Pattern Recognition Letters 98 (2017) 110–116

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Query specific re-ranking for improved cross-modal retrieval

Devraj Mandal, Soma Biswas*

Department of Electrical Engineering, Indian Institute of Science, Bangalore, Karnataka, 560012, India

ARTICLE INFO

Article history:
Received 2 March 2017
Available online 6 September 2017

MSC:
41A05
41A10
65D05
65D17

Keywords:
Cross-modal
Image-text
Re-rank

ABSTRACT

Cross-modal retrieval tasks like image-to-text, audio-to-image retrieval, etc. are an important area of research. Different algorithms have been developed to address these tasks. In this work, we propose a novel query specific re-ranking based approach to improve the retrieval performance of any given baseline approach. For each query, the top K-retrieved results of the baseline algorithm are used to compute its class-rank order feature. Based on this feature of the query and the highly relevant examples within the top K-retrieved results, each training example is given a score indicating its relevance to the query, which is finally used to train the query-specific regressor. The new score given by this regressor to each retrieved example is then used to re-rank them. The proposed approach does not require knowledge of the baseline algorithm, and also does not extract additional features from the data. Thus it can be used as an add-on to any existing algorithm for improved retrieval performance. Experiments with several state-of-the-art cross-modal algorithms across different datasets show the effectiveness of the proposed re-ranking algorithm.

1. Introduction

Cross-modal retrieval is an important area of research in the field of computer vision and pattern recognition with a wide range of applications. For example, given a text query, we may want to retrieve semantically meaningful images from the database. A few examples of cross-modal data matching considered in this work are shown in Fig. 1. Several approaches have been proposed in the literature to address this task [1–3].

In this work, we propose a novel query specific re-ranking approach for improving the retrieval results of any baseline algorithm. The input to the algorithm is the retrieval results of the baseline approach along with their similarity (or distance) scores and also the training data used by the baseline approach. The majority of the cross-modal approaches aim to find the relation between the different modalities. In this work, we ask the question: *can we use the relative positioning of the query and retrieved data with respect to its own modality to improve the retrieved results?* For each query, first a class-rank order feature is computed based on the top K-retrieved results of the baseline algorithm. Based on this new feature, a subset of the top K-retrieved results, termed as the highly relevant set is chosen. This is based on our confidence as to which of them actually belong to the same class as the query. These are used to compute a score for each training example, which is then used to train the query-specific regressor. The new score given by this regressor to each retrieved data is finally used to re-rank them.

Extensive experimental evaluation is performed on several baseline algorithms and on different datasets to justify the usefulness of the proposed re-ranking approach. Specifically, for the baseline algorithms, we use a variety of classical approaches like Canonical Correlation Analysis (CCA) [1,2], dictionary learning approaches like Coupled Dictionary Learning (CDL) [4], deep learning based approaches like Deep CCA with Auto-encoders (DCCA-AE) [5] and so on. The approach has been tested on a variety of multi-modal databases like Wikipedia [6], LabelMe [7], Materials [8] and Multiple Features [9] and significant improvements over the Mean Average Precision (MAP) and rank-1 accuracy has been observed. The contributions of this work are as follows:

1. We propose a novel query-specific re-ranking framework, which is able to improve the baseline retrieval results for cross-modal tasks and can be used as an add-on to any existing approach.
2. The proposed approach does not generate new features, and also does not require knowledge of the inner workings of the baseline method.
3. The approach is able to improve the baseline results of various approaches over a variety of different datasets involving a wide variety of cross-modal retrieval tasks.

* Corresponding author.
E-mail addresses: devraj89@ee.iisc.ernet.in (D. Mandal), soma.biswas@ee.iisc.ernet.in (S. Biswas).

http://dx.doi.org/10.1016/j.patrec.2017.09.008
0167-8655/

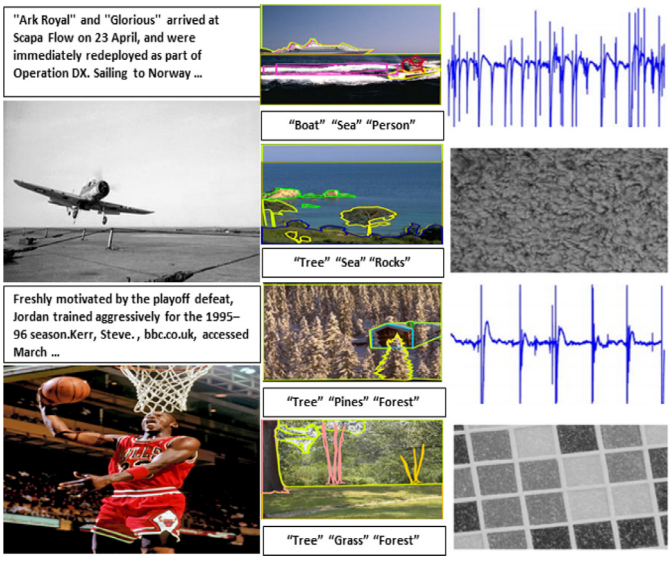


Fig. 1. A few examples of cross-modal data matching. First two columns: image-text modalities (Wikipedia and LabelMe datasets); Third column: audio signatures and corresponding images of texture surfaces (Materials dataset).

The rest of the paper is organized as follows. Section 2 discusses the related works. Details of the proposed approach are described in Section 3. The experimental results are given in Section 4 and the paper concludes with a brief discussion.

2. Related works

Here we discuss some of the related works in the literature on cross-modal matching as well as re-ranking.

Cross-modal approaches: First, we discuss the relevant literature for cross-modal matching. Given paired data of two modalities, Canonical Correlation Analysis (CCA) [1,2] learns a lower dimensional feature space from the two modalities. To handle the non-linear relationship between the data, kernel trick has been employed to devise Kernel CCA [1,2]. The constraint of paired data has been removed in the formulation of mean CCA, cluster CCA [10] and their kernelized versions. Generalized Multiview Analysis (GMA) [3] mathematically formulates the cross-modal analysis problem as a constrained quadratic program and provides a solution by generalized eigenvalue approach. GMA is shown to be a supervised extension of CCA and an extension towards its kernel form has also been designed. Scalable variants of CCA has been developed by using randomness to capture the intrinsic non-linear relationship between data from two modalities [11].

As an alternative to the non-parametric KCCA, Deep Canonical Correlation Analysis (DCCA) [12] uses neural networks to learn complex non-linear transformations of the two views of data such that the resulting representations are highly linearly correlated. Auto-encoders combined with a deeper representation of CCA in a jointly optimized framework has also been devised for cross-modal retrieval tasks [5]. Relatively Paired Space Analysis (RPSA) [13] uses the relative pairing of information to build a discriminative latent model while solving a maximum margin problem. Coupled Dictionary based Learning Methods (CDL) [4] learns two projections over the sparse representation domain to learn a common subspace for cross-modal matching.

Re-ranking approaches: Now, we will discuss the relevant literature on re-ranking. In [14], a novel clustering algorithm for tagging a face dataset has been provided. The rank-order distance is motivated by the observation that faces of the same person usually share their top neighbors. The rank-order distance has been extended for solving person re-identification problems [15]. This problem has also been solved by using soft biometric attributes [16], appearance attribute subspaces [17], learning intra-camera discriminative models [18], and even bi-directional ranking methods [19] involving both the probe and gallery data as the query. Deep metric learning [20] involving a siamese deep neural network is used to learn the color feature, texture feature and a metric jointly for practical person re-identification. Visual saliency and consistency have been used to design a re-ranking algorithm in [21]. Contextual spaces aiming to explore relationships between images have also been used for re-ranking [22].

A few more re-ranking approaches proposed in the literature can be found in [23–26]. These query based adaptive re-ranking methods usually generate positive and negative image pairs for each query to re-rank the results. Though these methods show impressive performance in the person re-identification problem, they are mainly suitable for image-image matching and it is not clear how they can be extended for the cross-modal scenario. The POP algorithm in [24] uses a user-based interactive "one-shot" approach to identify positive and negative image examples to improve re-identification. The work in [26] proposes an iterative approach in a dictionary learning framework to handle re-ranking problems. In contrast, the proposed framework works with the original features and thus can be used with any existing algorithm provided the training data and the original distance scores are available. The re-ranking concept is slightly different from fine-tuning a deep model, where the model is usually first trained on a large auxiliary dataset and then fine-tuned to adapt to the target dataset. For this work, there is just one training dataset, and the proposed algorithm utilizes the same features that is used by the baseline algorithm.

Now, we will briefly describe some popular regression techniques which are used in our work [27]. Regression is a statistical process for estimating the relationships between a dependent variable and one or more independent variables. Linear regression [27] expresses the dependent variable using linear functions where the model parameters are estimated by studying the provided sample data. Linear regression (LR) has a simple closed form solution with techniques to use different regularization constraints to prevent over-fitting. Support Vector Regression (SVR) [27] finds a function which is bounded by a particular deviation from the target variables. In addition, SVR gives the option to embed the input data into high dimension using kernels and hence exploits the concept of non-linearity to achieve better performance in general.

3. Proposed approach

In this section, we describe in detail the proposed query-specific re-ranking algorithm.

3.1. Problem definition

Let the two modalities be denoted by $\mathbf{X}$ and $\mathbf{Y}$ and let the training data for the two modalities used by the baseline algorithm be denoted by $\mathbf{X}_{tr}$ and $\mathbf{Y}_{tr}$ respectively. Let the labels of the training data be denoted as $\mathbf{X}^L_{tr}$ and $\mathbf{Y}^L_{tr}$, where $L = \{1, 2, \ldots, C\}$, with C being the total number of classes. For a given query X_q and a baseline algorithm, let $\mathbf{Y}_q = \{Y_1, Y_2, \ldots, Y_N\}$ be the retrieved results with distances $\{d_1, d_2, \ldots, d_N\}$ from the query, where, $d_{i+1} \geq d_i$, for $i = \{1, 2, \ldots, N\}$. Given this information, the goal is to re-rank the retrieved data $Y_1, Y_2, \ldots, Y_N$ for improved retrieval performance as compared to the baseline algorithm, without any other information regarding the actual principles of the baseline algorithm.

3.2. Query specific-re-ranking algorithm

In this work, we utilize same modality matching to improve and complement the performance of cross-modality matching. First, the K-nearest neighbors from the retrieved results for the query are selected and the class rank-order feature which gives their class memberships are computed. Though we assume that majority of the K-nearest neighbors belong to the same class as the query, some of them may be incorrect, which can adversely affect the re-ranking performance. So using the class rank-order, we select a subset from the K-nearest neighbors, which are more likely to be correct, and term them as highly relevant data. Using these, the query specific regression function is learned which is finally used to re-rank the original retrieved data. Thus, the proposed approach has four main steps: (1) Computation of the class rank-order; (2) Computation of highly relevant data; (3) Learning the query-specific regression function and (4) Re-ranking of the retrieved results.

Computation of class rank-order: For query X_q, as in [14], we assume that majority of the first K-retrieved results $Y_1, Y_2, \ldots, Y_K$ with $K < N$ are correct, which means that they belong to the same class as the query.

First, we compute the class rank-order for the query and for each of the first K-retrieved results. Class rank-order indicates the class membership ranks of a data item which is measured using the relative position of the data with respect to data samples from its own modality. Consider $\{\mu_x^1, \mu_x^2, \ldots, \mu_x^C\}$ and $\{\mu_y^1, \mu_y^2, \ldots, \mu_y^C\}$ to be the set of mean vectors of the different classes for modalities **X** and **Y** computed from the training data. The relative position of the query with respect to these means is computed using

$$d_q^i = (X_q - \mu_x^i)^T \mathbf{M}_X (X_q - \mu_x^i) \qquad \{i = 1, 2, \ldots, C\} \tag{1}$$

where, $\mathbf{M}_X$ is the Mahalanobis metric learned using the training set to ensure better separation of the data in its own modality [28]. Class rank-order can be computed for the query by finding the order (position) of the classes after the distances are sorted in ascending order given as

$$R_q = Order(Sort(d_q^i)) \quad \forall i \tag{2}$$

where, *Sort* operation arranges the distances in an ascending manner and *Order* returns the order of the classes. For example, if the normalized distances of the query from the different means are (0.25, 0.2, 0.4, 0.1, 0.05) (assuming $C = 5$), then $R_q = (5, 4, 2, 1, 3)$. Similarly, we compute class rank-order corresponding to each of the first K-retrieved results using $\mathbf{M}_Y$.

Computation of highly relevant data: Let R_q and $\bar{R_q}$ denote the original and improved class rank-order of the query. We explain the computation of the improved class-rank order using the example in Fig. 2. We count the position of occurrence of each class in the R_k's and weigh them by using a monotonically decreasing function (inset). First, the relative importance of each class $rel(c)$ where $c = \{1, 2, \ldots C\}$ is computed as: $rel(c) = \sum_{t=1}^{C} w(t) * f_c(t) \ \ \forall c$. Here, $w(t)$ is the weighting function value for position t and $f_c(t)$ denotes the number of times the class $\{c\}$ occurs at the position $\{t\}$. Higher relative importance of a class implies higher confidence that the query belongs to that particular class. The order of the final counts (in a descending manner) then basically determines $\bar{R_q}$. In the example, we observe that since class {3} occurs at a higher position in the class rank-order of all the K-nearest neighbours, its place is moved up in $\bar{R_q}$ (the opposite happens for class {5}).

This improved $\bar{R_q}$ is then used to compute the highly relevant data $\mathcal{R}$, which is used to denote the subset of the retrieved results which are likely to be correct. We use a simple, but effective strategy to compute $\mathcal{R}$. $\mathcal{R} = \{k \mid ||\bar{R_q} - R_k||_h \leq T \ \ \forall k \in K\}$, where, T is a threshold set by cross validation and $||v||_h$ defines the number of non-zero entries of vector v. Note that the retrieved data whose distance is greater than T may also be correct, and thus they cannot be treated as negative samples for the query. We thus learn a regressor using $\mathcal{R}$ as the positive examples.

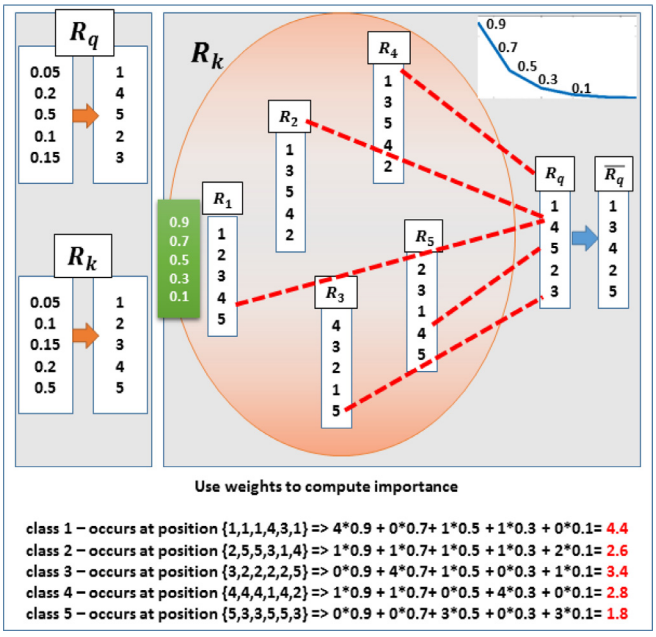


Fig. 2. Computation of improved class rank-order. Computation of the relative importance of each class is also shown at the bottom.

Learning the query-specific regression function: Next, we utilize the available training data and the highly relevant data to compute a regression function which will be used to re-rank the retrieved results. It may seem surprising how the same training data that was used by the baseline algorithm can be re-used for improving the retrieval results. The main intuition is that most of the cross-modal algorithms are based on learning the relation between the two modalities. In this work, we are complementing these approaches with the additional information that is present within the same modality, thus the same training data can be utilized in a complementary manner to improve the ranking results.

Let us assume that the query is from modality **X** and the retrieved items are from modality **Y**. For this setting, we learn a regressor using the training samples $\mathbf{Y}_{tr}$. Each training data in $\mathbf{Y}_{tr}$ is given a score which is a weighted combination of two distances L_1 and L_2 as given by

$$L_{tr} = w_1 \times L_1 + w_2 \times L_2 \tag{3}$$

where, w_1 and w_2 are set by performing cross validation experiments. The two scores L_1 and L_2 are given as follows:

1. L_1 indicates how similar the training sample is to the query, which is evaluated using the class-rank order of the query. The position of the true class of the training sample in $\bar{R_q}$ is given by the *Pos* operator. If the true class of the training sample has a higher rank in $\bar{R_q}$, it implies that there is higher confidence that the sample will belong to the same class as the query, which gives a lower value of the distance L_1. We define $L_1 = Pos(\bar{R_q})^2$.
2. L_2 captures the semantic match within its own modality. Assuming that the highly relevant items $\mathcal{R}$ are correct, we would like to give smaller distances to training examples which closely resemble the items in $\mathcal{R}$. We define $L_2 = \min_{r \in \mathcal{R}} d(Y_{tr}^i, r)$, i.e., which returns the smallest distance with all the highly relevant items. The more similar Y_{tr}^i is to the set $\mathcal{R}$, the smaller will be L_2.

Using these distances of the training examples, we learn a regression function to predict the new distances of each retrieved item. Though any suitable regression function can be used, here we use two widely used functions, namely the linear regression and support vector regression.

1. **Linear Regression (LR)**: Given the training samples $\{Y_{tr}^i, L_{tr}^i\}_{i=1}^n$, where Y_{tr}^i denotes the ith training sample and L_{tr}^i is the corresponding score, we find a linear data-fitting model $L_{tr}^i(Y_{tr}^i, W) = W^T Y_{tr}^i$ parameterized by W. Using matrix notations, we denote $\mathbf{Y}_{tr} = [Y_{tr}^1, \ldots, Y_{tr}^n]^T$ and $\mathbf{L}_{tr} = [L_{tr}^1, \ldots, L_{tr}^n]^T$. Then W can be learned using the closed form solution $W = (\mathbf{Y}_{tr}^T \mathbf{Y}_{tr})^{-1} \mathbf{Y}_{tr}^T \mathbf{L}_{tr}$.
2. **Support Vector Regression (SVR)**: For learning the model parameters W and b in support vector machine [27], the following problem needs to be solved

$$\begin{aligned} \min_{W,b} \quad & \frac{1}{2}||W||_F^2 \\ s.t. \quad & L_{tr}^i - W^T Y_{tr}^i - b \leq \epsilon \\ & W^T Y_{tr}^i + b - L_{tr}^i \leq \epsilon \end{aligned} \tag{4}$$

where ϵ is a free parameter which serves as a threshold. The idea is that all the predictions must lie within a ϵ-tube of the true values. As it is a convex optimization problem, it is solved using the primal-dual technique [27].

Re-rank the retrieved results : The learned regression function W is used to compute a new distance d^{reg} for each of the retrieved data items in $\mathbf{Y}_q = [Y_1, Y_2, \ldots Y_N]$ given by

$$d_i^{reg} = W^T Y_i \quad \forall i \in \{1, .., N\} \tag{5}$$

The final distance d_i^f of the retrieved data is a weighted combination of the original distance d_i^b given by the baseline algorithm and the new distance d_i^{reg} computed using the query-specific regression function:

$$d_i^f = \alpha d_i^b + (1 - \alpha) d_i^{reg} \quad \forall i \in \{1, .., N\} \tag{6}$$

where, the weight α can be chosen to weigh the contributions from the original and the re-ranking algorithm. The main steps of the proposed algorithm are given in Algorithm 1.

Algorithm 1 Proposed query-specific re-ranking algorithm.

1: **Input** : Training data $\mathbf{X}_{tr}$, $\mathbf{Y}_{tr}$, Labels $\mathbf{X}_{tr}^L$, $\mathbf{Y}_{tr}^L$, query X_q, retrieved data $\mathbf{Y}_q$ and baseline distances d^b.
2: **Output** : Re-ranked retrieved data
3: **Step 1: Compute class rank-order**
- Get the K-nearest neighbours $\mathbf{Y}_K$
- Compute class means μ_X, μ_Y and metric $\mathbf{M}_X, \mathbf{M}_Y$.
- Compute R_q and $R_k \ \forall k \in K$.

4: **Step 2: Compute highly relevant data**
- Compute the updated class-membership $\bar{R}_q$.
- Compute highly relevant set $\mathcal{R}$.

5: **Step 3: Learn the query-specific regression function**
- Compute L_1 and L_2 for all elements in $\mathbf{Y}_{tr}$.
- Use (3) to get $\mathbf{L}_{tr}$.
- Learn W by using **LR** or **SVR**.

6: **Step 4: Re-rank the retrieved results**
- Compute d^{reg} using learnt regression function using (5).
- Find the final distance d^f using (6).
- Re-rank all the elements in $\mathbf{Y}_q$ using this new distance.

Dataset	Description	Features	Train : Test Split	Protocol
LabelMe	2688 outdoor scene images from 8 different categories with their label tags.	Images – GIST features. Text – Word Frequency vector.	Train – 200 examples from each category Test – Remaining samples.	MAP
Full Wikipedia	2866 Text and Image data pairs from 10 categories.	Images – 128-d dense SIFT bag of visual words. Text – 10-topic LDA model.	Train : Test = 2173 : 693	MAP
Reduced Wikipedia	A reduced corpus consisting of 100 examples each from 5 categories of the Full Wikipedia dataset.	Same as Full Wikipedia.	Train : Test = 333 : 167	MAP
Materials	Images and their corresponding audio signatures for 17 different types of material surfaces. Data has 130 audio signatures and 3200 images.	Audio data is encoded using cepstral features and then clustered to get a bag-of-words representation. Text data is encoded using local binary patterns.	Features in the audio domain are duplicated to form one-to-one pairing. Random 50:50 split per class is done to generate train : test splits.	MAP
Multiple Features	Images of digits (10 categories with each having 200 samples).	Image – six different features.	Train : Test = 2:1 split of the given samples.	Rank 1

Fig. 3. Details of the datasets used in our experimental evaluation.

Table 1
Results (MAP) for the different algorithms on the LabelMe dataset along with the corresponding re-ranked results using Linear Regression (LR) and Support Vector Regression (SVR).

	Image-Text			Text-Image		
	Baseline	LR	SVR	Baseline	LR	SVR
CCA	0.592	0.602	0.619	0.589	0.638	0.653
GMA	0.630	0.627	0.635	0.616	0.649	0.656
CCCA	0.612	0.614	0.620	0.600	0.624	0.638
DCCA	0.617	0.629	0.641	0.605	0.649	0.660
RPSA	0.549	0.560	0.563	0.568	0.607	0.635
CDL	0.387	0.480	0.512	0.397	0.576	0.584
R-KCCA	0.644	0.652	0.655	0.629	0.645	0.667
DCCA-AE	0.609	0.613	0.617	0.606	0.638	0.657

4. Experimental results

Here we present the results of extensive evaluation of the proposed re-ranking algorithm on several baseline methods and on different datasets using both linear regression and support vector regression. For ease of understanding, we present the details of all the datasets, the features used, the train:test split and the evaluation protocol in Fig. 3. For evaluation, we use two standard measures, namely mean average precision (MAP) and Rank 1 recognition accuracy. MAP is a measure of whether the retrieved data belongs to the same class (relevant) or does not belong to the same class (irrelevant). Rank 1 accuracy measures whether the first retrieved item in response to a query is correct or not.

4.1. Results on LabelMe dataset

The LabelMe database [7] consists of image-text pairs from eight different categories (Fig. 3). The LabelMe Toolbox is used to generate the feature vectors.

The results of the baseline methods and the re-ranked results using the proposed algorithm are given in Table 1. Among the methods considered, CCA, DCCA, CDL, R-KCCA, DCCA-AE are unsupervised methods, whereas CCCA, GMA and RPSA are supervised ones. We observe that the proposed re-ranking algorithm is able to improve the results of both the unsupervised as well as the supervised methods.

Fig. 4 shows the top-10 retrieval results using CCA and the proposed re-ranking algorithm for the two textual queries, *forest* and *tall building*. The images in bounding boxes indicate the incorrectly retrieved results. We observe that for both the queries, the incorrect results are pushed backwards in the list of retrieved results.

(a) Query- *forest*

(b) Query - *tall building*

Fig. 4. Top-10 retrieved results using CCA (top row) and the proposed re-ranking algorithm (bottom row) for two queries for the LabelMe dataset. We observe that the incorrect results (shown by bounding boxes) are pushed backwards in the list of retrieved results.

Table 2
Results (MAP) for the different algorithms on the Full Wikipedia dataset along with the corresponding re-ranked results using Linear Regression (LR) and Support Vector Regression (SVR).

	Image-Text			Text-Image		
	Baseline	LR	SVR	Baseline	LR	SVR
CCA	0.252	0.282	0.283	0.198	0.209	0.210
GMA	0.253	0.269	0.268	0.180	0.197	0.200
CCCA	0.284	0.297	0.299	0.222	0.224	0.224
DCCA	0.258	0.278	0.279	0.202	0.212	0.211
RPSA	0.252	0.265	0.267	0.190	0.199	0.197
CDL	0.163	0.237	0.236	0.137	0.184	0.182
R-KCCA	0.225	0.267	0.270	0.177	0.200	0.197
DCCA-AE	0.263	0.289	0.291	0.206	0.216	0.215

Table 3
Results (MAP) for the different algorithms on the Reduced Wikipedia dataset along with the corresponding re-ranked results using Linear Regression (LR) and Support Vector Regression (SVR).

	Image-Text			Text-Image		
	Baseline	LR	SVR	Baseline	LR	SVR
CCA	0.462	0.560	0.548	0.393	0.478	0.522
GMA	0.575	0.595	0.601	0.432	0.468	0.534
CCCA	0.595	0.628	0.628	0.528	0.533	0.566
DCCA	0.569	0.644	0.641	0.556	0.575	0.606
RPSA	0.584	0.617	0.622	0.523	0.529	0.566
CDL	0.454	0.561	0.560	0.390	0.472	0.533
R-KCCA	0.552	0.603	0.602	0.500	0.515	0.552
DCCA-AE	0.549	0.607	0.605	0.504	0.522	0.566

4.2. Results on Wikipedia dataset

The results for the **Full Wikipedia dataset** [6] for all the baseline algorithms and the proposed re-ranking algorithm are given in Table 2. We observe that the proposed algorithm is able to significantly improve the results for all the baseline algorithms.

We also evaluate the proposed algorithm on the **Reduced Wikipedia dataset** [29]. The dataset details are given in Fig. 3. The results are shown in Table 3 where we observe significant gains over all the baseline methods.

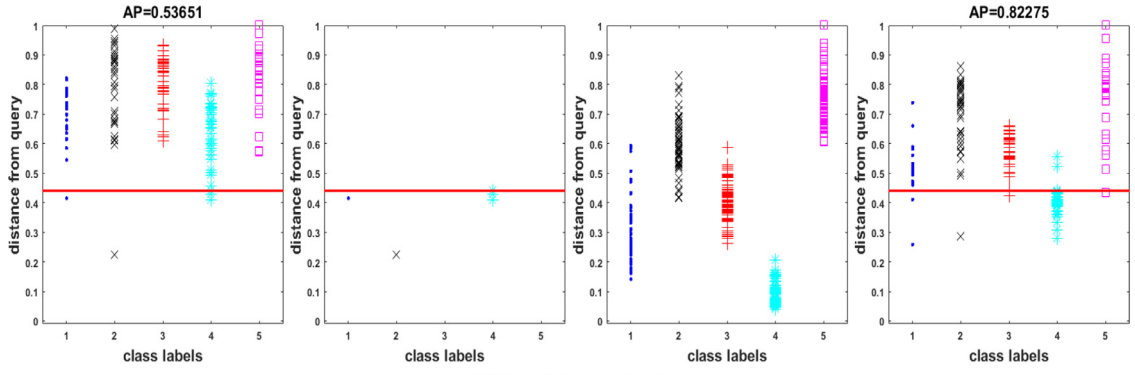
An illustration of the proposed algorithm is shown in Fig. 5 for two different examples from the Reduced Wikipedia dataset. In the figures, the x-axis denotes the class labels and the y-axis denotes the distance from the probe. The red horizontal line is the threshold value for selecting the K-nearest neighbours. The first column shows the original distances of the different retrieved data of all the classes from the query, from which the K-nearest neighbours are selected (second column). The correct class label of the query is given in the caption. The third column shows the distances given to the training data for regression. We observe that as desired, the training data corresponding to the correct query class (class 4 and 5 for the top and bottom examples) has lower distances. Finally, the fourth column shows the re-ranked distances of the retrieved data. The average precision, original and after the re-ranking for each query is also shown. We observe that more number of correct elements are pushed up the ranking order (with their distances decreased) by the re-ranking algorithm.

4.3. Materials dataset

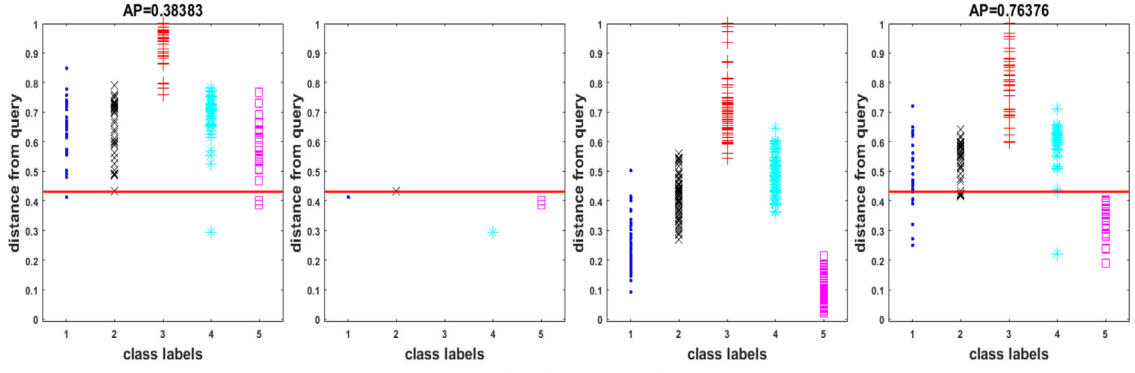
The materials dataset [8] contains two different modalities, namely images and their corresponding audio signatures for 17 different types of materials (Fig. 3). Since pairwise correspondence is not present in this dataset, so we duplicated the samples for the audio domain in the training set, so that there are equal number of feature vectors per class for both the modalities. For the experiments, we randomly divide the features from each class equally to generate the training and testing sets. The results of the baseline method and the re-ranked results are given in Table 4. For this dataset also, we observe significant improvements over all the baseline approaches.

4.4. Multiple features dataset

The multiple features dataset [9] consist of ten classes of handwritten digits (Fig. 3). In this case, instead of learning across differ-



(a) Query belongs to class 4



(b) Query belongs to class 5

Fig. 5. Illustration of the proposed re-ranking algorithm for two different queries. The first column shows the original distances of the different retrieved data of all the classes from the query, from which the K-nearest neighbors are selected (second column). The correct class label of the query is given in the caption. The third column shows the distances given to the training data for regression. Finally, the fourth column shows the re-ranked distances of the retrieved data. The average precision, original and after the re-ranking for each query is also shown.

Table 4
Results (MAP) for the different algorithms on the Materials dataset along with the corresponding re-ranked results using Linear Regression (LR) and Support Vector Regression (SVR).

	Image-Audio			Audio-Image		
	Baseline	LR	SVR	Baseline	LR	SVR
CCA	0.731	0.753	0.751	0.785	0.841	0.857
GMA	0.752	0.760	0.770	0.766	0.775	0.823
CCCA	0.799	0.811	0.812	0.814	0.852	0.856
DCCA	0.775	0.789	0.791	0.800	0.836	0.844
RPSA	0.788	0.804	0.804	0.782	0.819	0.848
CDL	0.782	0.796	0.794	0.815	0.843	0.857
R-KCCA	0.821	0.830	0.827	0.841	0.865	0.870
DCCA-AE	0.813	0.831	0.832	0.827	0.863	0.866

Table 5
Average rank-1 accuracy (%) over all the feature domains for the different algorithms and the proposed algorithm on the Multiple Features dataset.

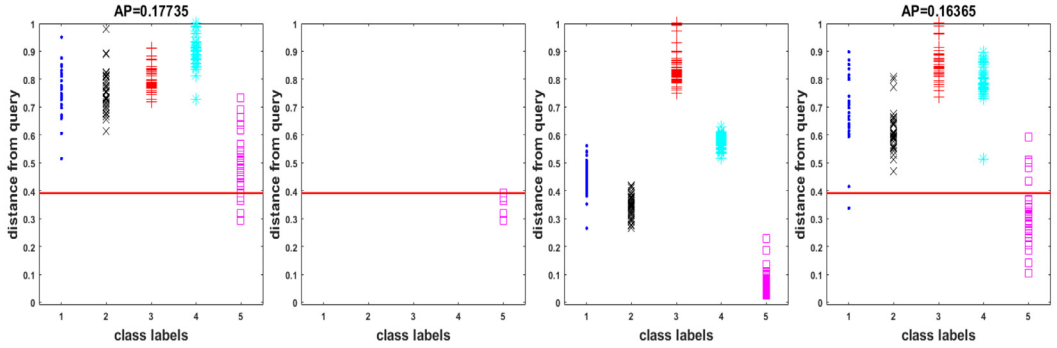
	Average rank-1 accuracy (%)		
	Baseline	LR	SVR
CCA	79.30	82.80	83.20
GMA	83.50	86.00	86.10
CCCA	84.40	86.70	86.80
DCCA	81.70	84.00	84.40
RPSA	83.50	86.60	86.60
CDL	69.70	75.90	74.20
R-KCCA	78.40	82.40	81.40
DCCA-AE	87.10	88.30	88.80

ent modalities, the task is to learn across different feature domains. The source data represents images in one feature domain and the target data has images in some other feature domain. For evaluation, we follow the same protocol as in [29] and randomly split the dataset into two-thirds (and one-thirds) per class to form the training (and testing set) respectively. The results using SVM classifier as in [29] are shown in Table 5. We observe significant gains using the proposed re-ranking algorithm.

5. Analysis and discussion

In this paper, we proposed a re-ranking algorithm for improving the retrieval performance of cross-modal algorithms. The algorithm is based on the assumption that the nearest neighbour set of the retrieved elements contains relevant items with respect to that particular query. If the assumption is satisfied, we observe that the proposed algorithm is able to achieve significant improvements over different baseline methods and across several different datasets. The proposed approach does not extract any new feature from the data and does not require knowledge of the inner workings of the baseline approach. Thus it can be used as an add-on with any cross-modal approach.

However, the re-ranked results may even become worse if the assumption is not satisfied. Consider the two examples in Fig. 6 from the Reduced Wikipedia (top row) and LabelMe dataset (bottom row). For the top example, the query is from class 1 and the retrieved results is shown in column 1. However, since the nearest neighbours are from class 5, in the re-ranked distances, elements from the wrong class are pushed up the ranking list thereby decreasing the average precision from 0.177 to 0.163. For the bottom example also, we observe that since more number of incor-



(a) Query belongs to class 1.

(b) Query - *coast*

Fig. 6. Two examples from the Reduced Wikipedia dataset (top) and the LabelMe dataset (bottom) when the proposed re-ranking algorithm fails. We observe that if the majority of the K-nearest neighbours are incorrect, the re-ranked results can be even worse.

rect matches are retrieved in the K-NN set, the re-ranked results get worse.

In our implementation, we have used LR and SVR to learn the query-specific regression function. We have experimented with different kernels for SVR namely - linear, polynomial (order 3) and the radial basis kernel. Across all the datasets, we observed that in general, SVR with radial basis kernel gave the best performance and hence we have reported the results with that kernel. We also conducted one experiment by varying the number of support vectors for SVR. We observed that as the number of support vectors decreased, the performance improved. This is expected since lesser number of support vectors implies lesser number of points close to the margin which leads to less error.

References

[1] D.R. Hardoon, S.R. Szedmak, J.R. Shawe-taylor, Canonical correlation analysis: an overview with application to learning methods, Neural Comput. 16 (2004) 2639–2664.
[2] H. Hotelling, Relations between two sets of variates, Biometrika 28 (3–4) (1936) 321–377.
[3] A. Sharma, A. Kumar, H. Daume, D.W. Jacobs, Generalized multiview analysis: a discriminative latent space, in: CVPR, 2012, pp. 2160–2167.
[4] D.A. Huang, Y.C. Wang, Coupled dictionary and feature space learning with applications to cross-domain image synthesis and recognition, in: ICCV, 2013, pp. 2496–2503.
[5] W. Wang, R. Arora, K. Livescu, J. Bilmes, On deep multi-view representation learning, in: ICML, 2015, pp. 1083–1092.
[6] J.C. Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. Lanckriet, R. Levy, N. Vasconcelos, On the role of correlation and abstraction in cross-modal multimedia retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 36 (3) (2014) 521–535.
[7] B.C. Russell, A. Torralba, K.P. Murphy, W.T. Freeman, LabelMe: a database and web-based tool for image annotation, Int. J. Comput. Vis. 77 (1–3) (2008) 157–173.
[8] C.H. Lampert, O. Krömer, Weakly-paired maximum covariance analysis for multimodal dimensionality reduction and transfer learning, in: ECCV, 2010, pp. 566–579.
[9] M. Lichman, UCI machine learning repository, 2013, URL: http://archive.ics.uci.edu/ml.
[10] N. Rasiwasia, D. Mahajan, V. Mahadevan, G. Aggarwal, Cluster canonical correlation analysis, in: AISTATS, 2014, pp. 823–831.
[11] D.L. Paz, S. Sra, A.J. Smola, Z. Ghahramani, B. Schölkopf, Randomized nonlinear component analysis, in: ICML, 2014, pp. 1359–1367.
[12] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correlation analysis, in: ICML, 2013, pp. 1247–1255.
[13] Z. Kuang, K. Wong, Relatively-paired space analysis: learning a latent common space from relatively-paired observations, Int. J. Comput. Vis. 113 (3) (2015) 176–192.
[14] C. Zhu, F. Wen, J. Sun, A rank-order distance based clustering algorithm for face tagging, in: CVPR, 2011, pp. 481–488.
[15] W. Li, Y. Wu, M. Mukunoki, M. Minoh, Common-near-neighbor analysis for person re-identification, in: ICIP, 2012, pp. 1621–1624.
[16] L. An, X. Chen, M. Kafai, S. Yang, B. Bhanu, Improving person re-identification by soft biometrics based reranking, in: ICDSC, 2013, pp. 1–6.
[17] S. Khamis, C.H. Kuo, V.K. Singh, V.D. Shet, L.S. Davis, Joint learning for attribute-consistent person re-identification, in: Computer Vision-ECCV 2014 Workshops, Springer, 2014, pp. 134–146.
[18] R.F. Prates R.F. de C., W. Robson S., Appearance-based person re-identification by intra-camera discriminative models and rank aggregation, in: ICB, 2015, pp. 65–72.
[19] Q. Leng, R. Hu, C. Liang, Y. Wang, J. Chen, Bidirectional ranking for person re-identification, in: ICME, 2013, pp. 1–6.
[20] D. Yi, Z. Lei, S.Z. Li, Deep metric learning for practical person re-identification, arXiv preprintarXiv:1407.4979 (2014).
[21] J. Huang, X. Yang, X. Fang, W. Lin, R. Zhang, Integrating visual saliency and consistency for re-ranking image search results, IEEE Trans. Multimed. 13 (4) (2011) 653–661.
[22] D.C.G. Pedronette, R. da S Torres, Exploiting contextual spaces for image re-ranking and rank aggregation, in: ICMR, 2011, p. 13.
[23] A.J. Ma, P. Li, Query based adaptive re-ranking for person re-identification, in: ACCV, 2014, pp. 397–412.
[24] C. Liu, C. Loy, S. Gong, G. Wang, Pop: person re-identification post-rank optimisation, in: ICCV, 2013, pp. 441–448.
[25] B. Prosser, W.S. Zheng, S. Gong, T. Xiang, Q. Mary, Person re-identification by support vector ranking, in: BMVC, 2, 2010, pp. 21.1–11.
[26] G. Lisanti, I. Masi, A.D. Bagdanov, D.B. A., Person re-identification by iterative re-weighted sparse ranking, IEEE Trans. Pattern Anal. Mach. Intell. 37 (8) (2015) 1629–1642.
[27] C.M. Bishop, Pattern recognition and machine learning, 2006.
[28] M. Koestinger, M. Hirzer, P. Wohlhart, P.M. Roth, H. Bischof, Large scale metric learning from equivalence constraints, in: CVPR, 2012, pp. 2288–2295.
[29] Y.R. Yeh, C.H. Huang, Y.C.F. Wang, Heterogeneous domain adaptation and classification by exploiting the correlation subspace, IEEE Trans. Image Process. 23 (5) (2014) 2009–2018.