte, russland, posselt, menschenrechtsverletzungen, zusammenarbeit, nderungsantrag, verfahrensantrag
EN	amendment, amendments, russia, human rights, cooperation, resolution, of order
ES	enmienda, enmiendas, rusia, n de orden, de orden, reglamento, posselt
FI	ihmisoikeuksien, ihmisoikeuksia, tyã jã, tarkistuksen, tarkistusta, tarkistus, tarkistuksia
FR	amendement, amendements, posselt, russie, rã solution, russe, l amendement
IT	emendamenti, emendamento, risoluzione, russia, regolamento, cooperazione, bielorussia
NL	amendement, mensenrechten, amendementen, rusland, van orde, resolutie, samenwerking
PT	ponto de ordem, de ordem, alteraã, alteraã ã, direitos humanos, directiva, regimento
SV	resolutionen, ryssland, ordningsfrã, posselt, arbetsordningen, samarbete, ryska
DA	omdelt, dagsordenen, tak, er omdelt, protokollen fra, protokollen, strukturfondene
DE	tagesordnung, der tagesordnung, das protokoll der, kommissar, wurde verteilt, wurde verteilt gibt, haushalt
EN	commissioner, president commissioner, agenda, budget, commissioner the debate, commissioner the, item is
ES	comisario, distribuido, gracias, acta de la, comisaria, presupuesto, comisario el debate
FI	esityslistalla, kiitoksia, kiitos, esityslistalle, esityslistalla on, lissabonin, esityslistan
FR	merci, commissaire, jour appelle, du jour appelle, tã distribuã, ã tã distribuã, jeudi
IT	commissario, grazie, ringrazio, sono osservazioni, commissario la discussione, vi sono osservazioni, giovedã
NL	rondgedeeld, zijn rondgedeeld, de orde is, orde is, commissaris, orde is het, begroting
PT	obrigado, presidente senhor, conselho, acta, hã alguma, hã alguma observaã, comissã rio estã
SV	tack, r jag fã, rã det, kommissionsledamot, budget, har delats ut, kommissionã
DA	belarus, rusland, eu, tak, tak hr, formandskonferencen, hviderusland
DE	belarus, russland, eu, sitzung wird um, danke herr, zusammenarbeit, russischen
EN	belarus, russia, eu, the sitting was, sitting was, thank you, cooperation
ES	rusia, sesiã n a, belarã, belarã s, gracias, alemania, â seã or
FI	kiitos, kiitoksia, istunto, eu, euroopan, klo istunto, istunto pã
FR	belarus, russie, merci, levã e ã, merci monsieur, est levã e, ance est levã
IT	bielorussia, russia, la ringrazio, grazie, cooperazione, ue, la seduta
NL	rusland, uur gesloten, de vergadering wordt, samenwerking, eu, dank u, vergadering wordt
PT	ssia, ue, obrigado, suspensa ã s, suspensa ã, aprova a acta, sessã o ã
SV	vitryssland, ryssland, tack, eu, tack herr, ryska, talmanskonferensen
DA	artikel, algeriet, affald, myndigheder, dsstraffen, dã dsstraffen, fn
DE	gemeinsame aussprache ist, todesstrafe, algerien, die gemeinsame aussprache, unterbrochene sitzungsperiode
EN	joint debate is, joint debate, the joint debate, romania, amendment, algeria, article
ES	declaro, argelia, enmienda, debate conjunto, sesiã n del, rumania, el debate conjunto
FI	kuolemanrangaistuksen, yhteiskeskustelu, yhteiskeskustelu on pã, yhteiskeskustelu on, algerian, avatuksi
FR	amendement, roumanie, article, session du parlement, rapport, discussion commune est, la session du
IT	emendamento, romania, articolo, algeria, discussione congiunta, relazione, israele
NL	gecombineerd debat is, amendement, roemeniã, algerije, afval, zijn rondgedeeld, artikel
PT	declaro, sessã o do, encerrada a discussã, discussã o conjunta, israel, aprova a resoluã, artigo
SV	artikel, den gemensamma debatten, gemensamma debatten, gemensamma debatten ã, algeriet, avfall, dsstraffet

Table 4.6: Concept vectors for the GIRT experiment

EN	discourse, mine, software, unemployment, minority, automation, family policy, migration, social policy, cognitive
DE	diskurs, einzel handel, sozial politik, minderheit, software, automatisierung, arbeitslosigkeit, familien politik
EN	apprentice, youth, netherlands, success, reception, rhinelandpalatinate, structural change, credit, decision
DE	jugend, niederlande, lehrling, wissenschaft bundes, rezeption, struktur wandel, erfolg, kunst, verlag,
EN	netherlands, film, medium, productivity, counsel, medium policy, late migrant, gdr, social policy, television
DE	niederlande, aussiedler, sozial politik, ddr, medien politik, ubersiedler, wohlfahrt, beratung, film, produktivitat
EN	rhinelandpalatinate, theology, mathematics, election, customer, industrial, brazil, suicide, schleswigholstein
DE	industrie, theologie, stahl, mathematik, kunden, beruf bildung, brasilien, soziologie, schleswigholstein, golf
EN	youth, bremen, prognosis, deregulation, communication, democratization, radio, economic theory, business cycle
DE	jugend, bremen, konjunktur, arbeit recht, kommunikation, prognose, deregulierung, recht, arbeit markt, politisch
EN	fdj, air, technical change, intervention, anthropology, commuter, project, housework, industrial sociology
DE	luft, anthropologie, fdj, kultur anthropologie, industrie soziologie, offentlich dienst, intervention
EN	hungary, inflation, korea, incentive, educational, aesthetics, simmel, education, television, cognitive
DE	bildung, ungarn, psychotherapie, korea, aussiedler, anreiz, inflation, asthetik, kognitiv, beruf bildung
EN	rhinelandpalatinate, party, policy, gdr, labor market, simulation, environmental, house policy, israel
DE	ddr, politik, arbeit markt, sozial wissenschaft, partei, spd, simulation, sed, forschung sozial, wohnung
EN	party, food, brazil, copyright, service, marxism, adorno, dentist, party system, australia
DE	partei, beruf bildung, brasilien, land bundes, parteien, adorno, gerontologie, australien, alt mensch, alt
EN	democratization, leisure, family, disarmament, wage, quality circle, postmodernism, innovation, turkey, circle
DE	demokratisierung, frei zeit, abrustung, qualitat zirkel, familie, zirkel, lohn, post modern, innovation

Chapter 5

Sparse kernel canonical correlation analysis

In this chapter we present two sparse variants of canonical correlation analysis (CCA). The first variant looks to create dual sparsity and is called the sparse kernel canonical correlation analysis (SKCCA). The second variant, attempts at both primal and dual sparsity and is simply referred to as the primal-dual sparse canonical correlation analysis (primal-dual SCCA). We start the presentation with motivations and general definitions.

5.1 Motivations and General Presentation

We analyse a framework for sparsity called matching pursuit that iteratively pursues parsimonious functions in order to approximate the entire problem. The algorithm originally appeared in the signal processing community [13] but has recently received much interest in the machine learning community with several authors proposing new variants/applications. Most notably [20] have proposed a sparse kernel principal components analysis (SKPCA) and [22] proposed kernel matching pursuit (KMP). The latter contribution is simply a kernel version of the matching pursuit algorithm that allows for non-linear separation of the data via the use of a sparse number of kernel basis vectors. This corresponds to constructing a sparse kernel least squares regression algorithm. The first algorithm was not marketed as a sparse KPCA but instead as an attempt at creating low rank approximations of the kernel matrix. However, it turns out that this algorithm is exactly a sparse version of kernel principal components analysis (KPCA). Armed with these two matching pursuit variants we go on to propose an equivalent formulation, in the matching pursuit setting, for kernel canonical correlation analysis (KCCA). We apply a similar trick from SKPCA and KMP, that allows the addition of a basis vector with maximum importance to the current set of basis vectors. The idea is to maximise the *Rayleigh quotient* for the KCCA problem without projecting the data into the common space defined by the set of eigenvectors. However, by carrying this out we are susceptible to choosing the same basis vectors in subsequent iterations. To avoid this situation we carry out an orthogonality step (deflation), equivalent to KMP, that guarantees spanning into a space orthogonal to all previously chosen basis vectors.

For KCCA we denote the paired samples by

$$S = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_m, \mathbf{y}_m)\} \subseteq (\mathcal{X} \times \mathcal{Y})^m,$$

where $\mathcal{X}$ and $\mathcal{Y}$ are the two input spaces. We will always assume that the examples are already projected into the kernel defined feature space, so that the kernel matrix $\mathbf{K}$ has entries $\mathbf{K}_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. For the paired sample we assume the projection into appropriate feature spaces with corresponding kernel matrices $\mathbf{K}_x$

and $\mathbf{K}_y$ respectively. Also, for a matrix $\mathbf{K}$, bracketed notation $\mathbf{K}[i,j]$ denotes the element in the ith row and jth column. Matrices $\mathbf{K}[i,:]$ and $\mathbf{K}[:,j]$ correspond to row i and column j, respectively, of the kernel $\mathbf{K}$. When using a set of indices $\mathbf{i}$ (say) then $\mathbf{K}[\mathbf{i},\mathbf{i}]$ denotes the square matrix defined solely by the index set $\mathbf{i}$. The transpose of a matrix $\mathbf{K}$ is denoted by $\mathbf{K}'$. Finally, a vector will be denoted by a lowercase bold font letter $\mathbf{x}$ (say) and its transpose denoted by $\mathbf{x}'$.

Assume we are given two views $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^m$ and $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^m$ of the same data where $\mathbf{x}_i$ and $\mathbf{y}_i$ are row vectors of length n and that both matrices have been centered. We then compute projections $P_x : \mathbf{x} \mapsto \mathbf{x}'\mathbf{w}_x$ and $P_y : \mathbf{y} \mapsto \mathbf{y}'\mathbf{w}_y$. The idea of canonical correlation analysis (CCA) is to maximise the correlation $corr(P_x(\mathbf{X}), P_y(\mathbf{Y}))$ between the data in their corresponding projection space. Taking the maximum correlation of these two projections reduces to the following maximisation problem:

$$\begin{aligned} \rho &= \max_{\mathbf{w}_x,\mathbf{w}_y} \frac{\mathbf{w}_x'\mathbf{X}\mathbf{Y}'\mathbf{w}_y}{\sqrt{\mathbf{w}_x'\mathbf{X}\mathbf{X}'\mathbf{w}_x\mathbf{w}_y'\mathbf{Y}\mathbf{Y}'\mathbf{w}_y}} \\ &= \max_{\mathbf{w}_x,\mathbf{w}_y} \frac{\mathbf{w}_x'\mathbf{C}_{xy}\mathbf{w}_y}{\sqrt{\mathbf{w}_x'\mathbf{C}_{xx}\mathbf{w}_x\mathbf{w}_y'\mathbf{C}_{yy}\mathbf{w}_y}}, \end{aligned} \tag{5.1}$$

where $\mathbf{C}_{xy} = \mathbf{C}_{yx}'$ is the covariance matrix between $\mathbf{X}$ and $\mathbf{Y}$ and $\mathbf{C}_{xx}$ and $\mathbf{C}_{yy}$ the covariance matrices of $\mathbf{X}$ and $\mathbf{Y}$, respectively.

The above maximisation problem can be evaluated by solving a *generalised eigenproblem* of the form:

$$\mathbf{A}\mathbf{w} = \lambda\mathbf{B}\mathbf{w}, \tag{5.2}$$

where $(\mathbf{w}, \lambda)$ are the eigenvector-eigenvalue pair corresponding to the solution and $\mathbf{A}$, $\mathbf{B}$ are square matrices. The CCA generalised eigenproblem (see [19] for a full derivation) can be written as :

$$\begin{bmatrix} \mathbf{0} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{C}_{xx} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{bmatrix}. \tag{5.3}$$

The fact that this problem only looks for linear relationships in data and does not tackle non-linear data sets limits its use in many contexts. By applying the kernel trick several authors [10, 1] have proposed a kernel version of canonical correlation analysis in order to tackle non-linear relations.

Let us map each training example to a higher dimensional space using a feature mapping $\phi : \mathbf{x} \mapsto \phi(\mathbf{x})$. In the case of a linear kernel each n-dimensional vector $\mathbf{x}$ is mapped with the identity $\phi(\mathbf{x}) = \mathbf{x}$ that corresponds to the following kernel matrix $\mathbf{K}_x = \mathbf{X}'\mathbf{X}$. Therefore by replacing $\mathbf{X}$ and $\mathbf{Y}$ in Equation (5.1) by (linear) kernels $\mathbf{K}_x = \mathbf{X}'\mathbf{X}$ and $\mathbf{K}_y = \mathbf{Y}'\mathbf{Y}$ and expressing $\mathbf{w}_x = \mathbf{X}\boldsymbol{\alpha}_x$ and $\mathbf{w}_y = \mathbf{Y}\boldsymbol{\alpha}_y$, we get the following kernel CCA generalised eigenproblem

$$\begin{bmatrix} \mathbf{0} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_x \\ \boldsymbol{\alpha}_y \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{K}_x^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_y^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_x \\ \boldsymbol{\alpha}_y \end{bmatrix}, \tag{5.4}$$

where $\mathbf{K}_{xy} = \mathbf{K}_x\mathbf{K}_y$, $\mathbf{K}_{yx} = \mathbf{K}_y\mathbf{K}_x = \mathbf{K}_{xy}'$, $\mathbf{K}_x^2 = \mathbf{K}_x\mathbf{K}_x$ and $\mathbf{K}_y^2 = \mathbf{K}_y\mathbf{K}_y$. Note that any kernel can be used in the above setting.

The solution of the above eigenproblem may lead to overfitting (see [6, 19]) and to avoid this the following regularised version has been proposed,

$$\begin{bmatrix} \mathbf{0} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_x \\ \boldsymbol{\alpha}_y \end{bmatrix} = \lambda \begin{bmatrix} (1-\tau_x)\mathbf{K}_x^2 + \tau_x\mathbf{K}_x & \mathbf{0} \\ \mathbf{0} & (1-\tau_y)\mathbf{K}_y^2 + \tau_y\mathbf{K}_y \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_x \\ \boldsymbol{\alpha}_y \end{bmatrix}, \tag{5.5}$$

where $0 < \tau_x, \tau_y < 1$ are regularisation parameters that each penalise the norms of the weight vectors $\mathbf{w}_x$ and $\mathbf{w}_y$, respectively. The solution of the generalised eigenproblem is of order $\mathcal{O}(m^3)$ complexity. Another downside is that twice the size of the kernel matrices must be stored in memory in order to solve the generalised eigenvalue problem as each copy of the matrix must be stored in each of the larger matrices given in in Equation(5.4). These problems have been tackled by reducing to solving a standard eigenproblem which saves on memory but the overall saving in time complexity is not significant. Also, Gram Schmidt orthogonalisation procedures have been used to tackle these problems. However, analysing data sets with more than $10,000$ data points is typically a real challenge with any of these variants of KCCA. We now present methods that deliver dual sparsity which result in fast training/testing times and tractability for larger data sets.

5.2 Basis selection

We would like to construct (dual) sparse kernel canonical correlation analysis (SKCCA) algorithms. By sparsity we mean using a small subset of basis vectors from the sample $(\mathbf{x}_i, \mathbf{y}_i), i = 1, \ldots, m$. The sparse index set of basis vectors will be contained in an index vector $\mathbf{i}$. The sparse CCA problem can be expressed as:

$$\rho = \max_{\tilde{\mathbf{w}}_x, \tilde{\mathbf{w}}_y} \frac{\tilde{\mathbf{w}}_x' \mathbf{X}\mathbf{Y}' \tilde{\mathbf{w}}_y}{\sqrt{\tilde{\mathbf{w}}_x' \mathbf{X}\mathbf{X}' \tilde{\mathbf{w}}_x \tilde{\mathbf{w}}_y' \mathbf{Y}\mathbf{Y}' \tilde{\mathbf{w}}_y}},$$

where $\tilde{\mathbf{w}}_x \in span\{\mathbf{X}[:, \mathbf{i}]\}$ and $\tilde{\mathbf{w}}_y \in span\{\mathbf{Y}[:, \mathbf{i}]\}$. This equation can be converted into its dual by taking advantage of the fact that the primal weight vectors can be written in terms of a linear combination of the training examples and the dual weight vectors:

$$\tilde{\mathbf{w}}_x = \mathbf{X}[:, \mathbf{i}]\tilde{\boldsymbol{\alpha}}_x, \tag{5.6}$$
$$\tilde{\mathbf{w}}_y = \mathbf{Y}[:, \mathbf{i}]\tilde{\boldsymbol{\alpha}}_y. \tag{5.7}$$

We can substitute these two expressions into the CCA problem:

$$\rho = \max_{\mathbf{i}, \tilde{\boldsymbol{\alpha}}_x, \tilde{\boldsymbol{\alpha}}_y} \frac{\tilde{\boldsymbol{\alpha}}_x' \mathbf{X}[:, \mathbf{i}]' \mathbf{X}\mathbf{Y}' \mathbf{Y}[:, \mathbf{i}] \tilde{\boldsymbol{\alpha}}_y}{\sqrt{\tilde{\boldsymbol{\alpha}}_x' \mathbf{X}[:, \mathbf{i}]' \mathbf{X}\mathbf{X}' \mathbf{X}[:, \mathbf{i}] \tilde{\boldsymbol{\alpha}}_x \tilde{\boldsymbol{\alpha}}_y' \mathbf{Y}[:, \mathbf{i}]' \mathbf{Y}\mathbf{Y}' \mathbf{Y}[:, \mathbf{i}] \tilde{\boldsymbol{\alpha}}_y}}.$$

Furthermore, we have $\mathbf{K}_x[:, \mathbf{i}] = \mathbf{X}'\mathbf{X}[:, \mathbf{i}]$ and $\mathbf{K}_y[:, \mathbf{i}] = \mathbf{Y}'\mathbf{Y}[:, \mathbf{i}]$, therefore we have the sparse kernel CCA problem:

$$\rho = \max_{\mathbf{i}, \tilde{\boldsymbol{\alpha}}_x, \tilde{\boldsymbol{\alpha}}_y} \frac{\tilde{\boldsymbol{\alpha}}_x' \mathbf{K}_x[:, \mathbf{i}]' \mathbf{K}_y[:, \mathbf{i}] \tilde{\boldsymbol{\alpha}}_y}{\sqrt{\tilde{\boldsymbol{\alpha}}_x' \mathbf{K}_x^2[\mathbf{i}, \mathbf{i}] \tilde{\boldsymbol{\alpha}}_x \tilde{\boldsymbol{\alpha}}_y' \mathbf{K}_y^2[\mathbf{i}, \mathbf{i}] \tilde{\boldsymbol{\alpha}}_y}},$$

where $\tilde{\boldsymbol{\alpha}}_x$ and $\tilde{\boldsymbol{\alpha}}_y$ are sparse dual eigenvectors. This leads to, for fixed $\mathbf{i}$, the sparse KCCA generalised eigenproblem of the form

$$\begin{bmatrix} \mathbf{0} & \mathbf{K}_{xy}[\mathbf{i}, \mathbf{i}] \\ \mathbf{K}_{yx}[\mathbf{i}, \mathbf{i}] & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\alpha}}_x \\ \tilde{\boldsymbol{\alpha}}_y \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{K}_x^2[\mathbf{i}, \mathbf{i}] & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_y^2[\mathbf{i}, \mathbf{i}] \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\alpha}}_x \\ \tilde{\boldsymbol{\alpha}}_y \end{bmatrix}. \tag{5.8}$$

Therefore, the solution of Equation (5.8) leads to a sparse set of dual weight vectors. The sparse kernel canonical correlation analysis algorithm is given in Function `sparseKCCA`. This is the sparse KCCA algorithm that we will use throughout this section whenever we are given two kernels $\mathbf{K}_x$, $\mathbf{K}_y$ and a (small) index set $\mathbf{i}$.

1: set $\tilde{\mathbf{X}} = \mathbf{K}_x[:, \mathbf{i}]$ and $\tilde{\mathbf{Y}} = \mathbf{K}_y[:, \mathbf{i}]$
2: create matrices $\mathbf{K}_x^2[\mathbf{i}, \mathbf{i}] = \tilde{\mathbf{X}}'\tilde{\mathbf{X}}$, $\mathbf{K}_y^2[\mathbf{i}, \mathbf{i}] = \tilde{\mathbf{Y}}'\tilde{\mathbf{Y}}$, $\mathbf{K}_{xy}[\mathbf{i}, \mathbf{i}] = \tilde{\mathbf{X}}'\tilde{\mathbf{Y}}$ and $\mathbf{K}_{yx}[\mathbf{i}, \mathbf{i}] = \tilde{\mathbf{K}}'_{xy}$
3: solve the following generalised eigenvalue problem:

$$\begin{bmatrix} \mathbf{0} & \mathbf{K}_{xy}[\mathbf{i}, \mathbf{i}] \\ \mathbf{K}_{yx}[\mathbf{i}, \mathbf{i}] & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\alpha}}_x \\ \tilde{\boldsymbol{\alpha}}_y \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{K}_{xx}[\mathbf{i}, \mathbf{i}] & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{yy}[\mathbf{i}, \mathbf{i}] \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\alpha}}_x \\ \tilde{\boldsymbol{\alpha}}_y \end{bmatrix}$$

Output: sparse eigenvectors $\tilde{\boldsymbol{\alpha}}_x$, $\tilde{\boldsymbol{\alpha}}_y$ and eigenvalues λ.

Function `sparseKCCA` $(\mathbf{K}_x, \mathbf{K}_y, \mathbf{i})$

Now that we have described the `sparseKCCA` algorithm we turn our attention to finding the best sparse index set $\mathbf{i}$ (basis selection) of training examples to use in conjunction with the sparse KCCA.

Using a similar strategy to [20] we now show that maximising the quotient of the CCA problem leads to a fast method for choosing basis vectors. However, after picking a basis vector we must project into a space orthogonal to it, and describe a deflation step that guarantees future basis vectors chosen are orthogonal to all others. We would like to construct a very fast calculation that enables us to quickly find basis vectors. The work of Smola and Schölkopf [20] maximised the Rayleigh quotient for kernel principal components analysis and hence we would like to maximise the generalised Rayleigh quotient that we have for CCA,

$$\max_i \rho_i = \frac{\mathbf{e}_i' \mathbf{K}_x \mathbf{K}_y \mathbf{e}_i}{\sqrt{\mathbf{e}_i' \mathbf{K}_x^2 \mathbf{e}_i \mathbf{e}_i' \mathbf{K}_y^2 \mathbf{e}_i}}, \tag{5.9}$$

where $\mathbf{e}_i$ is the ith unit vector. At each iteration we look to find the basis vector that maximises the quotient given by Equation (5.9). Note that maximising this quotient is equivalent to finding the most maximally correlated data points in feature space and not the projected space as would be the case if we look for eigenvectors that maximise the quotient (5.1) We can rewrite this equation in terms of the kernel basis vectors by observing that $\mathbf{K}_x[i, :] = \mathbf{e}_i' \mathbf{K}_x$ and $\mathbf{K}_y[:, i] = \mathbf{K}_y \mathbf{e}_i$, which gives

$$\max_i \rho_i = \frac{\mathbf{K}_x[i, :]\mathbf{K}_y[:, i]}{\sqrt{(\mathbf{K}_x[i, :]\mathbf{K}_x[:, i])\,(\mathbf{K}_y[i, :]\mathbf{K}_y[:, i])}} = \frac{\mathbf{K}_x[i, :]\mathbf{K}_y[:, i]}{\sqrt{\mathbf{K}_x^2[i, i]\mathbf{K}_y^2[i, i]}}. \tag{5.10}$$

Once the index i has been chosen such that it maximises the above equation then the following orthogonality procedure (deflation) is carried out to make sure future chosen bases are sufficiently far (geometrically) from those already added to the set $\mathbf{i}$.

Initially, at the first step $j = 1$, let $\mathbf{K}_x^j = \mathbf{K}_x$ and $\mathbf{K}_y^j = \mathbf{K}_y$ denote the deflated kernel matrices at the jth iteration. To find the deflated matrices at step $j + 1$ we use the following single sided deflation taken from the kernel matching pursuit (KMP) literature (see [22]),

$$\mathbf{K}_x^{j+1} = \left(\mathbf{I} - \frac{\boldsymbol{\tau}_x \boldsymbol{\tau}_x'}{\boldsymbol{\tau}_x' \boldsymbol{\tau}_x}\right) \mathbf{K}_x^j = \mathbf{K}_x^j - \frac{\boldsymbol{\tau}_x \left(\boldsymbol{\tau}_x' \mathbf{K}_x^j\right)}{\boldsymbol{\tau}_x' \boldsymbol{\tau}_x}, \tag{5.11}$$

$$\mathbf{K}_y^{j+1} = \left(\mathbf{I} - \frac{\boldsymbol{\tau}_y \boldsymbol{\tau}_y'}{\boldsymbol{\tau}_y' \boldsymbol{\tau}_y}\right) \mathbf{K}_y^j = \mathbf{K}_y^j - \frac{\boldsymbol{\tau}_y \left(\boldsymbol{\tau}_y' \mathbf{K}_y^j\right)}{\boldsymbol{\tau}_y' \boldsymbol{\tau}_y}, \tag{5.12}$$

where $\boldsymbol{\tau}_x = \mathbf{K}_x^j[:, \mathbf{i}_j]$ and $\boldsymbol{\tau}_y = \mathbf{K}_y^j[:, \mathbf{i}_j]$ such that $j = |\mathbf{i}|$, $\mathbf{i}_j$ is the latest element added to vector $\mathbf{i}$ and $\mathbf{I}$ is the identity matrix. The deflation at each stage can be computed in $\mathcal{O}(m^2)$ time where m are the number of examples. This is because the bracketed computation in $\boldsymbol{\tau}_x \left(\boldsymbol{\tau}_x' \mathbf{K}_x\right)$ can be computed first, and then subsequent operations. We repeat the quotient maximisation and deflation procedure until d basis vectors have been chosen. This protocol is described in Algorithm 4.

Algorithm 4 A fast greedy algorithm for choosing basis vectors

Input: Two views $\mathbf{K}_x$, $\mathbf{K}_y$, sparsity parameter $k > 0$,

1: initialise an index vector $\mathbf{i} = [\,]$.
2: **for** $i = 1$ to d **do**
3: $\quad \mathbf{i}_i = \arg\max_{j=1,\ldots,m} \frac{\mathbf{K}_x[j,:]\mathbf{K}_y[:,j]}{\sqrt{\mathbf{K}_x^2[j,j]\mathbf{K}_y^2[j,j]}}$
4: $\quad$ set $\boldsymbol{\tau}_x = \mathbf{K}_x[:, \mathbf{i}_i]$ and $\boldsymbol{\tau}_y = \mathbf{K}_y[:, \mathbf{i}_i]$
5: $\quad$ deflate kernel matrices like so:

$$\mathbf{K}_x = \left(\mathbf{I} - \frac{\boldsymbol{\tau}_x \boldsymbol{\tau}_x'}{\boldsymbol{\tau}_x' \boldsymbol{\tau}_x}\right) \mathbf{K}_x$$

$$\mathbf{K}_y = \left(\mathbf{I} - \frac{\boldsymbol{\tau}_y \boldsymbol{\tau}_y'}{\boldsymbol{\tau}_y' \boldsymbol{\tau}_y}\right) \mathbf{K}_y$$

6: **end for**
7: run `sparseKCCA(`$\mathbf{K}_x$`,` $\mathbf{K}_y$`,` $\mathbf{i}$`)` with final $\mathbf{i}$ to find $\tilde{\boldsymbol{\alpha}}_x$, $\tilde{\boldsymbol{\alpha}}_y$ and λ

Output: index vector $\mathbf{i}$, and eigenvectors $\tilde{\boldsymbol{\alpha}}_x$, $\tilde{\boldsymbol{\alpha}}_y$

Another alternative for basis selection is to compute the Rayleigh quotient of Equation (5.10) once and then choose the d indices generating the largest correlation values. Clearly the algorithm has a constant complexity of $\mathcal{O}(m^2)$ and does not have the added complexity of deflation at each step. This algorithm is described in Algorithm 5. We show in the SKCCA experimental section that this algorithm competes in accuracy with the fast algorithm of Algorithm 4 for data sets of varying size.

Algorithm 5 A faster greedy algorithm for choosing basis vectors

Input: Two views $\mathbf{K}_x$, $\mathbf{K}_y$, sparsity parameter $k > 0$,

1: initialise an index vector $\mathbf{i} = [\,]$.
2: set $\mathbf{i}$ to the index of top d values of $\frac{\mathbf{K}_x[j,:]\mathbf{K}_y[:,j]}{\sqrt{\mathbf{K}_x^2[j,j]\mathbf{K}_y^2[j,j]}} \quad \forall j = 1, \ldots, m$
3: run `sparseKCCA(`$\mathbf{K}_x$`,` $\mathbf{K}_y$`,` $\mathbf{i}$`)` with final $\mathbf{i}$ to find $\tilde{\boldsymbol{\alpha}}_x$, $\tilde{\boldsymbol{\alpha}}_y$ and λ

Output: index vector $\mathbf{i}$, and eigenvectors $\tilde{\boldsymbol{\alpha}}_x$, $\tilde{\boldsymbol{\alpha}}_y$

5.3 Primal-Dual Sparse CCA

We introduce a new convex least square variant of CCA which seeks a semantic projection that uses as few relevant features as possible to explain as much correlation as possible. In previous studies, CCA had either been formulated in the primal or dual (kernel) representation for both views. These formulations, coupled with the need for sparsity, could prove insufficient when one desires or is limited to a primal-dual representation, i.e. one wishes to learn the correlation of words in one language that map to documents in another. We address these possible scenarios by giving SCCA in a primal-dual framework in which one view is represented in the primal and the other in the dual (kernel defined) representation. We compare SCCA with KCCA on a bilingual English-French and English-Spanish data-set for a mate retrieval task,

as well as a word generation task, i.e. using a new document from the testing corpus to find a sparse set of words from the training corpus that gives the highest correlation. We show that in the mate retrieval task SCCA performs as well as KCCA when the number of original features is small and SCCA outperforms KCCA when the number of original features is large. This emphasises SCCA's ability to learn the semantic space from only the relevant features. In the word generation task SCCA surpasses KCCA throughout, alleviating the need for a threshold on how many words are to be used. It is important to stress that the word generation task is different from document summarisation [21, 11], which retains the original semantic context of the document but in a summarised form. By contrast, we are interested in retrieving (generating) a small set of key words that best represent the document.

The motivation for formulating a primal-dual sparse CCA is largely intuitive when faced with real-world problems combined with the need to understand or interpret the found solutions. It can be interesting to notice that there are other real applications suitable to be dealt with this way.

- Enzyme prediction; in this problem one would like to uncover the relationship between the enzyme sequence, or more accurately the sub-sequences within each enzyme sequence that are highly correlated with the possible combination of the enzyme reactants. We would like to find a sparse primal weight representation on the enzyme sequence which correlates highly to sparse dual feature vector on the reactants. This will allow a better understanding of the enzyme structure relationship to reactions.

- Bilingual analysis; when learning the semantic relationship between two languages, we may want to understand how one language maps from the word space (primal) to the contextual document (dual) space of another language. In both cases we do not want a complete mapping from all the words to all possible contexts but to be able to extract an interpretable relationship from a sparse word representation from one language to a particular and specific context (or sparse combination of) in the other language.

- Brain analysis; here, one would be interested in finding a (primal) sparse voxel[1] activation map to some (dual) non-linear stimulus activation (such as musical sequences, images and various other multidimensional input). The potential ability in finding only the relevant voxels to the stimuli would remove the particularly problematic issue of thresholding the full voxel activation maps that are conventionally generated.

For this scope we limit ourselves to the bilingual textual based problems.

Throughout the paper we consider the setting when one is interested in a primal representation for the first view and a dual representation for the second view, although it is easily shown that the given derivations hold for the inverted case (i.e. a dual representation for the first view and a primal representation for the second view) and therefore is omitted.

Consider a sample from a pair of multivariate random vectors of the form $(\mathbf{x}_a^i, \mathbf{x}_b^i)$ each with zero mean where $i = 1, \ldots, \ell$. Let X_a and X_b be matrices whose columns are the corresponding training samples and let $K_b = X_b' X_b$ be the kernel matrix of the second view and $\mathbf{w}_b$ be expressed as a linear combination of the training examples $\mathbf{w}_b = X_b \mathbf{e}$.

[1] A voxel is a pixel representing the smallest three-dimensional point volume referenced in an fMRI (functional magnetic resonance imaging) image of the brain. It is usually approximately 3mm×3mm.

The primal-dual CCA problem can be expressed as a primal-dual Rayleigh quotient

$$\begin{aligned}\rho &= \max_{\mathbf{w}_a,\mathbf{w}_b} \frac{\mathbf{w}_a' X_a X_b' \mathbf{w}_b}{\sqrt{\mathbf{w}_a' X_a X_a' \mathbf{w}_a \mathbf{w}_b X_b X_b' \mathbf{w}_b}} \\ &= \max_{\mathbf{w}_a,\mathbf{e}} \frac{\mathbf{w}_a' X_a X_b' X_b \mathbf{e}}{\sqrt{\mathbf{w}_a' X_a X_a' \mathbf{w}_a \mathbf{e}' X_b' X_b X_b' X_b \mathbf{e}}} \\ &= \max_{\mathbf{w}_a,\mathbf{e}} \frac{\mathbf{w}_a' X_a K_b \mathbf{e}}{\sqrt{\mathbf{w}_a' X_a X_a' \mathbf{w}_a \mathbf{e}' K_b^2 \mathbf{e}}}, \end{aligned} \tag{5.13}$$

where we choose the primal weights $\mathbf{w}_a$ of the first representation and dual features $\mathbf{e}$ of the second representation such that the correlation ρ between the two vectors is maximised. As we are able to scale $\mathbf{w}_a$ and $\mathbf{e}$ without changing the quotient, the maximisation in equation (5.13) is equal to maximising $\mathbf{w}_a' X_a K_b \mathbf{e}$ subject to $\mathbf{w}_a' X_a X_a' \mathbf{w}_a = \mathbf{e}' K_b^2 \mathbf{e} = 1$. For simplicity let $X = X_a$, $\mathbf{w} = \mathbf{w}_a$ and $K = K_b$.

Having provided the initial primal-dual framework we proceed to reformulate the problem as a convex sparse least squares optimisation problem. We are able to show that maximising the correlation between the two vectors $K\mathbf{e}$ and $X'\mathbf{w}$ can be viewed as minimising the angle between them. Since the angle is invariant to rescaling, we can fix the scaling of one vector and then minimise the norm[2] between the two vectors

$$\min_{\mathbf{w},\mathbf{e}} \|X'\mathbf{w} - K\mathbf{e}\|^2 \tag{5.14}$$

subject to $\|K\mathbf{e}\|^2 = 1$, notice that we now no longer impose $\|X'\mathbf{w}\|^2 = 1$. This intuition is formulated in the following theorem,

Theorem 1. *Vectors $\mathbf{w}, \mathbf{e}$ are an optimal solution of equation (5.13) if and only if there exist μ, γ such that $\mu\mathbf{w}, \gamma\mathbf{e}$ are an optimal solution of equation (5.14).*

($\Rightarrow$) by contradiction. Suppose $\mathbf{w}, \mathbf{e}$ are an optimal solution of equation (5.13). Let γ be such that $\|\gamma K\mathbf{e}\|^2 = \gamma^2\|K\mathbf{e}\|^2 = 1$. Now let μ minimise

$$\|\mu X'\mathbf{w} - \gamma K\mathbf{e}\|^2.$$

and assume that $\mu\mathbf{w}, \gamma\mathbf{e}$ are not an optimal solution of equation (5.14). Then there exists $\hat{\mathbf{w}}, \hat{\mathbf{e}}$ such that

$$\|X'\hat{\mathbf{w}} - K\hat{\mathbf{e}}\|^2 < \|\mu X'\mathbf{w} - \gamma K\mathbf{e}\|^2$$

with $\|K\hat{\mathbf{e}}\| = 1$. Without loss of generality assume that scaling $\hat{\mathbf{w}}$ minimises the norm in the left-hand term of the previous inequality , so $K\hat{\mathbf{e}} - X'\hat{\mathbf{w}} \perp X'\hat{\mathbf{w}}$. $\gamma K\mathbf{e} - \mu X'\mathbf{w} \perp \mu X'\mathbf{w}$ because we chose μ to minimize the norm. Hence $\hat{\mathbf{w}}' X X'\hat{\mathbf{w}} = \hat{\mathbf{w}}' X K\hat{\mathbf{e}}$ and $\mu^2\mathbf{w}' X X'\mathbf{w} = \mu\gamma\mathbf{w}' X K\mathbf{e}$. Expanding the two norms in the inequality above, we have

$$\hat{\mathbf{w}}' X X'\hat{\mathbf{w}} - 2\hat{\mathbf{w}}' X K\hat{\mathbf{e}} + 1 < \mu^2\mathbf{w}' X X'\mathbf{w} - 2\mu\gamma\mathbf{w}' X K\mathbf{e} + 1 \Rightarrow -\hat{\mathbf{w}}' X K\hat{\mathbf{e}} < -\mu\gamma\mathbf{w}' X K\mathbf{e}.$$

It follows that

$$\frac{\hat{\mathbf{w}}' X K\hat{\mathbf{e}}}{\sqrt{\hat{\mathbf{w}}' X X'\hat{\mathbf{w}}\hat{\mathbf{e}}' K^2\hat{\mathbf{e}}}} = \sqrt{\hat{\mathbf{w}}' X K\hat{\mathbf{e}}} > \sqrt{\mu\gamma\mathbf{w}' X K\mathbf{e}} = \frac{\mu\gamma\mathbf{w}' X K\mathbf{e}}{\sqrt{\mu^2\mathbf{w}' X X'\mathbf{w}\gamma^2\mathbf{e}' K^2\mathbf{e}}} = \frac{\mathbf{w}' X K\mathbf{e}}{\sqrt{\mathbf{w}' X X'\mathbf{w}\mathbf{e}' K^2\mathbf{e}}}$$

contradicting the optimality of $\mathbf{w}, \mathbf{e}$.

[2] We define $\|\cdot\|$ to be the $2-$norm.

($\Leftarrow$) [by contradiction] Suppose for $\mathbf{w}, \mathbf{e}$ there exists μ, γ such that $\mu\mathbf{w}, \gamma\mathbf{e}$ are an optimal solution of equation (5.14) satisfying $\rho(\hat{\mathbf{w}}, \hat{\mathbf{e}}) > \rho(\mathbf{w}, \mathbf{e})$ for some $\hat{\mathbf{w}}, \hat{\mathbf{e}}$ where ρ is the correlation value. Rescale $\hat{\mathbf{w}}, \hat{\mathbf{e}}$ as in the first part with $\hat{\mu}, \hat{\gamma}$ and a reverse inequality follows for the norms

$$\|\hat{\mu}X'\hat{\mathbf{w}} - \hat{\gamma}K\hat{\mathbf{e}}\|^2 < \|\mu X'\mathbf{w} - \gamma K\mathbf{e}\|^2$$

contradicting the optimality of $\mu\mathbf{w}, \gamma\mathbf{e}$. □

Constraining the $2-$ norm of $K\mathbf{e}$ (or $X'\hat{\mathbf{w}}$) will result in a non convex problem, this is trivially shown by testing the Hessian for being positive/negitive semi-definite. Therefore, rather than constraining the $2-$norm of $K\mathbf{e}$ we fix the ∞-norm of the vector $\mathbf{e}$. This will be achieved by fixing each index $\mathbf{e}_k = 1$ in turn and optimising the $1-$norm of the remaining coefficients. We are also now able to optimise the $1-$norm of $\mathbf{w}$ without effecting the convexity of the problem. This gives the final primal-dual optimisation

$$\min_{\mathbf{w},\mathbf{e}} \|X'\mathbf{w} - K\mathbf{e}\|^2 + \mu\|\mathbf{w}\|_1 + \gamma\|\mathbf{e}\|_1 \tag{5.15}$$

subject to $\mathbf{e}_k = 1$, giving us an optimisation problem for each k.

We propose a novel method for solving the primal-dual optimisation in equation (5.15), where the suggested algorithm minimises the gap between the primal and dual solutions using a greedy search on $\mathbf{w}, \mathbf{e}$. This is carried out by iteratively solving between the primal and dual problems in turn. We give the proposed algorithm as the following high-level pseudo-code. A more complete description will follow later;

```
Repeat
1. Use dual solution to solve primal view optimisation
2. Check whether all primal constraints hold
3. Use primal solution to solve dual view optimisation
4. Check whether all dual constraints hold
5. Check whether 2. holds, IF not go to 1.
End
```

We have yet to address how to determine which elements in $\mathbf{w}, \mathbf{e}$ are to be non-zero. From the primal-dual derivation, a lower and upper bound is computed. Combining the derived bound with the constraints, provides us with a criterion for selecting the non-zero elements for both $\mathbf{w}$ and $\mathbf{e}$. Only the respective indices that violate the bound and the various constraints are needed to be set (so that the bound and constraints will no longer be violated).

We proceed with the crux of the matter and give the derivation of our problem. The minimisation

$$\min_{\mathbf{w},\mathbf{e}} \|X'\mathbf{w} - K\mathbf{e}\|^2 + \mu\|\mathbf{w}\|_1 + \gamma\|\mathbf{e}\|_1$$

subject to $\mathbf{e}_k = 1$ can be written as

$$\mathbf{w}'XX'\mathbf{w} + \mathbf{e}'K^2\mathbf{e} - 2\mathbf{w}'XK\mathbf{e} + \mu\|\mathbf{w}\|_1 + \gamma\|\mathbf{e}\|_1$$

subject to $\mathbf{e}_k = 1$, which in turn gives the corresponding Lagrangian

$$\mathcal{L} = \mathbf{w}'XX'\mathbf{w} + \mathbf{e}'K^2\mathbf{e} - 2\mathbf{w}'XK\mathbf{e} + \mu(\|\mathbf{w}\|_1) + \gamma(\mathbf{e}'\mathbf{j}) - \boldsymbol{\beta}'\mathbf{e},$$

subject to

$$\begin{aligned} \mu &\geq 0 \\ \gamma &\geq 0 \\ \boldsymbol{\beta} &\geq \mathbf{0}, \end{aligned}$$

where $\boldsymbol{\beta}$ is the Lagrangian variable upholding the constraint $\mathbf{e}_k = 1$ and μ, γ are positive scale factors as discussed in Theorem 1. To simplify the $1-$norm derivation we express $\mathbf{w}$ by its positive and negative components[3] such that $\mathbf{w} = \mathbf{w}^+ - \mathbf{w}^-$ subject to $\mathbf{w}^+, \mathbf{w}^- \geq 0$. This allows us to rewrite the Lagrangian as §

$$\begin{aligned} \mathcal{L} &= (\mathbf{w}^+ - \mathbf{w}^-)'XX'(\mathbf{w}^+ - \mathbf{w}^-) + \mathbf{e}'K^2\mathbf{e} - 2(\mathbf{w}^+ - \mathbf{w}^-)'XK\mathbf{e} \\ &\quad -\boldsymbol{\alpha}^{-\prime}\mathbf{w}^- - \boldsymbol{\alpha}^{+\prime}\mathbf{w}^+ - \boldsymbol{\beta}'\mathbf{e} + \gamma(\mathbf{e}'\mathbf{j}) + \mu((\mathbf{w}^+ + \mathbf{w}^-)'\mathbf{j}), \end{aligned} \tag{5.16}$$

where $\mathbf{j}$ is the all ones vector. The corresponding Lagrangian in equation (5.16) is subject to

$$\begin{aligned} \mu &\geq 0 \\ \gamma &\geq 0 \\ \boldsymbol{\alpha}^+ &\geq \mathbf{0} \\ \boldsymbol{\alpha}^- &\geq \mathbf{0} \\ \boldsymbol{\beta} &\geq \mathbf{0}. \end{aligned}$$

The two new Lagrangian variables $\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-$ are to uphold the constraints on $\mathbf{w}^+, \mathbf{w}^-$. In the following section we will show that the constraints on the Lagrangian variables will form the baseline criterion for selecting the non-zero elements from $\mathbf{w}$ and $\mathbf{e}$.

Taking derivatives of equation (5.16) in respect to $\mathbf{w}^+$, $\mathbf{w}^-$, $\mathbf{e}$ and equating to zero gives

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}^+} &= 2XX'(\mathbf{w}^+ - \mathbf{w}^-) - 2X'K\mathbf{e} - \boldsymbol{\alpha}^+ + \mu\mathbf{j} = \mathbf{0} \\ \frac{\partial \mathcal{L}}{\partial \mathbf{w}^-} &= -2XX'(\mathbf{w}^+ - \mathbf{w}^-) + 2X'K\mathbf{e} - \boldsymbol{\alpha}^- + \mu\mathbf{j} = \mathbf{0} \\ \frac{\partial \mathcal{L}}{\partial \mathbf{e}} &= 2K^2\mathbf{e} - 2KX'\mathbf{w} - \boldsymbol{\beta} + \gamma'\mathbf{j} = \mathbf{0}, \end{aligned} \tag{5.17}$$

adding the first two equations gives

$$\begin{aligned} \boldsymbol{\alpha}^+ &= 2\mu\mathbf{j} - \boldsymbol{\alpha}^- \\ \boldsymbol{\alpha}^- &= 2\mu\mathbf{j} - \boldsymbol{\alpha}^+, \end{aligned}$$

implying a lower and upper bound on $\boldsymbol{\alpha}^-, \boldsymbol{\alpha}^+$ of

$$\begin{aligned} \mathbf{0} &\leq \boldsymbol{\alpha}^- \leq 2\mu\mathbf{j} \\ \mathbf{0} &\leq \boldsymbol{\alpha}^+ \leq 2\mu\mathbf{j}. \end{aligned}$$

We use the bound on $\boldsymbol{\alpha}$ as an indication as to which $\mathbf{w}$'s are to be updated by only updating the $\mathbf{w}_i$'s whose corresponding $\boldsymbol{\alpha}_i$ violates the bound. Similarly, we only update $\mathbf{e}_i$ that has a corresponding $\boldsymbol{\beta}_i$ value smaller than 0.

We are able to rewrite the derivative with respect to $\mathbf{w}^+$ in terms of $\boldsymbol{\alpha}^-$

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}^+} &= 2XX'(\mathbf{w}^+ - \mathbf{w}^-) - 2X'K\mathbf{e} - 2\mu\mathbf{j} + \boldsymbol{\alpha}^- + \mu\mathbf{j} \\ &= 2XX'(\mathbf{w}^+ - \mathbf{w}^-) - 2X'K\mathbf{e} - \mu\mathbf{j} + \boldsymbol{\alpha}^-. \end{aligned}$$

[3] This means that $\mathbf{w}^+/\mathbf{w}^-$ will only have the positive/negative values of $\mathbf{w}$ and zero elsewhere.

We wish to compute the update rule for the selected indices of $\mathbf{w}$. Taking the second derivatives of equation (5.16) in respect to $\mathbf{w}^+$ and $\mathbf{w}^-$, gives

$$\begin{aligned}\frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}^+} &= 2XX' \\ \frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}^-} &= -2XX',\end{aligned}$$

so for the $\mathbf{i}_i$, the unit vector with entry 1, we have an exact Taylor series expansion t^+ and t^- respectively for $\mathbf{w}^+$ and $\mathbf{w}^-$ as

$$\begin{aligned}\hat{\mathcal{L}}(\mathbf{w}^+ + t^+\mathbf{i}_i) &= \mathcal{L} + \frac{\partial \mathcal{L}}{\partial \mathbf{w}_i^+}t^+ + \frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_i^+}(t^+)^2 \\ \hat{\mathcal{L}}(\mathbf{w}^- + t^-\mathbf{i}_i) &= \mathcal{L} + \frac{\partial \mathcal{L}}{\partial \mathbf{w}_i^-}t^- + \frac{\partial^2 \mathcal{L}}{\partial \mathbf{w}_i^-}(t^-)^2\end{aligned}$$

giving us the exact update for $\mathbf{w}^+$ by setting

$$\begin{aligned}\frac{\partial \hat{\mathcal{L}}(\mathbf{w}^+ + t^+\mathbf{i}_i)}{\partial t^+} &= \left(2XX'(\mathbf{w}^+ - \mathbf{w}^-) - 2X'K\mathbf{e} - \boldsymbol{\alpha}^+ + \mu\mathbf{j}\right)_i + 4(XX')_{ii}t^+ = 0 \\ \Rightarrow t^+ &= \frac{1}{4(XX')_{ii}}\left[2X'K\mathbf{e} - 2XX'(\mathbf{w}^+ - \mathbf{w}^-) - \boldsymbol{\alpha}^- + \mu\mathbf{j}\right]_i,\end{aligned}$$

where $(\cdot)_i$ or $[\cdot]_i$ refers to the i'th index in a vector and $(\cdot)_{ii}$ refers to the i'th element on the diagonal of a matrix. Therefore the update for $\mathbf{w}^+$ is $\Delta\mathbf{w}^+ = t^+$. We also compute the exact update for $\mathbf{w}^-$ as

$$\begin{aligned}\frac{\partial \hat{\mathcal{L}}(\mathbf{w}^- + t^-\mathbf{i}_i)}{\partial t^-} &= \left(-2XX'(\mathbf{w}^+ - \mathbf{w}^-) + 2X'K\mathbf{e} - \boldsymbol{\alpha}^- + \mu\mathbf{j}\right)_i + 4(XX')_{ii}t^- = 0 \\ \Rightarrow t^- &= -\frac{1}{4(XX')_{ii}}\left[2X'K\mathbf{e} - 2XX'(\mathbf{w}^+ - \mathbf{w}^-) - \boldsymbol{\alpha}^- + \mu\mathbf{j}\right]_i,\end{aligned}$$

similarly, the update for $\mathbf{w}^-$ is $\Delta\mathbf{w}^- = t^-$. Recall that $\mathbf{w} = (\mathbf{w}^+ - \mathbf{w}^-)$, hence the update rule for $\mathbf{w}$ is

$$\hat{\mathbf{w}}_i \leftarrow \mathbf{w}_i + (\Delta\mathbf{w}_i^+ - \Delta\mathbf{w}_i^-).$$

Therefore we find that the new value of $\mathbf{w}$ should be

$$\hat{\mathbf{w}}_i \leftarrow \mathbf{w}_i + \frac{1}{2(XX')_{ii}}\left[2X'K\mathbf{e} - 2XX'\mathbf{w} - \boldsymbol{\alpha}^- + \mu\mathbf{j}\right]_i.$$

We must also consider the update of $\mathbf{w}_i$ when $\boldsymbol{\alpha}_i$ is within the constraints and $\mathbf{w}_i \neq 0$, i.e. previously $\boldsymbol{\alpha}_i$ had violated the constraints triggering the updated of $\mathbf{w}_i$ to be non zero. Notice from equation (5.17) that

$$2(XX')_{ii}\mathbf{w}_i + 2\sum_{j\neq i}(XX')_{ij}\mathbf{w}_j = 2(X'K\mathbf{e})_i - \boldsymbol{\alpha}_i + \mu.$$

It is easy to observe that the only component which can change is $2(XX')_{ii}\mathbf{w}_i$, therefore as we need to update $\mathbf{w}_i$ towards zero. Hence when $\mathbf{w}_i > 0$ the update is

$$\begin{aligned}2(XX')_{ii}\Delta\mathbf{w}_i &= 2\mu - \alpha_i \\ \Delta\mathbf{w}_i &= \frac{2\mu - \alpha_i}{2(XX')_{ii}}\end{aligned}$$

else when $\mathbf{w}_i < 0$ then the updated is

$$\begin{aligned} 2(XX')_{ii}\Delta\mathbf{w}_i &= 0 - \alpha_i \\ \Delta\mathbf{w}_i &= \frac{-\alpha_i}{2(XX')_{ii}} \end{aligned}$$

where the update rule is $\hat{\mathbf{w}}_i \leftarrow \mathbf{w}_i - \Delta\mathbf{w}_i$. In the updating of $\mathbf{w}$ we ensure that $\mathbf{w}_i, \hat{\mathbf{w}}_i$ do not switch sign, i.e. we will always pause on zero before updating in any new direction.

We continue by taking second derivatives of the Lagrangian in equation (5.16) with respect to $\mathbf{e}$, which gives

$$\frac{\partial^2\mathcal{L}}{\partial\mathbf{e}} = 2K^2,$$

so for the $\mathbf{i}_i$ the unit vector with entry 1 we have an exact Taylor series expansion

$$\hat{\mathcal{L}}(\mathbf{e} + t\mathbf{i}_i) = \mathcal{L} + \frac{\partial\mathcal{L}}{\partial\mathbf{e}_i}t + \frac{\partial^2\mathcal{L}}{\partial\mathbf{e}_i}(t)^2$$

giving us the following update rule for $\mathbf{e}_i$

$$\begin{aligned} \frac{\partial\hat{\mathcal{L}}(\mathbf{e} + t\mathbf{i}_i)}{\partial t} &= (2K^2\mathbf{e} - 2KX'\mathbf{w} - \boldsymbol{\beta} + \gamma'\mathbf{j})_i + 4K_{ii}^2 t = 0 \\ \Rightarrow t &= \frac{1}{4K_{ii}^2}\left[2KX'\mathbf{w} - 2K^2\mathbf{e} + \boldsymbol{\beta} - \gamma'\mathbf{j}\right]_i, \end{aligned}$$

the update for $\mathbf{e}$ is $\Delta\mathbf{e} = t$. The new value of $\mathbf{e}$ should be

$$\hat{\mathbf{e}}_i \leftarrow \mathbf{e}_i + \frac{1}{4K_{ii}^2}\left[2KX'\mathbf{w} - 2K^2\mathbf{e} + \boldsymbol{\beta} - \gamma'\mathbf{j}\right]_i.$$

Combining all the pieces we give a more complete description of the algorithm

```
Repeat
1. Check the primal indices that violate the bound and constraints
2. Solve primal view optimisation for the selected indices
3. Re-compute variables using the solution from 2.
4. Check the dual indices that violate the bound and constraints
5. Solve dual view optimisation for the selected indices
6. Re-compute variables using the solution from 5.
7. Check whether all the constraints hold AND
    Check that the difference between the previously computed primal,
    dual variables is small, IF not go to 1.
End
```

We give the complete algorithm as pseudo-code in Algorithm 6. To ensure orthogonality of the extracted features [19] for each $\mathbf{e}$ and corresponding $\mathbf{w}$, we compute the residual matrices X_j, $j = 1, \ldots, k$ by projecting the columns of the data X_j' onto the orthogonal complement of $X_j X_j'\mathbf{w}_j$, a procedure known as deflation,

$$X_{j+1} = X_j\left(I - \mathbf{u}_j\mathbf{p}_j'\right),$$

where U is a matrix with columns $\mathbf{u}_j = X_j X_j' \mathbf{w}_j$ and P is a matrix with columns $\mathbf{p}_j = \frac{X_j X_j' \mathbf{u}_j}{\mathbf{u}_j' X_j X_j' \mathbf{u}_j}$. The extracted projection directions can be computed (following [19]) as $U(P'U)^{-1}$. Similarly we deflate for the dual view

$$K_{j+1} = \left(I - \frac{\tau_j \tau_j'}{\tau_j' \tau_j}\right) K_j \left(I - \frac{\tau_j \tau_j'}{\tau_j' \tau_j}\right),$$

where $\tau_j = K_j'(K_j' \mathbf{e}_j)$ and compute the projection directions as $B(T'KB)^{-1}T$ where B is a matrix with columns $K_j \mathbf{e}_j$ and T has columns τ_j. We do not address the order selection of k while in he experimental Section we iteratively solve for all $k = 1, \ldots, n$

5.4 Experiments with SKCCA

In this section we used English-Spanish corpus consisting of $500, 3000, 7000$ training samples with $40,629$ English features and $57,796$ Spanish features. The features represent the number of words in each language. Both corpuses are pre-processed with Term Frequency Inverse Document Frequency (TFIDF) followed by centering and normalisation. The linear kernel was used for both views. The KCCA regularisation parameter was heuristically fixed to be 0.03,

The retrieval is assessed using mate retrieval as task, and two measures. We call the first measure the "window" measure: it counts the number of times documents are retrieved using their pair as a query. The window size we use is 10 and if the pair can be retrieved within the top 10 correlation values then we count the document as being retrieved, otherwise it is considered not to have been retrieved (error). However, a problem with this measure is that it does not take into account the positions from which the documents were retrieved. The second measure rectifies this problem and is called "average precision" and is the standard average precision method employed in Information Retrieval tasks. Here we associate a weight for the position that a document may be retrieved from in the second language. Position 1 has the highest weight and position m (where m is the number of test examples) has the lowest weight. The average precision sums up weights of the positions that the documents are retrieved from and takes an average as the final measure of retrieval rate. Clearly, the average precision measure is more robust because it takes into account the position of the documents retrieved. This is in contrast with the window method that is content with a document that is contained in the top 10 (say) highest correlation values.

Data Set	KCCA			SKCCA			SKCCA (faster)		
	train	test	total	train	test	total	train	test	total
Small	8	20	28	9	5	14	2	5	7
Medium	1707	1995	3702	334	224	558	59	224	283
Large	24693	27733	52426	5242	698	5940	1873	695	2568

Table 5.1: Training and test times in seconds for small, medium and large data set sizes (English-Spanish data)

As you can see from Figures 5.1, 5.2 and 5.3 the KCCA algorithm requires different parameter values in order to produce stable results. Also, as the data sets increases, so do the parameter values, where in Figure 5.3 the best parameter value is 0.75. We hypothesise that as the size of the data set increases than so too does the value of the regularisation parameter needed in order to obtain good generalisation. However, the upper bound of this parameter value is 1 and so at some point this value will be reached and KCCA may start to generate trivial solutions where it will begin to overfit (as was shown for the smaller parameter values in all three plots). We cannot however show that this assertion is correct for very large

Algorithm 6 The SCCA algorithm

input: Data matrix $\mathbf{X} \in \mathbb{R}^{N \times \ell}$, Kernel matrix $\mathbf{K} \in \mathbb{R}^{\ell \times \ell}$ and $\mathbf{e}_k = 1$.

% Initialisation:
$\mathbf{w} = \mathbf{0}, \mathbf{j} = \mathbf{1}$

$\mu = \frac{1}{M} \sum_i^M |(2XK\boldsymbol{e})_i|$
$\gamma = \frac{1}{N} \sum_i^N |(2K^2\boldsymbol{e})_i|$

$\boldsymbol{\alpha}^- = 2X'K\mathbf{e} + \mu\mathbf{j}$
$I = (\boldsymbol{\alpha} < \mathbf{0}) \,|||\, (\boldsymbol{\alpha} > 2\mu\mathbf{j})$

repeat
  % Update the found weight values:
  repeat
    for $i = 1$ to length of I **do**
      if $\boldsymbol{\alpha}_{I_i} > 2\mu$ **then**
        $\boldsymbol{\alpha}_{I_i} = 2\mu$
        $\hat{\mathbf{w}}_{I_i} \leftarrow \mathbf{w}_{I_i} + \frac{1}{2(XX')_{I_i,I_i}} \left[2(X'K\mathbf{e})_{I_i} - 2(XX'\mathbf{w})_{I_i} - \boldsymbol{\alpha}^-_{I_i} + \mu\right]$
      else if $\boldsymbol{\alpha}_{I_i} < 0$ **then**
        $\boldsymbol{\alpha}_{I_i} = 0$
        $\hat{\mathbf{w}}_{I_i} \leftarrow \mathbf{w}_{I_i} + \frac{1}{2(XX')_{I_i,I_i}} \left[2(X'K\mathbf{e})_{I_i} - 2(XX'\mathbf{w})_{I_i} - \boldsymbol{\alpha}^-_{I_i} + \mu\right]$
      else
        if $\mathbf{w}_{I_i} > 0$ **then**
          $\hat{\mathbf{w}}_{I_i} \leftarrow \mathbf{w}_{I_i} - \frac{2\mu - \boldsymbol{\alpha}_{I_i}}{2(XX')_{I_i,I_i}}$
        else if $\mathbf{w}_{I_i} < 0$ **then**
          $\hat{\mathbf{w}}_{I_i} \leftarrow \mathbf{w}_{I_i} + \frac{\boldsymbol{\alpha}_{I_i}}{2(XX')_{I_i,I_i}}$
        end if
      end if
      if $sign(\mathbf{w}_{I_i}) \neq sign(\hat{\mathbf{w}}_{I_i})$ **then**
        $\mathbf{w}_{I_i} = 0$
      else
        $\mathbf{w}_{I_i} = \hat{\mathbf{w}}_{I_i}$
      end if
    end for
  until convergence over $\mathbf{w}$

  % Find the dual values that are to be updated
  $\boldsymbol{\beta} = 2K^2\mathbf{e} - 2KX\mathbf{w} + \gamma\mathbf{j}$
  $J = (\boldsymbol{\beta} < \mathbf{0})$

  % Update the found dual projection values
  repeat
    for $i = 1$ to length of J **do**
      if $J_i \neq k$ **then**
        $\mathbf{e}_{J_i} \leftarrow \mathbf{e}_{J_i} + \frac{1}{4K^2_{J_i J_i}} \left[2(KX'\mathbf{w})_{J_i} - 2(K^2\mathbf{e})_{J_i} - \gamma\right]$
        if $\mathbf{e}_{J_i} < 0$ **then**
          $\mathbf{e}_{J_i} = 0$
        else if $\mathbf{e}_{J_i} > 1$ **then**
          $\mathbf{e}_{J_i} = 1$
        end if
      end if
    end for
  until convergence over $\mathbf{e}$

  % Find the weight values that are to be updated
  $\boldsymbol{\alpha}^- = 2X'K\mathbf{e} - 2XX'\mathbf{w} + \mu\mathbf{j}$
  $I = (\boldsymbol{\alpha} < \mathbf{0}) \,|||\, (\boldsymbol{\alpha} > 2\mu\mathbf{j})$

until convergence

Output: Feature directions $\mathbf{w}, \mathbf{e}$

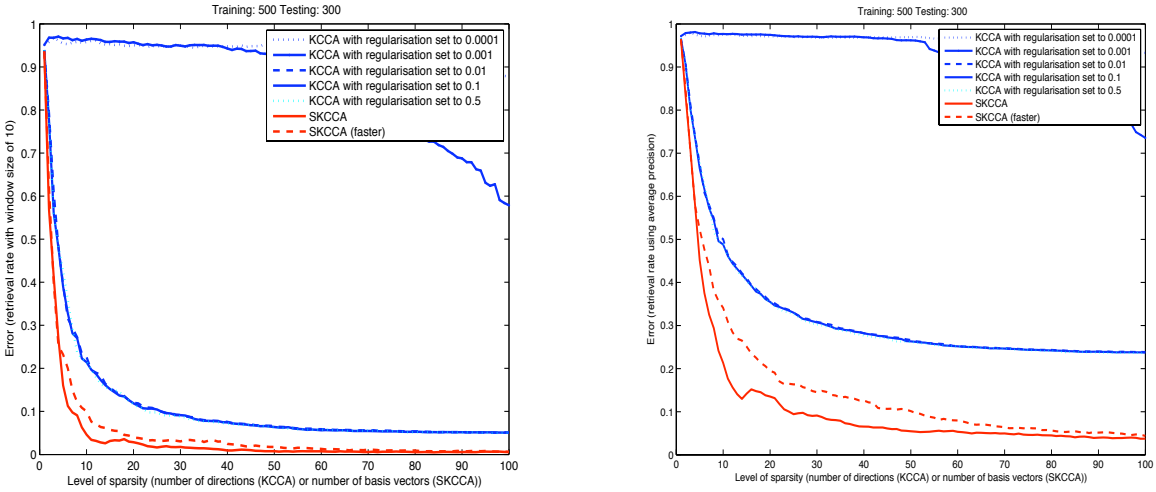

Figure 5.1: A small sized data set – average test error for retrieval over 5 different splits of the data. Left: average error using window (method) of size 10. Right: average error using average precision.

data sets because it is not possible to train on such data sets ($>$ 20000 training points, say). This was one of the motivations for proposing a sparse variant and the figures show that the SKCCA is stable throughout the experiments with different sized data sets. This means that sparsity in KCCA is a good means of regularisation. The sparse functions also deliver solutions that are close to KCCA for the larger data set but with one major difference, the speed in which it carries out these computations. Observe, from Table 5.1 that the SKCCA is up to 10 times faster than KCCA in terms of training and testing time and that the SKCCA (faster), without deflation, is up to 20 times quicker (for the large data set).

A future research direction is to train SKCCA on data sets with $>>$ 20000 data points. However, currently one issue that remains is the need to store the full kernel matrix in memory. This can however, be resolved by using a sub-sampling method to identify an initial subset from which to choose basis vectors from and avoid the need to store the entire kernel. Initial experiments towards this research direction are encouraging and show no deterioration in generalisation ability from both SKCCA algorithms presented above.

5.5 Experiments with Primal-dual sparse CCA

In the following experiments we use two paired English-French and English-Spanish corpora. The English-French corpus consists of 300 samples with 2637 English features and 2951 French features while the English-Spanish corpus consists of $1,000$ samples with $40,629$ English features and $57,796$ Spanish features. The features represent the number of words in each language. Both corpuses are pre-processed with Term Frequency Inverse Document Frequency (TFIDF) followed by centering and normalisation. The linear kernel was used for the dual view. The KCCA regularisation parameter was heuristically fixed to be 0.03, while SCCA had no parameters to tune.

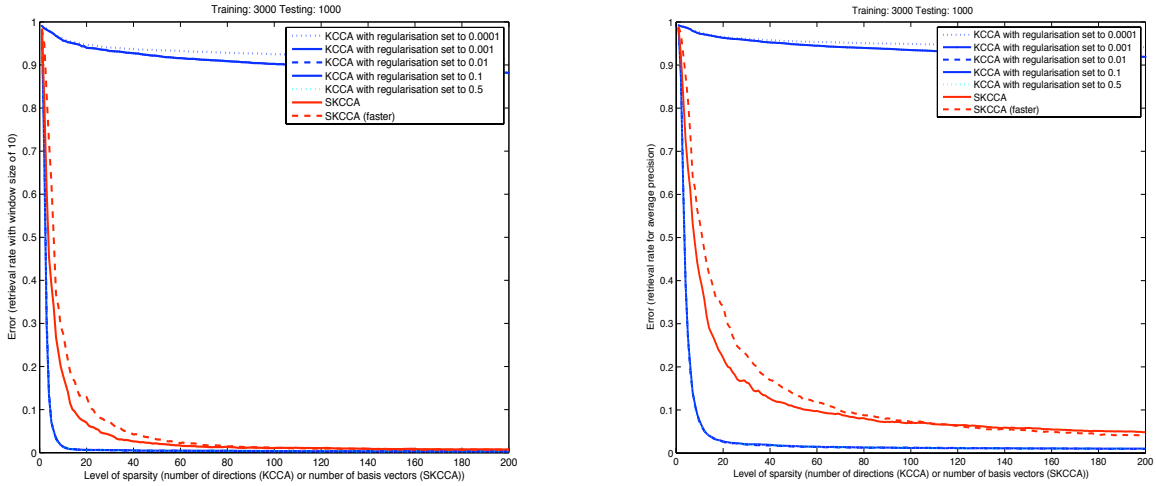

Figure 5.2: A medium sized data set – average test error for retrieval over 5 different splits of the data. Left: average error using window (method) of size 10. Right: average error using average precision.

5.5.1 Mate Retrieval

Our initial experiment is of mate-retrieval, in which a document from the test corpus of one language is considered as the query and only the mate document from the other language is considered relevant. In the following experiments the results are an average of retrieving the mate for both English and French (English and Spanish) and have been repeated 10 times with a random train-test split.

We compute the mate-retrieval by projecting the query document as well as the paired (other language) test documents into the learnt semantic space where the inner product between the projected data is computed. Let q be the query in one language and K_s the kernel matrix of the inner product between the second language's testing and training documents

$$l = \left\langle \frac{q'\mathbf{w}}{\|q'\mathbf{w}\|}, \frac{K_s\mathbf{e}}{\|K_s\mathbf{e}\|} \right\rangle.$$

The resulting inner products l are then sorted by value. We measure the success of the mate-retrieval task using average precision, this assesses where the correct mate within the sorted inner products l is located. Let I_j be the index location of the retrieved mate from query q_j, the average precision p is computed as

$$p = \frac{1}{M}\sum_{j=1}^{M}\frac{1}{I_j},$$

where M is the number of query documents.

We start by giving the results for the English-French mate-retrieval as shown in Figure 5.4. The left plot depicts the average precision ($\pm$ standard deviation) when 50 documents are used for training and the remaining 250 are used as test queries. The right plot in Figure 5.4 gives the average precision ($\pm$ standard deviation) when 100 documents are used for training and the remaining 200 for testing. It is interesting to observe that even though SCCA does not learn the common semantic space using all the features (plotted

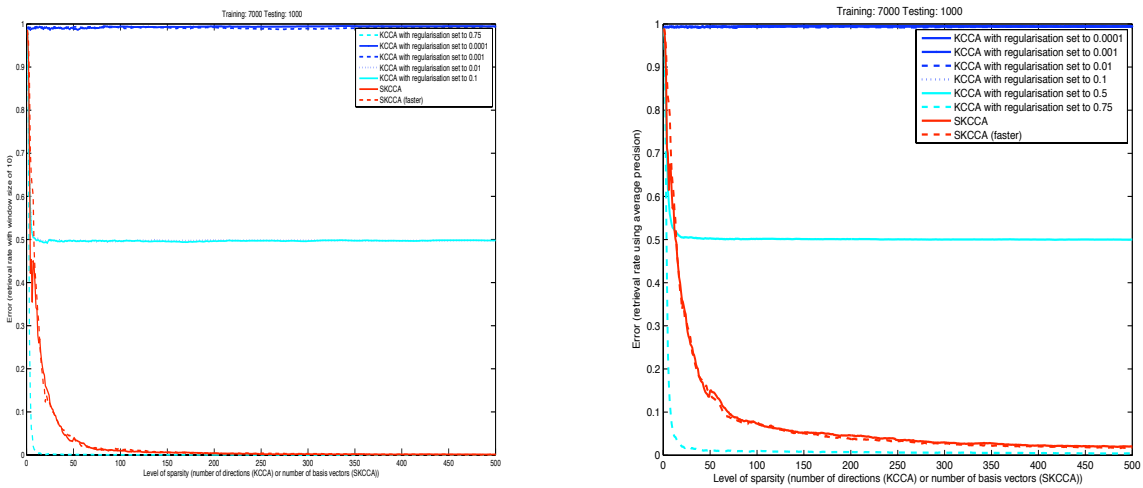

Figure 5.3: A large sized data set – average test error for retrieval over 2 different splits of the data. Left: average error using window (method) of size 10. Right: average error using average precision.

in Figure 5.5) for either primal or dual views (although SCCA will use full dual features when using the full number of projections) its error is extremely similar to that of KCCA and in fact converges with it when a sufficient number of projections are used. It is important to emphasise that KCCA uses the full number of documents (50 and 100) and the full number of words (an average of 2794 for both languages) to learn the common semantic space. For example, following the left plot in Figure 5.4 and the additional plots in Figure 5.5 we are able to observe that when 35 projections are used KCCA and SCCA show a similar error. However, SCCA uses approximately 142 words and 42 documents to learn the semantic space, while KCCA uses 2794 words and 50 documents.

The second mate-retrieval experiment uses the English-Spanish paired corpus. In each run we randomly split the 1000 samples into 100 training and 900 testing paired documents. The results are plotted in Figure 5.6 where we are clearly able to observe SCCA outperforming KCCA throughout. We believe this to be a good example of when too many features hinder the learnt semantic space. The level of SCCA sparsity is plotted in Figure 5.7. In comparison to KCCA which uses all words $(49,212)$ SCCA uses a maximum of 460 words.

The performance of SCCA, especially in the latter English-Spanish experiment, shows that we are indeed able to extract meaningful semantics between the two languages, using only the relevant features.

5.5.2 Multilingual Document Annotation

We are faced with the problem of finding a set of words or alternatively creating a new document d^* from the paired language that best matches our query. In the case of SCCA we find this to be straightforward. Let $\mathbf{k}$ be the vector of inner products between the query and training documents, we solve the following optimisation

$$\min_{\mathbf{w}} \|X'\mathbf{w} - \mathbf{k}\|^2 + \mu\|\mathbf{w}\|_1,$$

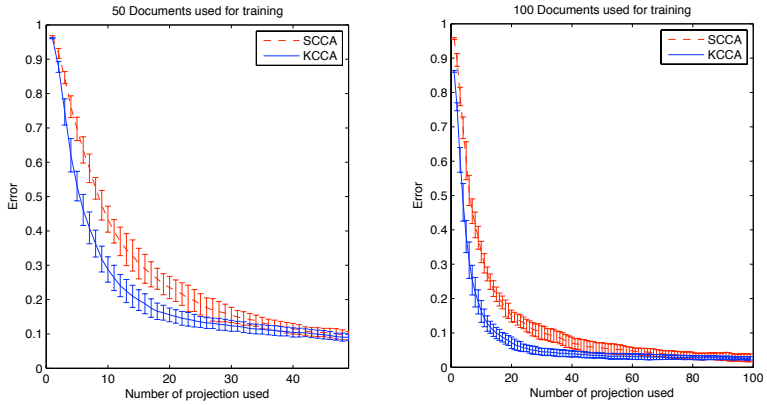

Figure 5.4: English-French: The average precision error (1-p) with $\pm$ standard division error bars for SCCA and KCCA for different number of projections used for the mate-retrieval task. The left figure is for 50 training and 250 testing documents while the right figure is for 100 training and 200 testing documents.

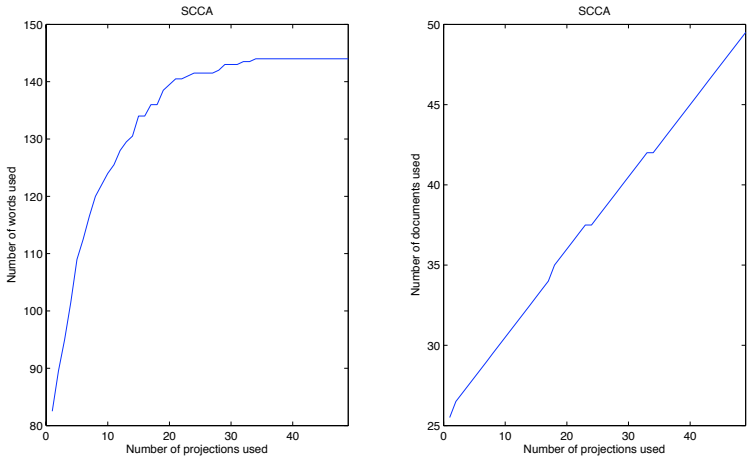

Figure 5.5: English-French: Level of Sparsity - The following figure is an extension of Figure 5.4 which uses 50 documents for training. The left figure plots the number of words used while the right figure plots the number of documents used with the number of projections. For reference, KCCA uses all the words (average of 2794) and documents (50) for all number of projections.

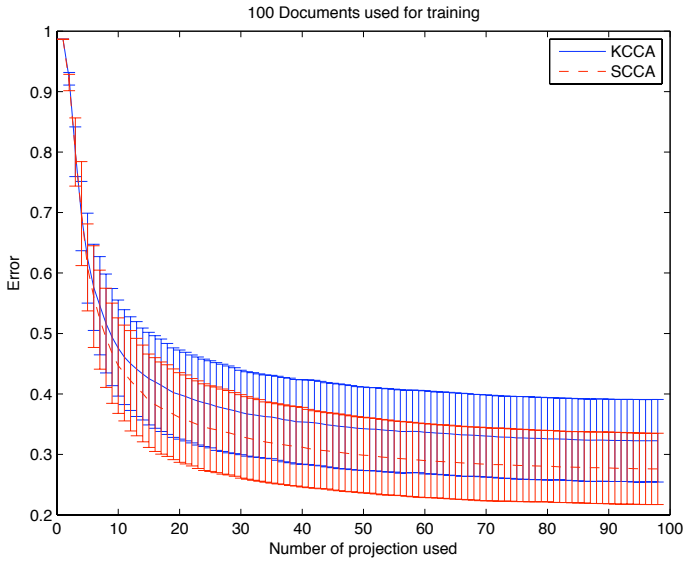

Figure 5.6: English-Spanish: The average precision error (1-p) with $\pm$ standard division error bars of SCCA and KCCA for different number of projections used for the mate-retrieval task. We use 100 documents for training and 900 for testing documents.

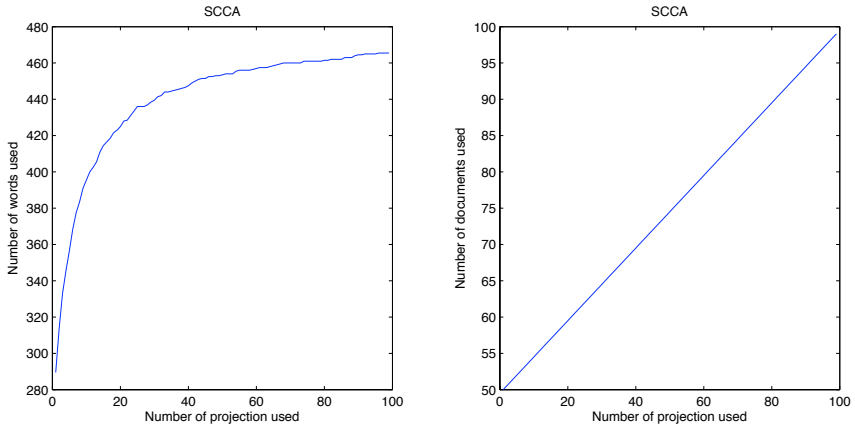

Figure 5.7: English-Spanish: Level of Sparsity - The following figure is an extension of Figure 5.6 which uses 100 documents for training. The left figure plots the number of words used and while the right figure plots the number of documents used with increasing number of projections. For reference, KCCA uses all the words (average of $49,212$) and documents (100) for all number of projections.

which is equivalent to the optimisation in equation (5.15). Notice that we are able to drop $\mathbf{e}$ from the optimisation as $\mathbf{e}$ is a unit vector and in this formulation we only have a single query. The new generate document is given by the sparse weight vector $d^* = \mathbf{w}^+$. We only take the positive spectrum of the vector of $\mathbf{w}$ ($\mathbf{w}^+$) as the negative values ($\mathbf{w}^-$) will correspond to words which are most irrelevant to the query[4]. This could be thought of as, given the similarity between the query and the training documents, finding (or approximate) a sparse set of words from the training corpus such that the error (correlation) is minimised (maximised).

We compare the above to KCCA by proposing the following method for generating a new document d^*. Based on the idea of CCA we are looking for a vector that has maximum covariance to the query with respect to the weight matrices α and β. We do this by reducing to a case similar to SCCA, where we work in dual space for the query language and in primal space for the other language. Let $f = K_a^i \boldsymbol{\alpha}$, where the vector K_a^i contains the kernelised inner products between the query i and the documents occurring in the training set. We have $\max_{d^*} \langle f, W_b' d^* \rangle$, where $W_b = X_b \boldsymbol{\beta}$ is the matrix containing the weight vectors as rows. The need to use the weight vectors for the documents limits us to the use of linear kernels. In order to create a combination of words, we let the vector d^* be a convex combination of the columns of the identity matrix, thus it satisfies the constraints

$$\sum_{i=1}^{n} d_i^* = 1, \quad d_i^* \geq 0 \;\; i = 1, \ldots, n. \tag{5.18}$$

The problem becomes $\max_{d^*} f' W_b' d^*$ under the constraints. Let $c = f' W_b'$ we have $\max_{d^*} c d^*$. Due to the constraints in equation (5.18) the components of the optimum solution d^* is equal to

$$d_i^* = \begin{cases} 1 & i = \arg\max_j c_j, \\ 0 & \text{otherwise.} \end{cases}$$

This generates a document containing a single word. We modify the original maximisation problem by constraining $d_i^* < c < 1$. The optimum solution will now include words above a threshold T. The new relaxed formulation will generate a document with a varying number of words, depending on T. We are able to use the value of c_j to rank the relevance of the selected words. We do this by sorting the values of $\mathbf{c}$ and taking the keywords relating to the largest values of $\mathbf{c}$ above threshold T. We modify T such that the number of words selected is equivalent to the number of words generated by the SCCA method for the same query (note $c = \frac{1}{n_i}$ where $n_i = \#$ words chosen by SCCA).

Results

In the following experiments we use a 'leave-one-out' procedure. The performance is measured by assessing whether the generated words pre-existed within the query. In other words, given a query q_i in one language, its paired translation $t(q_i)$ in the other language, and its corresponding generated set of words $(d^*)_i$ the performance is equal to

$$p = \frac{1}{M} \sum_{i=1}^{M} \frac{\#\{(d^*)_i \in t(q_i)\}}{n_i},$$

where M is the number of queries, and n_i is the number of words generated by SCCA. In the following plots we will compare the performance of SCCA with the single weight vector per query in comparison to the change in KCCA's performance when increasing the number of projections used. The presented

[4] We will have negative words as the data is centred.

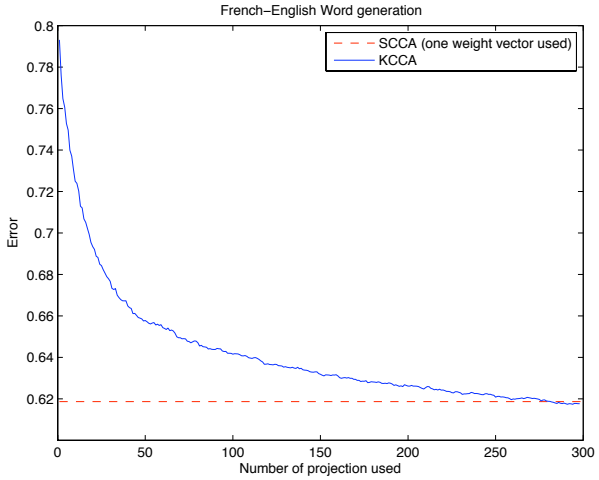

Figure 5.8: English-French: In the following figure we display the mean average error $(1 - p)$ of KCCA generating the correct set of words to a query from the paired language for a different number of projections used. This is compared to the average error of SCCA which only uses a direction per query. Both methods have been run on a 'leave-one-out' basis for 300 samples.

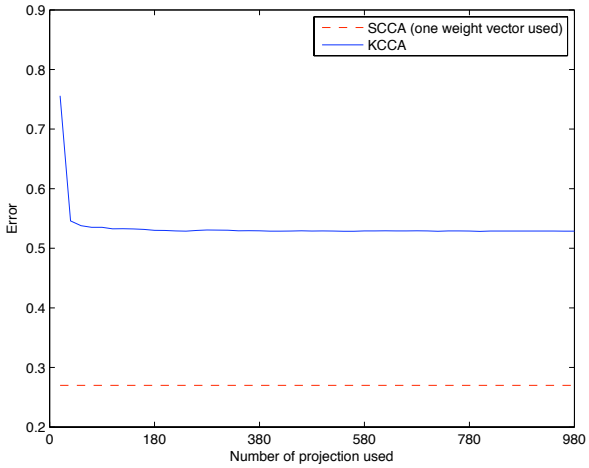

Figure 5.9: English-Spanish: In the following figure we display the mean average error $(1 - p)$ of KCCA generating the correct set of words to a query from the paired language using a different number of projections. This is compared to the average error of SCCA which only uses a direction per query. Both methods have been run on a 'leave-one-out' basis for 1000 samples.

results are an average for correctly generating words in language L_1 that maximise the correlation with the query from language L_2 and of correctly generating words in language L_2 that maximise the correlation to the query from language L_1, i.e. the result is averaged over both languages.

In Figures 5.8 and 5.9 we plot the mean average error (i.e. $1 - p$) for generating a set of words from the paired language of the query. The error for the English-French corpus is plotted in Figure 5.8 whereas the error for the English-Spanish corpus is plotted in Figure 5.9. We are able to observe across the plots in Figures 5.8 and 5.9 that KCCA achieves a similar performance to SCCA when the full number of projection directions are used to derive the common semantic space. We believe these results to be very promising. Due to the leave-one-out structure of the experiments, KCCA requires computation of all the feature projections on the training data for each new query (299 training samples in the English-French and 999 training samples in the English-Spanish) whereas SCCA computes a single sparse weight vector per query and achieves the same, if not lower, error than KCCA.

We are able to summarise the potential benefits of SCCA in the following list

- SCCA only requires a single weight vector whereas KCCA requires the full semantic information. One could argue that, in a different experimental setting, the full semantic information as needed by KCCA needs to be computed only once and retained in memory for usage on any number of queries. Whereas SCCA computes a new direction per query. It is easy to see how memory retention of the full semantic information could become memory inefficient when large training corpora are used.
- SCCA produces a sparse weight representation which in turn automatically drives the number of words used as the retrieved set, whereas a heuristic threshold is required in KCCA. We obtain better interpretability of the obtained solution

Chapter 6

Conclusion

The deliverable has introduced a number of innovations in one of the core technologies for CLTIA. Correlation analysis has proved itself an effective tool in developing language independent representations that can be used in a number of cross-lingual processing tasks. The deliverable has addressed three key shortcomings of the approach:

1. restriction to pairs of languages;
2. scaling beyond tens of thousands of documents;
3. mismatch between the statistics of text and that assumed by standard correlation methods.

In all three areas significant advances have been presented: an efficient extension to developing a latent representation for aligned corpora involving several languages has been shown to scale to very large numbers of documents; similarly methods introducing sparsity into the representation provide a better match with the underlying statistics of text data while again scaling to very much larger datasets. The developed methods have been tested on standard corpora including the EuroPARL and CLEF GIRT datasets giving impressive results and demonstrating the scalability of the approaches.

Bibliography

[1] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2003.

[2] Z. Bai, Demmel J., J. Dongarra, A. Ruhe, and eds van der Vorst, H. *Templates for the solution of algebraic eigenvalue problems*. Society for Industrial and Applied Mathematics Philadelphia, 2000.

[3] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the Society for Information Science*, 41(6):391–407, 1990.

[4] Blaz Fortuna, Nello Cristianini, and John Shawe-Taylor. *Kernel methods in bioengineering, communications and image processing*, chapter A Kernel Canonical Correlation Analysis For Learning The Semantics Of Text, pages 263–282. Idea Group Publishing, 2006.

[5] Hotelling H. The most predictable criterion. *Journal of Educational Psychology*, 26:139–142, 1935.

[6] David R. Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[7] Dobsa J. *Dubinska Analiza Teksta Uporabom Konceptnog Indeksiranja*. PhD thesis, Sveuciliste u Zagrebu, Fakultet Elektrotehnike i Raucunarstva, 2006.

[8] J. R. Kettenring. Canonical analysis of several sets of variables. *Biometrika*, 58:433–451, 1971.

[9] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT Summit 2005*, 2005.

[10] Pei Ling Lai and Colin Fyfe. Kernel and nonlinear canonical correlation analysis. *International Joint Conference on Neural Networks*, 4:4614, 2000.

[11] Jure Leskovec, Marko Grobelnik, and Natasa Milic-Frayling. Learning semantic sub-graphs for document summarization. *Proceedings of the 7th International Multi-Conference Information Society*, B:18–25, 2004.

[12] M. Littman, S. Dumais, and T. Landauer. Automatic cross-language information retrieval using latent semantic indexing, 1998.

[13] Stéphane Mallat and Zhifeng Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

[14] J. Loren Watterson Moody T. Chu. On a multivariate eigenvalue problem, part i: Algebraic theory and a power method. *SIAM Journal on Scientific Computing*, Vol.14 NO.5:1089–1106, 1993.

[15] Tatsunori Mori, Tomoharu Kokubu, and Takashi Tanaka. Cross-lingual information retrieval based on lsi with multiple word spaces.

[16] Horst P. Relations among m sets of measures. *Psychometrika*, 26:129–149, 1961.

[17] G. Salton. Developments in Automatic Text Retrieval. *Science*, 253:974–980, August 1991.

[18] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[19] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, U.K., 2004.

[20] Alex J. Smola and Bernhard Schölkopf. Sparse greedy matrix approximation for machine learning. In *Proceedings of 17th International Conference on Machine Learning*, pages 911–918. Morgan Kaufmann, San Francisco, CA, 2000.

[21] Hirao Tsutomu, Kazawa Hideto, Isozaki Hideki, Maeda Eisaku, and Matsumoto Yuji. Machine learning approach to multi-document summarization. *Journal of Natural Language Processing*, 10:81–108, 2003.

[22] Pascal Vincent and Yoshua Bengio. Kernel matching pursuit. *Machine Learning*, 48:165–187, 2002.