MULTIMODAL RETRIEVAL WITH ASYMMETRICALLY WEIGHTED TRUNCATED-SVD CANONICAL CORRELATION ANALYSIS

Youssef Mroueh, Etienne Marcheret, Vaibhava Goel
Multimodal Algorithms and Engines Group
IBM T.J Watson Research Center, USA
{mroueh, etiennem, vgoel}@us.ibm.com

ABSTRACT

Joint modeling of language and vision has been drawing increasing interest. A multimodal data representation allowing for bidirectional retrieval of images by sentences and vice versa is a key aspect of this modeling. In this paper we show that canonical correlation analysis (CCA) can be adapted to bidirectional retrieval by a simple task dependent asymmetric weighting, which solves optimally the retrieval problem in a least squares sense. While regularizing CCA is known to improve numerical stability as well as generalization performance, less attention has been brought to the efficient computation of the regularization path of CCA, which is key to model selection. In this paper we develop efficient algorithms to compute the full regularization path of CCA within the classical Tikhonov and the truncated SVD (T-SVD CCA) regularization frameworks. T-SVD CCA is new to the best of our knowledge, and its regularization path can be computed more efficiently than its Tikhonov counterpart.

1 INTRODUCTION: MULTIMODAL RETRIEVAL

Modeling jointly language and vision has attracted a lot of attention recently. Generative models such as deep recurrent networks for the language modeling, in conjunction with deep convolutional neural networks on the image side have shown remarkable success in the image captioning task Karpathy & Li (2015); Mao et al. (2014); Vinyals et al. (2015); Socher et al. (2011). Image and Text retrieval has been the focus of many recent works. Fang et al. (2015); Klein et al. (2015); Lebret et al. (2015); Kiros et al. (2015; 2014); Karpathy et al. (2014); Gong et al. (2014); Socher et al. (2014); Kulkarni et al. (2011); Farhadi et al. (2010). Given an image and sentence embeddings the goal is to design a multimodal representation that captures the correlation between the two modalities, and allows for bidirectional search in a joint embedding space. In this paper we address the bidirectional retrieval problem and show that an asymmetric task dependent embedding of the images and sentences yields better performance on both retrieval tasks. Key to our representation is the careful analysis and usage of Canonical Correlation Analysis (CCA) in bidirectional retrieval, and the development of an efficient model selection for the regularized CCA embedding. We report our retrieval results on the COCO benchmark Lin et al. (2014).

1.1 THIS PAPER: CONTRIBUTIONS

The main contributions of this paper are:

1. Asymmetric and task dependent Correlation Weighting for CCA: We cast the bidirectional retrieval problem as a least squares problem and motivate the retrieval in the CCA weighted space. We theoretically and experimentally show that asymmetrically weighting the canonical weights with the canonical correlations in a task dependent way, solves each task optimally in the least squares sense (See Table 1). This improves the performance of each task and achieves state of the art on the COCO benchmark with off the shelf unsupervised features.
2. **Efficient Tikhonov/Truncated SVD Regularization Path for CCA**: While CCA is an old and widely studied problem that has been used as a multimodal data representation in many machine learning applications, we found that a computationally efficient cross-validation for regularized CCA has been less studied. Model selection is not only important to the numerical stability of CCA it also improves the generalization ability of this simple yet powerful multimodal representation. To this end we present two regularization algorithms for the batch CCA problem based on the Bjork Golub Algorithm Golub et al. (1973) (Algorithm 1), that make the computation of the full regularization path for CCA efficient. We present our cross validation algorithms within two regularization frameworks: Tikhonov Regularization (Algorithm 2) and truncated SVD regularization (Algorithm 3). While Tikhonov regularization is more popular for regularized CCA Vinod (1976), we also derive the regularization path for CCA using truncated SVD covariances. Truncated SVD is a popular regularizer for least squares regression problems Hansen (1986), but to our knowledge has not been utilized before in the CCA Context. We show that the truncated SVD CCA regularization path can be computed more efficiently and competes favorably with Tikhonov regularization.

**Notation.** Given a multimodal training set \( S = \{(x_i, y_i) | x_i \in \mathcal{X} \subset \mathbb{R}^{m_x}, y_i \in \mathcal{Y} \subset \mathbb{R}^{m_y}, i = 1 \ldots n\} \), \((n > \max(m_x, m_y))\), let \( X \in \mathbb{R}^{n \times m_x} \) and \( Y \in \mathbb{R}^{n \times m_y} \) be the two data matrices corresponding to each modality. Define \( \mu_X, \mu_Y \) to be the means of \( X \) and \( Y \) respectively. Let \( C_{XX} = (X - \mu_X)^\top (X - \mu_X) \in \mathbb{R}^{m_x \times m_x} \) and \( C_{YY} = (Y - \mu_Y)^\top (Y - \mu_Y) \in \mathbb{R}^{m_y \times m_y} \) be the covariances matrices of \( X \) and \( Y \) respectively. Let \( C_{XY} = (X - \mu_X)^\top (Y - \mu_Y) \in \mathbb{R}^{m_x \times m_y} \) be the correlation matrix. Define \( I_k \) to be the identity matrix in \( k \) dimensions. SVD stands for the **thin** singular value decomposition. A validation set \( S_v \) is given for model selection. Our goal is to index a test set \( S^* = \{(x_i^*, y_i^*) | x_i^* \in X^*, y_i^* \in Y^*, i = 1 \ldots n^*\} \) for bidirectional search. \( X \) and \( Y \) are assumed to be centered.

### 1.2 Bidirectional Retrieval as a Least Squares Problem

We start by defining more formally the bidirectional retrieval tasks. Given pairs of high dimensional points \((x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\) where \( x_i \) corresponds to the feature representation of an image given by a deep convolutional neural network, and \( y_i \) a sentence embedding of an associated caption. Our goal is to index this multimodal data in a way that enables bidirectional retrieval. Enabling the image annotation task which consists of retrieving for an image query \( x \in \mathcal{X} \) the associated caption \( y \in \mathcal{Y} \) and vice versa. Whereas the image search task consists of retrieving for a caption query \( y \in \mathcal{Y} \) the associated image \( x \in \mathcal{X} \).

In this Section we cast the retrieval tasks as a simple least squares problem. In order to reduce the variance of our estimators we put ourselves in the whitened space of the data in \( X \) and \( Y \), i.e we consider the bidirectional mapping between the whitened data spaces. Assuming for the moment that both covariances are non singular, we can write the whitened data as \( X C^{-\frac{1}{2}}_{XX} \) and \( Y C^{-\frac{1}{2}}_{YY} \). We will show later how Tikhonov/Truncated SVD regularization alleviates this issue of non singular covariances. Hence for the image search task we can minimize the following problem:

\[
\min_{T \in \mathbb{R}^{m_x \times m_y}} \left\| X C^{-\frac{1}{2}}_{XX} T - Y C^{-\frac{1}{2}}_{YY} \right\|_{F}^2, \tag{1}
\]

This defines a linear transform which maps the image set to the query caption. We follow this general principle of mapping the space of the search to the space of the query rather than the converse as found in Socher et al. (2013). The solution of Problem (1) is given simply by \( T = C^{-\frac{1}{2}}_{XX} C_{XY} C^{-\frac{1}{2}}_{YY} \), hence for a query test sentence \( y^* \) we can find its corresponding image by finding:

\[
\arg \min_{x^* \in S^*_x} \left\| C^{-\frac{1}{2}}_{YY} y^* - T^\top C^{-\frac{1}{2}}_{XX} x^* \right\|_{2}^2, \tag{2}
\]

We define the image to caption mapping as follows:

\[
f_{1 \rightarrow c}(x^*) = T^\top C^{-\frac{1}{2}}_{XX} x^*. \tag{3}
\]

The image search problem, then reduces to transforming the image set through the mapping \( f_{1 \rightarrow c}\), and then finding the nearest neighbor of the query caption represented by the whitened vector \( C^{-\frac{1}{2}}_{YY} y^* \) in the transformed image set.
Swapping the roles of $X$ and $Y$ we obtain equivalently that the image annotation problem can be solved also by a linear least squares. We define the caption to image mapping:

$$f_{c \to i}(y^*) = TC_{YY}^{-\frac{1}{2}} y^*,$$

(4)

therefore the image annotation problem reduces to transforming the caption set via the mapping $f_{c \to i}$, and then finding the nearest neighbor of the query image represented by the whitened vector $C_{XX}^{-\frac{1}{2}} x^*$ in the transformed caption set:

$$\arg \min_{y^* \in S_y} \left\| C_{XX}^{-\frac{1}{2}} x^* - TC_{YY}^{-\frac{1}{2}} y^* \right\|^2.$$

(5)

1.3 Simplifying the Expressions with Singular Value Decomposition

In this Section we simplify the expressions for image search and image annotation given in Equations (3), and (4) respectively, using the the singular value decompositions of $X$ and $Y$. Let $X = U_x \Sigma_x V_x^\top$ be the SVD of $X$ and $Y = U_y \Sigma_y V_y^\top$ be the SVD of $Y$. It is easy to show that the whitened data $0 = X V_x \Sigma_x^{-1} = U_x$, and $Y C_{YY}^{-\frac{1}{2}} = Y V_y \Sigma_y^{-1} = U_y$. For $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, let $u_x$ and $u_y$ be the whitened data points:

$$u_x = \Sigma_x^{-1} V_x^\top x, \quad u_y = \Sigma_y^{-1} V_y y.$$

(6)

Note that: $T = C_{XX}^{-\frac{1}{2}} C_{XY} C_{YY}^{-\frac{1}{2}} = \Sigma_x^{-1} V_x \Sigma_x^{-1} X^\top V_y \Sigma_y^{-1} = U_x^\top U_y$. $T$ corresponds to the correlation in the whitened spaces of $X$ and $Y$. Hence the expression in (3) for the image to caption mapping is simply:

$$f_{i \to c}(x^*) = T^\top u_x^* = \sum_{i=1}^n \langle u_{x^*}, u_{x_i} \rangle u_{y_i}.$$

(7)

The image search problem corresponds to finding for a query sentence $y^*$ the image $x^*$ solving:

$$\arg \min_{x^* \in S_x^*} \| u_{y^*} - f_{i \to c}(x^*) \|^2.$$

(8)

Similarly the expression in (4) for the caption to image mapping is simply:

$$f_{c \to i}(y^*) = T u_y^* = \sum_{i=1}^n \langle u_{y^*}, u_{y_i} \rangle u_{x_i}.$$

(9)

The image annotation problem corresponds to finding for a query image $x^*$ the caption $y^*$ solving:

$$\arg \min_{y^* \in S_y^*} \| u_{x^*} - f_{c \to i}(y^*) \|^2.$$

(10)

We see that the operator $T$ plays a central role in the retrieval problem. For an image $x$, $f_{i \to c}(x) = T^\top u_x = \sum_{i=1}^n \langle u_x, u_{x_i} \rangle u_{y_i}$, is our best guess in the least squares sense for a sentence in the whitened space of $Y$. For a caption $y$, $f_{c \to i}(y) = T u_y = \sum_{i=1}^n \langle u_y, u_{x_i} \rangle u_{x_i}$, is our best guess in the least squares sense for an image in the whitened space of $X$.

2 Retrieval: From Least Squares to CCA Gaining in Efficiency

2.1 Canonical Correlation Analysis

We review in this section Canonical Correlation Analysis due to Hotelling (1936). For data matrices $X \in \mathbb{R}^{n \times m_x}$ and $Y \in \mathbb{R}^{n \times m_y}$, let $k = \min(m_x, m_y)$ the canonical correlation $\sigma_1, \ldots, \sigma_k$, and their corresponding pairs of correlations weights $\{u_i, v_i\}_{i=1\ldots k}$, columns of $U \in \mathbb{R}^{m_x \times k}$, and $V \in \mathbb{R}^{m_y \times k}$ are the solution of the following maximization problem:

$$\max_{U^\top C_{XX} U = I_k, V^\top C_{YY} V = I_k} Trace(U^\top C_{XY} V),$$

where $\sigma_i = u_i^\top C_{XY} v_i, i = 1 \ldots k$. Intuitively CCA finds the directions that are maximally correlated and that are orthonormal in the metric defined by each covariance matrix, respectively. The following Lemma due to Bjork and Golub shows that the canonical correlation weights can be computed using the singular value decomposition of the data matrices $X$ and $Y$, and the correlation matrix in the whitened space $T$ defined in Section 1.3:
Lemma 1 (Golub et al. (1973)). Let $X = U_x \Sigma_x V_x^T$, and $Y = U_y \Sigma_y V_y^T$ be the singular value decomposition of $X$ and $Y$ ($U_x \in \mathbb{R}^{n \times m_x}, \Sigma_x \in \mathbb{R}^{m_x \times m_x}, V_x \in \mathbb{R}^{m_x \times m_x}$). Let $k = \min(m_x, m_y)$, and $T = U_x^T U_y$. Let

$$T = P_y \Sigma_y P_y^T$$

be its SVD, $P_x \in \mathbb{R}^{m_x \times k}, \Sigma \in \mathbb{R}^{k \times k}, P_y \in \mathbb{R}^{m_y \times k}$. The canonical correlations of $X$ and $Y$ are the diagonal elements of $\Sigma$, with canonical weights of $X$ given by $U = V_x \Sigma_x^{-1} P_x$, and the canonical weights of $Y$ given by $V = V_y \Sigma_y^{-1} P_y$. (Proof in Appendix A).

Algorithm 1 summarizes the Bjork Golub procedure to compute CCA. While the original algorithm of Bjork and Golub uses the QR factorization of $X$ and $Y$, we follow Avron et al. (2014) in exposing the algorithm fully with SVD. Note that both $U$ and $V$ correspond to a whitening step followed by a projection to a common space of dimension $k$. The total computational complexity of this algorithm assuming $m_y < m_x$ is $O(n m_y^2 + n m_x^2 + m_x m_y^2)$. While The Bjork Golub SVD algorithm is intuitive and efficient, it is less popular in machine learning than the generalized eigenvalue implementation of CCA.

Algorithm 1 Bjork Golub

1: procedure Bjork Golub Algorithm ($X \in \mathbb{R}^{n \times m_x}, Y \in \mathbb{R}^{n \times m_y}$)
2: $[U_x, \Sigma_x, V_x] \leftarrow \text{SVD}(X)$. 
3: $[U_y, \Sigma_y, V_y] \leftarrow \text{SVD}(Y)$. 
4: $T = U_x^T U_y \in \mathbb{R}^{m_x \times m_y}$
5: $[P_x, \Sigma, P_y] = \text{SVD}(T)$
6: $U = V_x \Sigma_x^{-1} P_x$
7: $V = V_y \Sigma_y^{-1} P_y$
8: return $U, V$
9: end procedure

2.2 Bidirectional Retrieval Using CCA Decomposition

As we have seen in Sections 1.3 and 2.1, the correlation operator $T$ plays a central role in the least squares formulation as well in the CCA formulation. Turning back to the image to caption mapping given in equation (7), using the expressions of the whitened data points in Equation (6) and the SVD of $T$ given in Equation (11), we obtain:

$$f_{i \leftarrow c}(x^*) = T^T u_x^* = P_y \Sigma P_x^T \Sigma_x^{-1} V_x^T x^* = P_y \Sigma U^T x^*,$$

where $U$ is the canonical weight for $X$ and $\Sigma$ the canonical correlation matrix. Note that $f_{i \rightarrow c}$ is still $m_y$ dimensional vector, we can reduce its dimension by projecting it to the $k$-dimensional column space of $P_x$. It follows that: $P_y^T f_{i \rightarrow c}(x^*) = \Sigma U^T x^* \in \mathbb{R}^k$. On the other hand: $P_y^T u_{y^*} = P_y^T \Sigma_y^{-1} V_y^T y^* = V^T y^* \in \mathbb{R}^k$, where $V$ is the canonical correlation weight for $Y$.

For the image search problem, we can therefore perform the nearest neighbor search given in equation (8), in the $k$-dimensional reduced space defined by

$$(P_y^T f_{i \leftarrow c}(x^*), P_y^T u_{y^*}) = (\Sigma U^T x^*, V^T y^*).$$

It follows that the image search problem can be written using the canonical weights $U, V$ and the canonical correlation $\Sigma$. Note that the correlation matrix is weighting the canonical weight in an asymmetric way. Cosine similarities are usually used for retrieval with CCA, hence we use cosine similarity between the two embeddings. The image search problem reduces to finding for a query caption $y^*$, the image $x^*$ solving:

$$\arg \max_{x^* \in S^2} \frac{\langle \Sigma U^T x^*, V^T y^* \rangle}{\| \Sigma U^T x^* \| \| V^T y^* \|}.$$  \hspace{1cm} (12)

Similarly the caption to image mapping can be expressed for a caption $y^*$ as: $f_{c \rightarrow i}(y^*) = P_x \Sigma V^T y^*$, and we can perform the search in the $k$-dimensional column space of $P_x$:

$$(P_x^T u_{x^*}, P_x^T f_{c \rightarrow i}(y^*)) = (U^T x^*, \Sigma V^T y^*).$$
Using a cosine similarity the image annotation problem reduces to finding for a query image $x^*$, the caption $y^*$ solving:

$$\arg\max_{y^* \in S^+} \frac{\langle U^T x^*, \Sigma V^T y^* \rangle}{\|U^T x^*\| \|\Sigma V^T y^*\|}.$$  \hspace{1cm} (13)

Note that using the canonical weights and correlations in the image search and annotation given in Equations (12) and (13), the search is done in a space of dimension $k = \min(m_x, m_y)$, rather than a space of dimensions $m_x$ for image search as in Equation (8) and $m_y$ for image annotation as in Equation (10). This results in a much more efficient search than the least squares formulation. Note that $\Sigma$ appears in an asymmetric way in the embedding of the points depending on the task. Hence we call our method asymmetrically weighted CCA. Asymmetrically weighted CCA (Table 1) performs approximately\(^1\) the optimal nearest neighbor search in the least square sense for each task.

Table 1: Task dependent embeddings: Asymmetrically Weighted CCA. $x^*$ is a test image, $y^*$ is a test caption. $(U, V)$ are the canonical weights of $X$ and $Y$, $\Sigma$ is a diagonal matrix with the canonical correlations of $X, Y$ on its diagonal. We use the cosine similarity for performing the search.

<table>
<thead>
<tr>
<th>Task</th>
<th>Image Embedding</th>
<th>Caption Embedding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Search</td>
<td>$\Sigma U^T x^*$</td>
<td>$V^T y^*$</td>
</tr>
<tr>
<td>Image Annotation</td>
<td>$U^T x^*$</td>
<td>$\Sigma V^T y^*$</td>
</tr>
</tbody>
</table>

3 CCA REGULARIZATION PATH: TIKHONOV VERSUS TRUNCATED SVD REGULARIZATION

While CCA is a powerful statistical tool less attention has been brought to the efficient computation of the full regularization path of CCA in a learning context where we need to do model selection on a validation set. For now we have assumed that the covariances $C_{XX}$ and $C_{YY}$ are non-singular, and we presented an SVD version of the Bjork Golub Algorithm in this context. Regularizing CCA does not only allow for numerical stability in the non-singular case, efficient model selection on a validation set allows for better generalization properties and avoids overfitting. Tikhonov regularization is the most common regularization used in CCA and consists in replacing the covariances by $C_{XX} + \gamma_x I_{m_x}$ and $C_{YY} + \gamma_y I_{m_y}$, where $\gamma_x, \gamma_y > 0$ are the regularization parameters subject to cross-validation. We specialize the Bjork Golub algorithm in Section 3.1 for handling efficiently a grid of parameters $\gamma_x, \gamma_y$.

One main contribution of this paper is in introducing the truncated SVD regularization to the covariances in the CCA problem. We emphasize that our truncation is applied to the SVD of the data matrices $X$ and $Y$, not the SVD of the whitened correlation matrix $T$ in the Bjork Golub Algorithm. Truncating the SVD of $T$ and choosing an embedding dimension $k < \min(m_x, m_y)$ is sometimes referred as the truncated SVD CCA, hence our clarification. We replace the covariance $C_{XX}$ by its best $k_x$-rank approximation given by the SVD of $X$, and $C_{YY}$ by its best $k_y$-rank approximation given by the SVD of $Y$, where $k_x, k_y \in \mathbb{N}$, $k_x \leq m_x$, and $k_y \leq m_y$. In this framework $k_x, k_y$ are the regularization parameters and are subject to cross-validation. We show in Section 3.2, how to specialize the Bjork Golub algorithm to handle efficiently a grid of parameters $k_x, k_y$ accounting for the regularization path of CCA. Proofs are given in Appendix A.

3.1 TIKHONOV REGULARIZATION: REGULARIZING THE COVARIANCES

The Regularized CCA problem can be written in this form:

$$\max_{U^T (C_{XX} + \gamma_x I_{m_x}) U = I, V^T (C_{YY} + \gamma_y I_{m_y}) V = I} \text{Tr}(U^T C_{XY} V)$$ \hspace{1cm} (14)

Algorithm 2 shows how to compute the regularization path of CCA in an efficient way. The computational complexity of the cross-validation is dominated by the computation of SVD of $T$ within

\(^1\) Using thin SVD $P_y$ is not a complete orthonormal basis of $\mathbb{R}^{m_y}$. 

---

5
the loop that is $C = \min(O(m_x m_y^2), O(m_y m_x^2))$. If we have $N_x$ and $N_y$ values of $\gamma_x$ and $\gamma_y$, the total complexity is $N_x N_y C$. Note that this can be fully parallelizable since each SVD computation is independent from the rest. $\sigma_{x,1}$ and $\sigma_{x,m_x}$ are minimum and maximum singular value of $X$.

**Algorithm 2** Tikhonov Regularized CCA

1: **procedure** TIKHONOV REGULARIZED CCA($X \in \mathbb{R}^{n \times m_x}, Y \in \mathbb{R}^{n \times m_y}$)
2: \[ [U_x, \Sigma_x, V_x] \leftarrow \text{SVD}(X). \]
3: \[ [U_y, \Sigma_y, V_y] \leftarrow \text{SVD}(Y). \]
4: \[ T_0 = \Sigma_x (U_x^T U_y) \Sigma_y \in \mathbb{R}^{m_x \times m_y} \]
5: **for** $\gamma_x \in \{\sigma_{x,1}^2, \ldots, \sigma_{x,m_x}^2\}$ **do**  \>
6: \>
7: \>
8: \>
9: \>
10: \>
11: \>
12: \>
13: **end for**
14: **end for**
15: **return** $U, V, \Sigma_x^{\gamma_x}, \gamma_x, \gamma_y$ with best validation performance for each task.
16: **end procedure**

**Lemma 2.** Algorithm 2 computes the regularization path of Tikhonov regularized CCA.

The values chosen for $\gamma_x, \gamma_y$ (Steps 5 and 6) are very important and are related to a technique called spectral filtering. \cite{Hansen1986}. A full discussion of the latter and its relation to truncated SVD is beyond the scope of this paper and will be subject to study in a forthcoming work.

### 3.2 Truncated-SVD Regularization: Truncating the Covariances

Let $k_x \leq m_x$ and let $X_{k_x}$ be the best $k_x$-rank approximation of $X$ given by the truncated SVD:

$$X_{k_x} = U_{k_x} \Sigma_{k_x} V_{k_x}^T, \quad U_{k_x} \in \mathbb{R}^{n \times k_x}, \Sigma_{k_x} \in \mathbb{R}^{k_x \times k_x}, V_{k_x} \in \mathbb{R}^{m_x \times k_x},$$

Similarly for $Y$, we define the best $k_y$-rank approximation ($k_y \leq m_y$):

$$Y_{k_y} = U_{k_y} \Sigma_{k_y} V_{k_y}^T, \quad U_{k_y} \in \mathbb{R}^{n \times k_y}, \Sigma_{k_y} \in \mathbb{R}^{k_y \times k_y}, V_{k_y} \in \mathbb{R}^{m_y \times k_y}$$

We define the truncated SVD CCA as follows:

$$\max_{U^T X_{k_x}^T X_{k_x} U = I, V^T Y_{k_y}^T Y_{k_y} V = I} Tr(U^T X^T Y V) \quad (15)$$

Algorithm 3, shows how to compute the regularization path of the truncated SVD CCA formulation in an efficient way. The computational complexity of the cross-validation is dominated by the computation of SVD of $T$ within the loop. While in Algorithm 2 for Tikhonov regularization $T$ is always a matrix of dimension $m_x \times m_y$. One advantage of the truncation in Algorithm 3 is that $T$ becomes a $k_x \times k_y$ dimension matrix, hence the SVD computation is more efficient and costs $\min(O(k_x k_y^2), O(k_y k_x^2))$. The total cost of the cross validation in Algorithm 3 is effectively more efficient than the one in Algorithm 2 (See Appendix B for timing experiments). Note that this also can be fully parallelizable since each SVD computation is independent from the rest. Note that one advantage of this truncated SVD for CCA is that the dimension of the embedding space $k = \min(k_x, k_y)$ is also automatically selected by the algorithm, while it is fixed in the Tikhonov case to $k = \min(m_x, m_y)$ (or is subject to another cross validation).
Algorithm 3: Truncated SVD -CCA

```plaintext
1: procedure CROSS VALIDATION TRUNCATED SVD -CCA (X ∈ ℜnxmx, Y ∈ ℜnymx)
2:   [Ux, Ξx, Vx] ← SVD(X).
3:   [Uy, Ξy, Vy] ← SVD(Y).
4:   T = UxT Uy ∈ ℜmxmx, mnxm, myxmy
5:   Wx = VxΣ−1x ∈ ℜmxmx
6:   Wy = VyΣ−1y ∈ ℜmyxmy
7: for kx ∈ [mx] do
8:   for ky ∈ [my] do
9:     [Pkx, Ξkx.kyx, Pky] = SVD(T1:kx,1:ky) ▷ T_{1:kx,1:ky} extracts the first kx rows and the first ky columns of T.
10:    U = Wx1:kx Pkx
11:    V = Wy1:ky Pky
13: (Bidirectional Retrieval is done using a sort list of the scores in Eqs (12), and (13.).)
14: end for
15: end for
16: return U, V, Ξkx.kyx, kx, ky with best validation performance for each task.
17: end procedure
```

Lemma 3. Algorithm 3 computes the regularization path of the truncated SVD regularized CCA.

4 Relation to Previous Work

Joint modeling of language and vision is a rapidly growing field, we focus here on some recent works that use bidirectional maps in the retrieval tasks and their relation to this paper. Klein et al. (2015), and Gong et al. (2014) used CCA to build a joint representation using the cosine similarity. In both works a symmetric weighting of the CCA canonical weights was used, i.e for an image caption pair (x∗, y∗), a joint embedding of the form (ΣαUT x∗, ΣαVT y∗), was used, where α = 0, in Klein et al. (2015) case and α > 0 in Gong et al. (2014) case. The symmetric weighting in Gong et al. (2014) was a heuristic found to improve performance and is not theoretically motivated as in the asymmetric weighting of this paper. Indeed our experimental results show that asymmetric weighting gives higher performance and makes the image search and the image annotation on par in term of performance. A discussion of the optimality of the asymmetric weighting and an empirical study is given in Appendix C.

Skip thought vectors introduced in Kiros et al. (2015) for representing sentences were used for bidirectional search in conjunction with VGG features Simonyan & Zisserman (2015) on the image side, with linear embeddings learned with a discriminative triplets loss, instead of a CCA loss. A similar discriminative loss was used for building a joint image and text representation in Lebret et al. (2015). One advantage of CCA over the discriminative losses is that the batch solution can be found efficiently and does not need stochastic optimization, as well as the efficient model selection introduced in this paper providing a means to select the dimension of the joint embedding leading to better generalization properties.

5 Numerical Results

We performed image annotation and search tasks on the COCO benchmark Lin et al. (2014) using our task dependent asymmetrically weighted CCA, as described in Table 1. Retrieval was performed using cosine similarities given in Equations (12) and (13). The training set contains 82783 images, along with 5 captions each. Similarly to Klein et al. (2015), we used the splits from Karpathy & Li (2015), and performed cross-validation on 5 validation splits of 5K images, and tested our models on 5 testing splits of 5K images each, as well as 25 splits of 1K images each. We report for both tasks the recall rate at one result, five results, or ten first results (r@1,5,10), as well as the median rank of the first ground truth retrieval. Cross validation was performed using the Asymmetrically Weighted Truncated SVD CCA (AWT-CCA) Algorithm 3 to select the model corresponding to the minimum median rank (regularization paths are given in Appendix D).
For feature extraction we follow Klein et al. (2015), and use on the image side the VGG CNN representation Simonyan & Zisserman (2015), where each image is cropped in 10 ways into 224 by 224 pixel images: the four corners, the center, and their x-axis mirror image. The mean intensity of each crop is subtracted in each color channel, and then encoded by VGG19 (the final FC -4096 layer). The average of the resulting 10 feature vectors corresponding to each crop is used as the image representation. On the text side, we use two sentence embeddings: the first one proposed by Klein et al. (2015) consisting in averaging the word2vec Mikolov et al. (2013) representation of words in each sentence. Word2vec representations are available on code.google.com/p/word2vec/, we followed the recommendation of Klein et al. (2015) in preprocessing the word2vec vocabulary matrix of COCO with PCA, to ensure averaging uncorrelated channels in the sentence embedding. The second embedding we used are skip thought vectors introduced in Kiros et al. (2015), which encodes sentences to vectors using an LSTM. Image and text features were centered before learning CCA.

<table>
<thead>
<tr>
<th>Test images:</th>
<th>Image search</th>
<th>Image annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r@1</td>
<td>r@5</td>
</tr>
<tr>
<td>1K test images:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRNN Karpathy &amp; Li (2015)</td>
<td>20.9</td>
<td>52.8</td>
</tr>
<tr>
<td>Mean Vec (vocabICA + CCA) Klein et al. (2015)</td>
<td>24.2</td>
<td>56.4</td>
</tr>
<tr>
<td>Mean Vec (vocabPCA+AWT-CCA)</td>
<td>26.68</td>
<td>59.59</td>
</tr>
<tr>
<td>Skip thoughts (AWT-CCA)</td>
<td>27.47</td>
<td>61.72</td>
</tr>
<tr>
<td>Skip thoughts + Triplets loss Kiros et al. (2015)</td>
<td>25.9</td>
<td>60</td>
</tr>
<tr>
<td>GMM+HGLMM+CCA Klein et al. (2015)</td>
<td>25.6</td>
<td>60.4</td>
</tr>
<tr>
<td>5K test images:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BRNN Karpathy &amp; Li (2015)</td>
<td>8.9</td>
<td>24.9</td>
</tr>
<tr>
<td>Mean Vec (vocabICA + CCA) Klein et al. (2015)</td>
<td>10.3</td>
<td>27.2</td>
</tr>
<tr>
<td>Mean Vec (vocabPCA+AWT-CCA)</td>
<td>11.63</td>
<td>30.35</td>
</tr>
<tr>
<td>Skip thoughts (AWT-CCA)</td>
<td>11.61</td>
<td>31.70</td>
</tr>
<tr>
<td>Skip thoughts + Triplets loss Kiros et al. (2015)</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>GMM+HGLMM + CCA Klein et al. (2015)</td>
<td>11.2</td>
<td>29.2</td>
</tr>
</tbody>
</table>

Table 2: Mean results of the test splits on the COCO benchmark (in %). Standard deviations are in the range (0.0059 – 0.0076). Numbers in bold refer to our method, not always the best.

We see in Table 2 that using off the shelf unsupervised features (VGG, word2vec, skip thoughts vectors), AWT-CCA achieves state of the art results at both image search and annotation, and competes with supervised features extraction such as GMM+HGLMM of Klein et al. (2015). More importantly the asymmetric weighting and the efficient model selection makes the performance of both tasks on par where we see an imbalance in the other methods.

6 Conclusion

In this paper we showed how an asymmetric weighting and a computationally efficient cross validation of CCA improves the performance of bidirectional retrieval tasks. While the exposition of this paper was on image and caption retrieval, our methods are generic and can be applied to any multimodal setting. We solved in this paper CCA in its batch formulation, for handling larger scale datasets we can use the randomized SVD of Halko et al. (2011) or the subsampled CCA of Avron et al. (2014). Stochastic Gradient Descent CCA Ma et al. (2015); Wang et al. (2015), SGD-CCA (Linear and Deep CCA) was proposed recently and can be applied to our problem. It would be interesting to study an efficient model selection for the SGD-CCA such as early stopping and truncated SVD for a finer model selection. A simple extension to kernel CCA regularization of this present paper would be to leverage random Fourier features for approximating Kernel CCA Lopez-Paz et al. (2014). Truncated SVD-CCA would be particularly interesting in this setting since $k_x$, $k_y$ would be balancing the regularization and the estimation error of the kernel by means of random features.

Note that the 1K results on Mean Vec (ICA + CCA) of Klein et al. (2015) and the skip thought of Kiros et al. (2015) are not comparable to ours since they are reported on only 1K test set, we report the mean of 25 splits 1K test sets.
REFERENCES


Kiros, Ryan, Salakhutdinov, Ruslan, and Zemel, Richard S. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, 2014.


Socher, Richard, Karpathy, Andrej, Le, Quoc V., Manning, Christopher D., and Ng, Andrew Y. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2014.


A Appendix: Proofs

Proof of Lemma 1. We give the proof here as it will be useful in the development of the full regularization path of CCA with truncated SVD regularization of the covariances $C_{XX}$ and $C_{YY}$. Let $P_x = \Sigma_x V_x^T U \in \mathbb{R}^{m_x \times k}$ equivalently $U = V_x \Sigma_x^{-1} P_x$ and $P_y = \Sigma_y V_y^T V \in \mathbb{R}^{m_y \times k}$ equivalently $V = V_y \Sigma_y^{-1} P_y$. Hence we obtain by this change of variable:

$$U^T X^T Y V = P_x^T \Sigma_x^{-1} V_x^T V_x \Sigma_x U_x^T U_y \Sigma_y V_y^T V_y \Sigma_y^{-1} P_y = P_x^T (U_x^T U_y) P_y.$$

Similarly: $U^T X^T U = P_x^T P_x$ $V^T Y^T YV = P_y^T P_y$. Hence replacing $U, V$ with $P_x, P_y$ we have:

$$p_x^T p_x = \max_{p_x, p_y} \text{Tr}(P_x^T (U_x^T U_y) P_y).$$

This is solved by an SVD of $T = U_x^T U_y, [P_x, \Sigma, P_y] = SV D(T), (P_x \in \mathbb{R}^{m_x \times k}, \Sigma \in \mathbb{R}^{k \times k}, P_y \in \mathbb{R}^{m_y \times k}),$ where $k = \min(m_x, m_y)$, and finally we have $U = V_x \Sigma_x^{-1} P_x, V = V_y \Sigma_y^{-1} P_y.$ \hfill \Box

Proof of Lemma 2. The proof of this Lemma is elementary we give it here for completion.

$$[U_x, \Sigma_x, V_x] = SV D(X) \quad U_x \in \mathbb{R}^{n \times m_x}, \Sigma_x \in \mathbb{R}^{m_x \times m_x}, V_x \in \mathbb{R}^{m_y \times m_x} \quad X = U_x \Sigma_x V_x^T.$$

$$[U_y, \Sigma_y, V_y] = SV D(Y) \quad U_y \in \mathbb{R}^{n \times m_y}, \Sigma_y \in \mathbb{R}^{m_y \times m_y}, V_y \in \mathbb{R}^{m_y \times m_y} \quad Y = U_y \Sigma_y V_y^T.$$

$$X^T X + \gamma_x I = V_x (\Sigma_x^2 + \gamma_x I) V_x^T.$$

$$Y^T Y + \gamma_y I = V_y (\Sigma_y^2 + \gamma_y I) V_y^T.$$

Let $P_x = \sqrt{\Sigma_x^2 + \gamma_x I} V_x U \in \mathbb{R}^{m_x \times k}$ equivalently $U = V_x (\Sigma_x^2 + \gamma_x I)^{-\frac{1}{2}} P_x$. Let $P_y = \sqrt{\Sigma_y^2 + \gamma_y I} V_y^T V \in \mathbb{R}^{m_y \times k}$ equivalently $V = V_y (\Sigma_y^2 + \gamma_y I)^{-\frac{1}{2}} P_y$. Hence we obtain by this change of variable:

$$U^T X^T Y V = P_x^T (\Sigma_x^2 + \gamma_x I)^{-\frac{1}{2}} V_x^T V_x \Sigma_x U_x^T U_y \Sigma_y V_y^T V_y (\Sigma_y^2 + \gamma_y I)^{-\frac{1}{2}} P_y = P_x^T (\Sigma_x^2 + \gamma_x I)^{-\frac{1}{2}} \Sigma_x (U_x^T U_y) \Sigma_y (\Sigma_y^2 + \gamma_y I)^{-\frac{1}{2}} P_y.$$

Let

$$T = (\Sigma_x^2 + \gamma_x I)^{-\frac{1}{2}} \Sigma_x (U_x^T U_y) \Sigma_y (\Sigma_y^2 + \gamma_y I)^{-\frac{1}{2}}.$$

Hence we obtain that:

$$[P_x, \Sigma, P_y] = SV D(T).$$

\hfill \Box

Proof of Lemma 3. Let

$$P_{k_x} = \Sigma_{k_x} V_{k_x}^T U \in \mathbb{R}^{k_x \times k}, \quad \text{equivalently } U = V_{k_x} \Sigma_{k_x}^{-1} P_{k_x} \in \mathbb{R}^{m_x \times k}, k = \min(k_x, k_y)$$

$$P_{k_y} = \Sigma_{k_y} V_{k_y}^T V \in \mathbb{R}^{k_y \times k}, \quad \text{equivalently } V = V_{k_y} \Sigma_{k_y}^{-1} P_{k_y} \in \mathbb{R}^{m_y \times k}, k = \min(k_x, k_y)$$

Hence we obtain by this change of variable:

$$U^T X^T Y V = P_{k_x} \Sigma_{k_x}^{-1} V_{k_x}^T V_x \Sigma_x U_x^T U_y \Sigma_y V_y^T V_y \Sigma_y^{-1} P_{k_y}.$$ 

Now we turn to:

$$\Sigma_{k_x}^{-1} (V_{k_x}^T V_x) \Sigma_x U_x^T = \Sigma_{k_x}^{-1} [I_{k_x \times k_x} 0_{k_x \times (m_x - k_x)}] \Sigma_x U_x^T$$

$$= \Sigma_{k_x}^{-1} [I_{k_x \times k_x} 0_{k_x \times (m_x - k_x)}] U_x^T$$

$$= [I_{k_x \times k_x} 0_{k_x \times (m_x - k_x)}] U_x^T$$

$$= U_{k_x}.$$ 

Hence we keep the first $k_x$ columns of $U_x$, that is $U_{k_x}$. The same argument hold for $U_{k_y}$. It follows that using truncated SVD, we have:

$$U^T X^T Y V = P_{k_x} U_{k_x}^T U_{k_y} P_{k_y}.$$
Hence truncated SVD-CCA can be solved finding:

\[ [P_{k_x}, \Sigma_{k_x}, P_{k_y}] = \text{SVD}(U_{k_x}^T U_{k_y}) \].

Turning now to \( U_{k_x}^T U_{k_y} \) this can be computed efficiently by precomputing \( T = U_{k_x}^T U_{k_y} \in \mathbb{R}^{m_x \times m_y} \) and then extracting the submatrix consisting of \( k_x \) rows and \( k_y \) columns. we return therefore \( U = V_{k_x} \Sigma_{k_x}^{-1} P_{k_x}, V = V_{k_y} \Sigma_{k_y}^{-1} P_{k_y} \).

\[ \Box \]

\section{Efficiency of T-SVD CCA versus Tikhonov CCA}

Table 3: Timing of the full regularization path: we report in this table the timing of computing the full regularization path of asymmetrically weighted CCA, within Truncated SVD and Tikohnov Regularization frameworks. The retrieval performance on the validation set (5K) is reported in the form image search/image annotation. The parameter grid for both cases is 20 \times 20. Experiments were conducted on a single Intel Xeon CPU E5-2667, 3.30GHz, with 265 GB of RAM and 25.6 MB of cache. We see that T-SVD is faster to compute and competes favorably with Tikhonov regularization. In the VGG/word2vec setup \( (m_x = 4096, m_y = 300) \) we obtain a 2x speedup factor. In the VGG/skip thoughts setup \( (m_x = 4096, m_y = 4800) \) we obtain a 6x speedup factor.

<table>
<thead>
<tr>
<th></th>
<th>r@1</th>
<th>r@5</th>
<th>r@10</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWT-SVD CCA (word2vec)</td>
<td>12.7/11.9</td>
<td>31.4/31.7</td>
<td>43.86/44.38</td>
<td>3763.654s</td>
</tr>
<tr>
<td>AW Tikhonov CCA (word2vec)</td>
<td>13.3/12.4</td>
<td>32.7/32.84</td>
<td>45.48/45.0</td>
<td>6981.697s</td>
</tr>
<tr>
<td>AWT-SVD CCA (Skip Thoughts)</td>
<td>12.5/12.98</td>
<td>32.4/32.9</td>
<td>44.96/45.46</td>
<td>24930.61s</td>
</tr>
<tr>
<td>AW Tikhonov CCA (Skip Thoughts)</td>
<td>12.50/12.44</td>
<td>33.1/32.54</td>
<td>45.24/45.22</td>
<td>163140.10s</td>
</tr>
</tbody>
</table>
C OPTIMALITY OF THE TASK DEPENDENT ASYMMETRIC WEIGHTING

![Graph showing R@10 versus α on the validation set for image annotation and image search](image)

Figure 1: Optimality of the asymmetric weighting: This plot shows R@10 versus α on the validation set, for image annotation and image search performed using the image embedding $\Sigma^\alpha U^\top$ and caption embedding $\Sigma^{1-\alpha}V^\top$, where $\alpha \in [0, 1]$. The cosine similarity was used between embeddings. In this experiment we used VGG image features and average word2vec sentence embedding. For each α we perform a thorough cross validation using truncated SVD CCA, and report the best R@10 for image annotation and image search. We see that $\alpha = 0$, i.e the embedding $(U^\top, \Sigma V^\top)$ is optimal for image annotation as predicted by the least squares formulation for image annotation in this paper. $\alpha = 1$, i.e the embedding $(\Sigma^\alpha U^\top, V^\top)$ is optimal for image search as predicted by the least squares formulation.

| $\Sigma^\alpha U^\top, \Sigma^\alpha V^\top$, $\alpha \geq 0$ (CCA) | Image search | | Image annotation |
|---|---|---|---|---|---|---|
| $\Sigma^\alpha U^\top, \Sigma^\alpha V^\top$, $\alpha = 0$ (CCA) | 8.86 25.14 35.96 | 9.23 25.05 35.43 |
| $\Sigma^\alpha U^\top, \Sigma^\alpha V^\top$, $\alpha = 1$ | **10.79 28.17 39.56** | **9.68 26.32 37.68** |
| $\Sigma^\alpha U^\top, \Sigma^\alpha V^\top$, $\alpha = 2$ | 9.74 26.01 36.95 | 8.23 22.75 33.30 |
| $\Sigma^\alpha U^\top, \Sigma^\alpha V^\top$, $\alpha = 3$ | 8.21 23.15 33.46 | 6.84 19.63 29.60 |
| $\Sigma^\alpha U^\top, \Sigma^\alpha V^\top$, $\alpha = 4$ | 7.00 20.51 30.36 | 5.88 17.44 26.64 |
| $\Sigma^\alpha U^\top, \Sigma^\alpha V^\top$, $\alpha = 5$ | 6.06 18.27 27.58 | 5.07 15.69 24.26 |
| $\Sigma^\alpha U^\top, \Sigma^\alpha V^\top$, $\alpha = 6$ | 5.39 16.42 25.21 | 4.45 14.14 22.14 |

Table 4: Comparison to Symmetric Weighting Heuristic of Gong et al. (2014): In the VGG/Word2vec setup a thorough cross validation using T-SVD CCA for embeddings of the form $(\Sigma^\alpha U^\top, \Sigma^\alpha V^\top)$, $\alpha \geq 0$, used jointly with the cosine similarity for retrieval tasks r@1,5, and 10. Results are reported here on the validation set. $\alpha = 0$ corresponds to using the CCA weights as in Klein et al. (2015). We see that this symmetric weighting boosts the performance for a particular $\alpha = 1$ as reported in Gong et al. (2014). The asymmetric weighting we propose outperforms the symmetric weighting heuristic as shown in this table, and boosts performance of both tasks.

D AWT-SVD CCA REGULARIZATION PATHS
Figure 2: Regularization Path for T-SVD CCA on bidirectional retrieval on the COCO Benchmark with VGG features (4096 dimensions) for the image and average word2vec for captions (300 dimensions). Cross validation was performed on a validation set of size 5K, on grid going from 400 to 4000 with step size of 100 for $k_x$, and from 50 to 300 with a step size of 5 for $k_y$. We report the average median rank of the retrieved query over five validation splits (lower is better, in blue).

Figure 3: Regularization Path for T-SVD CCA on bidirectional retrieval on the COCO Benchmark with VGG features (4096 dimensions) for the image and skip thought vectors for captions (4800 dimensions). Cross validation was performed on a validation set of size 5K, on a grid going from 400 to 4000 with a step size of 100 for $k_x$ and from 400 to 4800 with a step size of 100 for $k_y$. We report the average median rank of the retrieved query over five validation splits (lower is better, in blue).