Learning Class-to-Image Distance via Large Margin and L1-Norm Regularization

Zhengxiang Wang[1,2], Shenghua Gao[1,3], and Liang-Tien Chia[1]

[1] Nanyang Technological University, Singapore
[2] Fujitsu Research & Development Center Co., Ltd, Beijing, China
[3] Advanced Digital Sciences Center, Singapore
{wang0460,gaos0004,asltchia}@ntu.edu.sg

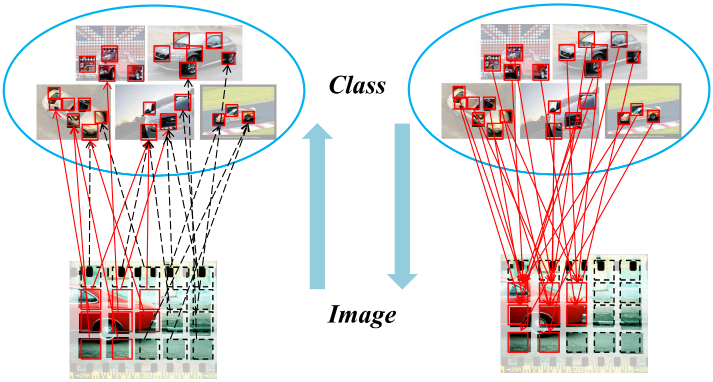
Abstract. Image-to-Class (I2C) distance has demonstrated its effectiveness for object recognition in several single-label datasets. However, for the multi-label problem, where an image may contain several regions belonging to different classes, this distance may not work well since it cannot discriminate local features from different regions in the test image and all local features have to be counted in the I2C distance calculation. In this paper, we propose to use Class-to-Image (C2I) distance and show that this distance performs better than I2C distance for multi-label image classification. However, since the number of local features in a class is huge compared to that in an image, the calculation of C2I distance is much more expensive than I2C distance. Moreover, the label information of training images can be used to help select relevant local features for each class and further improve the recognition performance. Therefore, to make C2I distance faster and perform better, we propose an optimization algorithm using L1-norm regularization and large margin constraint to learn the C2I distance, which will not only reduce the number of local features in the class feature set, but also improve the performance of C2I distance due to the use of label information. Experiments on MSRC, Pascal VOC and MirFlickr datasets show that our method can significantly speed up the C2I distance calculation, while achieves better recognition performance than the original C2I distance and other related methods for multi-labeled datasets.

1 Introduction

Recently Image-to-Class (I2C) distance [1] is proposed to overcome the information loss during feature quantization process in traditional bag-of-words (BOW) model and has demonstrated its effectiveness for object recognition in several single-label datasets. In the I2C distance, a feature set for each class is constructed by gathering all local features in the training images belonging to this class, while the test image is also represented by a set of densely sampled local features. The I2C distance from a test image to a certain class is defined as the sum of Euclidean distance between every local feature in the test image and its Nearest Neighbor (NN) in the class. In [1], they have shown I2C distance performs better than bag-of-words model in several single-label datasets, though

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part II, LNCS 7573, pp. 230–244, 2012.

the whole feature set must be kept during the test phase and therefore increases the memory usage.



(a) Image-To-Class Distance

(b) Class-To-Image Distance

Fig. 1. The illustration of I2C distance (a) and C2I distance (b). Red and black rectangles represent the local features located at the region of “car” and “street” respectively. The NN search directions are represented by arrows from one side to the other side. Red and black arrows represent the NN match between same region and different regions respectively. Though region “street” may not be useful in the “car” class, the NN match from region “street” to “car” in I2C distance (a) cannot be avoided, while in C2I distance (b) it can be avoided.

However, the I2C distance may not perform well in a multi-label problem. In this problem, a test image may contain several regions belonging to different classes. Therefore the resulting feature set for this test image contains local features belonging to multiple different classes and all local features have to be counted in the I2C distance calculation, since it is unknown which local feature belongs to which class before the I2C distance calculation. Consider an example test image in Figure 1 (a), which mainly contains two regions: “car” and “street”. When measuring the distance to each class, local features in every region of the test image are counted in the final I2C distance. In this example, local features located at “street” region are also counted in the I2C distance to class “car”, while distances between these features and their NNs in the class “car” (marked as black arrows) are useless and will make the I2C distance to class “car” inaccurate. Such problem also exists when measuring the distance between this image and class “street”. Since it is even more difficult to first detect the region of “car” and “street” in this test image before the I2C distance calculation, the I2C distance may not be suitable for the multi-label problem.

In this paper, we propose to use Class-to-Image (C2I) distance to measure the distance between image and class for the multi-label problem. We could assume all local features in each class are relevant to this class, either by manually removing irrelevant local features or using some feature selection technologies. Therefore we define the C2I distance as an inverse version of the I2C distance: the Euclidean distance between each local feature in the class feature set and its NN in the feature set of the test image is summed up to form its C2I distance. Such an inversion can solve the problem that I2C distance encounters in the multi-label problem. We explain that by taking the same test image as example. Figure 1 (b) shows the formation of C2I distance from class "car" to the test image. The feature set of class "car" is constructed during the training phase and we should assume most of its elements are located at the regions of "car". If a test image contains a car as in this example, the local features in class "car" will have better opportunities to find their NNs located at the regions of car in this test image and thus a smaller C2I distance than other images without a car. Irrelevant local features in the region of other objects such as "street" in this test image is less likely to be matched as NN features from class "car" and therefore less likely to influence the measurement accuracy of C2I distance compared to I2C distance.

There are two problems in C2I distance. First, the success of C2I distance in the previous example relies on the assumption that most of the local features in the feature set of class "car" are located at the regions of "car". Since the training images also contain multiple regions, we do not know which local features belong to class "car" and which do not. Manually labeling every region to its class label can solve this problem but this is very expensive and most datasets only provide a list of class labels for each training image without pixel-wise or bounding box label information. Second, the number of local features in a class is usually much more than that in an image, resulting in a time-consuming C2I distance calculation when compared to I2C distance.

To solve these problems, in this work, we propose an optimization algorithm using large margin constraint and L1-norm regularization to learn the C2I distance. The feature set of each class is first constructed by gathering all local features in the training images belonging to this class, same as that in I2C distance. Then we associate a weight to every local feature to distinguish the different importance of each local feature, and formulate an optimization problem to learn these weights. We try to use the label information to select relevant local features based on their weights and reduce the size of feature set by L1-norm regularization. Therefore, the objective function of our optimization problem is comprised of two parts: the error term and the regularization term. In the error term, label information is utilized by constraining in each triplet that the C2I distance to the relevant image should be smaller than that to irrelevant image with a large margin. In the regularization term, we use L1-norm on the weight vector to generate a sparse solution as in [2,3]. To solve this optimization problem, we also propose an efficient gradient descent based solver, which can update the weights and converge to the global optimum quickly. After the learning procedure, most of the weights get zero values due to the L1-norm regularization and those local features with zero weight can be removed from the feature set, while the remaining local features with non-zero weight are preserved to make up the feature set. This reduction not only speeds up the C2I distance calculation, but also decreases the memory usage. Using this optimization algorithm, we are able to get better recognition performance with the learned C2I distance even though the size of the resulting feature set is much smaller.

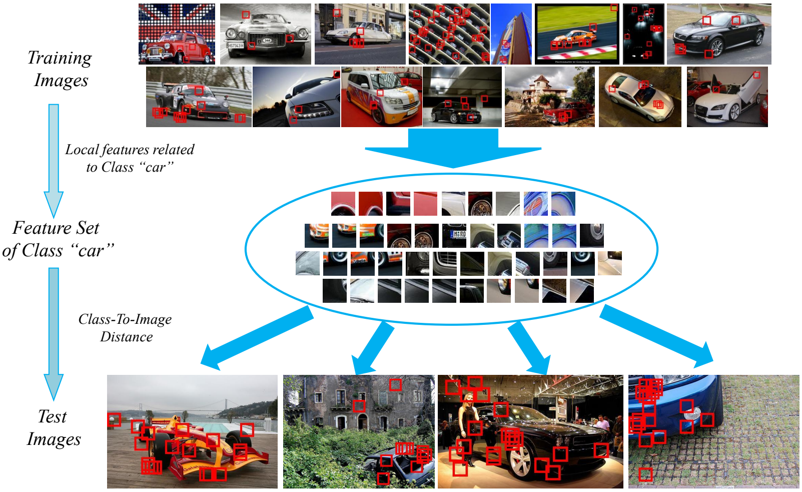

Training Images
Local features related to Class "car"
Feature Set of Class "car"
Class-To-Image Distance
Test Images

Fig. 2. The example of C2I distance from training feature set of class "car" to some test images containing a car object

We show in Figure 2 the example of our learned C2I distance from the training feature set of the class "car" for consistent description, to some test images containing the car. Local features are initially densely sampled from training images having a label of class "car". After the learning algorithm of our optimization framework, only those local features with non-zero weight (Marked with red rectangles in the Figure 2) are preserved to make up the final feature set. The local features of test image are also densely sampled to form the image feature set. To evaluate the C2I distance from the class "car" to each test image, every local feature in class "car" needs to find its NN in the feature set of each test image (Those local features matched as NN features in the test images are also marked with red rectangles in Figure 2, and we can see that most of them are correctly located at the region of car). The sum of Euclidean distance from every local feature in class "car" to its NN feature in a test image multiplied by the learned weight is defined as the C2I distance to this image. We summarize the main contributions of this paper as follows:

- We propose to use C2I distance for the multi-label problem and show that this distance performs better than I2C distance when test images contain multiple class labels.
- We propose an optimization algorithm using large margin constraint and L1-norm regularization to learn the C2I distance. With this optimization algorithm, we are able to get better recognition performance, while the size of feature set is significantly reduced.
- To solve this optimization problem, we propose an efficient gradient descent based solver, which can update the weight and converge to global optimum quickly.

We organize the rest of this paper as follows. Section 2 briefly reviews previous research related to our work. We describe the objective function and learning algorithm in Section 3, and evaluate its performance compared with other methods in Section 4. Finally, we conclude this paper in Section 5.

2 Related Work

Recent research tries to avoid the information loss in feature quantization process, some methods measure the distance from set to set directly and some use other strategies [4]. The former one can be roughly divided into three types: The first one is I2I distance as proposed by Frome *et al.* [5,6], which shows good performance in image classification and retrieval. However, its recognition performance in image classification is not comparable to the second type of I2C distance proposed in [1], which achieves state-of-the-art performance in several single-label datasets. The effectiveness of I2C distance attracts many later research work on it. For example, Wang *et al.* [7,8] learn a per-class Mahalanobis distance metric and weighted I2C distance for improving the recognition performance, Behmo *et al.* [9] learn an optimal NBNN by hinge-loss minimization to further enhance its generalization ability, Tuytelaars *et al.* [10] propose a kernelized version of I2C distance as complimentary to standard bag-of-words based kernels, McCann and Lowe [11] modify the formulation of I2C distance and propose a local NBNN method, which can not only speed up the distance calculation, but also improve the recognition performance.

However, I2C distance is not suitable for the multi-label problem since it cannot distinguish local features from different regions in a test image. This problem can be solved by the third type of C2I distance as shown in previous section. Wang *et al.* [12,13] first use this distance for multi-instance learning problems. However, they do not discuss the advantage of C2I distance over I2C distance for the multi-label problem. The reduction of local features for C2I distance calculation is also ignored, which is an important issue to make C2I distance practical since the number of local features in a class is much more than that in an image. Our work addresses this problem and uses L1-norm regularization to learn a sparse solution, so that the irrelevant local features receive a zero weight and can be removed directly from the feature set, which can not only accelerate the distance calculation, but also reduce the memory usage during the test phase. Our work is similar to the weight learning method in [8]. Both methods use large margin constraint while [8] use L2-norm regularization but we use L1-norm. This is also the difference between C2I and I2C distances. In I2C distance, it is not necessary to use L1-norm regularization since no local feature in each class requires to be removed before the NN search. Other similar methods of using large margin or weighting model for the multi-label problem includes [14,15].

3 Learning Class-to-Image Distance

We briefly describe the notations used in this paper before describing the objective function of our optimization problem. For a class C_i, all the local features extracted from the training images belonging to this class are gathered together to make up the initial feature set of this class, and are denoted as $F_{C_i} = \{f_{C_i1}, f_{C_i2}, \ldots, f_{C_im_i}\}$, where m_i represents the total number of local features in C_i and each local feature is denoted as $f_{C_ik} \in R^d, \forall k \in \{1, \ldots, m_i\}$. Similarly, the feature set of an image X_j is denoted as $F_{X_j} = \{f_{X_j1}, f_{X_j2}, \ldots, f_{X_jm_j}\}$, where m_j represents the total number of local features in image X_j. To calculate the C2I distance from class C_i to image X_j, every local feature f_{C_ik} in F_{C_i} needs to find its NN feature in F_{X_j}, which we denote as $f_{C_ik}^{X_j}$. The sum of Euclidean distance between every local feature f_{C_ik} in F_{C_i} and its NN feature $f_{C_ik}^{X_j}$ is defined as the C2I distance from class C_i to image X_j and is denoted as $Dist(C_i, X_j)$:

$$Dist(C_i, X_j) = \sum_{k=1}^{m_i} \| f_{C_ik} - f_{C_ik}^{X_j} \|^2 \tag{1}$$

where

$$f_{C_ik}^{X_j} = \underset{t=\{1,\ldots,m_j\}}{\arg\min} \| f_{C_ik} - f_{X_j,t} \|^2 \tag{2}$$

We associate a weight W_{C_ik} to every local feature f_{C_ik} in class C_i to distinguish the different importance of each local feature and learn a weighted C2I distance. All the weights can be put into a long vector W and a distance vector $D(C_i, X_j)$ is constructed as well with the same dimension to W, so the weighted C2I distance is formulated as:

$$W^T \cdot D(C_i, X_j) = \sum_{k=1}^{m_i} W_{C_ik} \cdot \| f_{C_ik} - f_{C_ik}^{X_j} \|^2 \tag{3}$$

The objective function of our optimization problem is composed of two terms: the regularization term and the error term. In the error term, we form the triplet constraints by selecting one class with one relevant image and one irrelevant image for each triplet and constrain that the C2I distance from class C_i to the relevant image X_p should be smaller than the distance to the irrelevant image X_n with a large margin:

$$W^T \cdot D(C_i, X_n) - W^T \cdot D(C_i, X_p) \geq 1 - \xi_{ipn} \tag{4}$$

where ξ_{ipn} is a slack variable in order to allow soft-margin. With these triplet constraints, the learned weights are able to discriminate the different importance of local features in the class. Higher important local features are likely to get higher value weight and less important local features will get smaller weight, while irrelevant local features will get zero value weight and are removed from the feature set.

In the regularization term, if we impose L2-norm regularization on the weight vector W like standard SVM framework, the sparseness of the weight vector cannot be guaranteed. Since the number of local features in a class is usually very large, the sparsity of the learned weight is very important to make C2I distance practical. Therefore, in this paper we use L1-norm regularization in the objective function and try to learn a more sparse weight vector. The objective function of our optimization framework with L1-norm regularization is formulated as follows:

$$\begin{aligned} &\min_{W} O(W) = \|W\|_1 + C \sum_{i,p,n} \xi_{ipn} && (5)\\ s.t. \forall\ i,p,n:\ &W^T \cdot (D(C_i, X_n) - D(C_i, X_p)) \geq 1 - \xi_{ipn} \\ &\xi_{ipn} \geq 0 \\ \forall\ k:\ &W(k) \geq 0 \end{aligned}$$

where C is used to control the trade-off between regularization term and error term. With this objective function, less important and irrelevant local features will be more likely to receive a zero weight and be removed from the feature set, making the size of final feature set smaller and therefore accelerating the C2I distance calculation. To solve this new optimization problem, we propose a gradient descend based method, which is able to effectively and efficiently update the weight vector.

Since we guarantee the non-negativeness of weight vector, the L1-norm $\|W\|_1$ can be reformulated as $W^T \cdot e$ where e is a constant vector with all entries to be 1 and same dimension to W, and this optimization problem is convex w.r.t. W. Therefore, the gradient descend based method can be used to iteratively update the weight vector and is guaranteed to converge to the global optimum. In each iteration, the weight vector is updated by taking a small step along the negative gradient direction to reduce the objective function and then negative components are truncated to zero to ensure the non-negativeness. To calculate the gradient of the objective function over the weight vector W, we first rewrite this objective function by replacing the slack variable ξ_{ipn} using the triplet constraints. To simplify, we denote $D(C_i, X_n) - D(C_i, X_p)$ by X_{ipn}. Since

$$\xi_{ipn} = 0 \quad if \qquad W^T \cdot X_{ipn} \geq 1 \tag{6}$$

and

$$\begin{aligned} \xi_{ipn} > 0 \ \ if \qquad &W^T \cdot X_{ipn} < 1 \\ and \qquad &W^T \cdot X_{ipn} + \xi_{ipn} = 1 \end{aligned} \tag{7}$$

at each iteration t, we can first scan over all triplets to find a set of unsatisfied triplets where $\xi_{ipn} > 0$. We denote N^t as the set of triplet indices such that $(i,p,n) \in N^t$ if $\xi_{ipn} > 0$. Then the objective function at t^{th} iteration can be rewritten as:

$$O(W) = W^T \cdot e + C \sum_{(i,p,n) \in N^t} (1 - W^T \cdot X_{ipn}) \tag{8}$$

The gradient $G(W^t)$ at t^{th} iteration can be calculated by taking the derivative of the above objective function w.r.t. W:

$$G(W^t) = e - C \cdot (e^T \cdot \sum_{(i,p,n) \in N^t} X_{ipn}) \tag{9}$$

Therefore, the weight vector W^t at t^{th} iteration is calculated as:

$$W^t = W^{t-1} - \alpha \cdot G(W^t) \tag{10}$$

where α is the step size, which is tuned during the iterations using the method in [16]. To satisfy the non-negativeness, all negative components of the weight vector are truncated to zero after each updating. Since the whole optimization problem is convex w.r.t. W, it is guaranteed to converge to the global optimum after iterative updating.

The main computation bottleneck in this solving method is the calculation of N^t in each iteration t, since every triplet constraint needs to be checked by calculating ${W^t}^T \cdot X_{ipn}$ (If ${W^t}^T \cdot X_{ipn} < 1$, then $\xi_{ipn} > 0$ and $(i,p,n) \in N^t$). This calculation is time-consuming as the dimension of W and X_{ipn} depends on the initial feature set size and is usually very huge, and the number of triplets is also large. To accelerate this calculation, we can keep an active set for triplets that have been violating the constraints and scanning only the triplets in the active set during each iteration. The full scanning over all triplets is only made at the beginning of the algorithm and every 10-20 iterations. Meanwhile, since the weight vector is usually very sparse after a few iterations, the calculation of ${W^t}^T \cdot X_{ipn}$ can be accelerated as only the non-negative components are involved in the calculation. The whole work flow of updating the weight vector for each class is summarized in Algorithm 1.

Algorithm 1. The Algorithm for Solving Our Optimization Problem

```
Input: step size $\alpha$, parameter $C$ and pre-calculated data $X_{ipn}$
$W^0 := e$ {Initialize the weight vector with all entries equal to 1}
Set $t := 0$
repeat
   Compute $N^t$ by checking each error term $\xi_{ipn}$
   Update $G(W^t)$:
         $G(W^t) := e - C \cdot (e^T \cdot \sum_{(i,p,n) \in N^t} X_{ipn})$
   Update $W^t$:
         $W^t := W^{t-1} - \alpha \cdot G(W^t)$
   Truncate negative components of $W^t$ to 0
   Calculate new objective function
   $t := t + 1$
until Objective function converged
Output: Weight Vector $W$
```

4 Experiment

In this section, We evaluate the performance of our method and compare it with other related methods on three multi-label image datasets: MSRC [1], Pascal VOC 2011 [17] and MirFlickr [18].

4.1 Dataset Setup

We briefly describe the setup for each of the three datasets as follows:

MSRC dataset contains 591 images and 21 class labels, and the average number of class labels per image is 2.46. We use 10-fold cross validation on this dataset. Specifically, the dataset is divided into 10 parts and each time 9 parts are used for training and the rest one is used for test, and the average result over 10 runs is reported.

VOC2011 dataset is a much larger dataset, which contains a total of 28,952 images annotated into 20 classes. However, the majority of images in this dataset are single-labeled, on average about 1.44 labels per image. To better evaluate our method for the multi-label problem, we select a subset where most of images have more than one label. Specifically, We randomly select images with multiple labels such that each class contains at least 50 training and 50 test images, which leads to 1777 images used in our experiment and the average number of class labels per image is increased to 2.12. This random selection is conducted 5 times and the average result is reported.

MirFlickr dataset consists of 25000 images downloaded from Flickr.com. There are 1386 tags which occur in at least 20 images. We use the most common 30 content based tags as suggested by [18] in our experiment and select at least 100 image for each class, resulting in average 2.47 class labels per image. The selected image collections are equally divided into the training and test sets and we run 5 random partitions to report the average result.

In each dataset, we use SIFT feature [19] as our descriptor densely sampled at every 10 pixels to make up the feature set for training and test images. We evaluate our method of Learning the C2I distance with weight learned using L1-norm regularization, which is denoted as **"LC2I-L1"**, and compare it to:

- **"LC2I-L1-FS"**, which only uses the local features with non-zero weight to calculate the C2I distance but does not multiply with the learned weight, to show the performance of our Feature Selection strategy;
- **"C2I"**, the original C2I distance summed up by Euclidean distance from all local features without feature selection and weighting;
- **"C2I-DVW-FS"**, C2I distance with the Feature Selection method of selecting Discriminative Visual Words used in [20];
- **"LC2I-L2"**, the Learned C2I distance with weight learned using L2-norm regularization, which is solved using the method in [5].

[1] `http://research.microsoft.com/en-us/projects/objectclassrecognition/`

We also compare with other related methods:

- **"I2C"**, the original NBNN method in [1];
- **"LI2C"**, the Learning of weighted I2C distance [8];
- **"Local NBNN"**, an improved I2C distance [11];
- **"NBNN Kernel"** [10], I2C distance as input of One-vs-All SVM with histogram intersection kernel
- **"BoW"**, Bag-of-Words model [21] for image representation with One-vs-All SVM of histogram intersection kernel.

We use mean average precision (MAP) to evaluate and compare different methods in our experiment as in [17]. All the NN searches involved in C2I and I2C distance calculations are approximated by kd-tree implemented by VLFeat [22], with 5 kd-trees in each feature set and a maximum of 100 comparisons for each NN search, which does not show much difference on the final MAP compared to that of accurate NN search.

4.2 Experiment Result

Table 1 shows the result of different methods on the three datasets, and we can see that the proposed method, "LC2I-L1" significantly outperforms all other methods. In particular, the improvement over the original C2I distance can be divided into two parts: feature selection by removing local features with zero value weight and the discrimination of different importance for the preserved local features by their associated non-zero weight.

The improvement over the first one of feature selection can be seen by comparing the performance of "LC2I-L1-FS" to "C2I". "LC2I-L1-FS" only uses the weight to select relevant local features without weighting in C2I distance calculation, and its improvement over the original C2I distance shows that this feature selection is able to remove the irrelevant local features that deteriorate the performance of C2I distance. These irrelevant local features have to be put into the initial feature set of each relevant class because the training images may also

Table 1. Performance comparison using MAP

Method	MSRC	VOC2011	MirFlickr
LC2I-L1	**0.6379 ± 0.0020**	**0.3522 ± 0.0036**	**0.2561 ± 0.0029**
LC2I-L1-FS	0.5679 ± 0.0037	0.3238 ± 0.0035	0.2306 ± 0.0061
C2I	0.3438 ± 0.0007	0.2180 ± 0.0016	0.1591 ± 0.0014
C2I-DVW-FS [20]	0.3387 ± 0.0041	0.2035 ± 0.0007	0.1534 ± 0.0012
LC2I-L2 [5]	0.5864 ± 0.0010	0.3016 ± 0.0001	0.2078 ± 0.0009
I2C (NBNN) [1]	0.3549 ± 0.0016	0.1508 ± 0.0011	0.1548 ± 0.0023
LI2C [8]	0.5231 ± 0.0017	0.1690 ± 0.0003	0.1619 ± 0.0016
Local NBNN [11]	0.6019 ± 0.0052	0.2669 ± 0.0018	0.1979 ± 0.0021
NBNN Kernel [10]	0.6054 ± 0.0033	0.2693 ± 0.0041	0.1782 ± 0.0009
BoW [21]	0.6110 ± 0.0024	0.3096 ± 0.0014	0.2299 ± 0.0033

belong to multiple classes but the information that which region in the image belongs to which class is not provided since this pixel-wise or bounding box human labeling is expensive. Due to the inclusion of these irrelevant local features, the original C2I distance does not show better performance compared to I2C distance on MSRC and MirFlickr datasets as shown in Table 1. However, with the feature selection to remove these irrelevant local features, the performance of C2I distance can be largely improved, as comparing "LC2I-L1-FS" to "C2I".

The improvement over the second part of non-zero weighting can be recognized by comparing the performance of "LC2I-L1" to "LC2I-L1-FS". This improvement shows the learned weight not only provides a feature selection strategy to reduce the size of feature set and remove irrelevant local features, but also discriminates the different importance of the preserved local features in the final feature set by their associated non-zero weight to further improve the C2I distance, which validates the effectiveness of our learning algorithm.

Compared to I2C distance related methods, our learned C2I distance is able to achieve better recognition performance under the multi-label problem. Though I2C distance of non-learning methods like NBNN and Local NBNN may benefit from a larger feature set sampled from denser grids, their performances are still worse than our method. To verify this, we have tried sampling at every 4 pixels and find Local NBNN is only slightly improved to 0.2846 in VOC2011 dataset. It should be noted the improvement over I2C distance methods is based on the multi-label problem. When more images are single-labeled, e.g. in the full VOC dataset, such large improvement is not expected. Due to the computation limitation of our working machine for the heavy NN search in distance calculation, we only try class "aeroplane", "bicycle", "car", "chair" and "person" for the full VOC 2007 dataset, and find the performance of our method is worse than I2C distance (0.502 vs 0.649 [23]), while in our multi-label subset version of these classes, our method significantly outperform NBNN (0.467 vs 0.247).

We also compare the performance of learning weight using L1-norm regularization to L2-norm. By comparing "LC2I-L1" to "LC2I-L2" in Table 1, we can see that even using L1-norm to learn a sparser solution, the recognition performance is still better than using L2-norm, which again shows the effectiveness of our learning algorithm. Meanwhile, the sparsity of the resulting feature set learned by L1-norm is much lower than L2-norm. This is shown in Table 2, where the sparsity is represented by the percentage of the number of preserved local features compared to the original feature set. With L1-norm regularization, only 3.35% local features are preserved in MSRC dataset and in VOC2011 and MirFlickr datasets, this is reduced to only 0.26% and 0.31% respectively. This is a much larger reduction on both computational cost of C2I distance calculation and memory usage, since their complexities are linear to the size of class feature set. To verify this reduction, we show in Table 3 the average running time of distance calculation for a test image over all classes and Table 2 shows the memory usage for loading the training feature sets of all classes. All of the experiments are running on a single core of **Intel x86 Xeon CPU E7320@2.13GHz** and **16GB** memory.

Table 2. sparsity comparison (Percentage of the number of local features compared to the original feature set)

Method	MSRC	VOC2011	MirFlickr
LC2I-L1	**3.35% ± 0.14**	**0.26% ± 0.01**	**0.31% ± 0.02**
LC2I-L2 [5]	12.50% ± 0.10	5.86% ± 0.16	1.58% ± 0.32
C2I-DVW-FS [20]	18.45% ± 0.15	16.20% ± 0.09	16.85% ± 0.12
C2I,I2C [1]	100%	100%	100%

Table 3. running time comparison (second)

Method	MSRC	VOC2011	MirFlickr
LC2I-L1	**0.61 ± 0.03**	**0.63 ± 0.05**	**0.61 ± 0.04**
LC2I-L2 [5]	3.02 ± 0.22	5.64 ± 0.53	1.86 ± 0.15
C2I-DVW-FS [20]	9.12 ± 0.41	21.08 ± 1.53	22.20 ± 1.01
C2I	51.95 ± 1.07	132.88 ± 5.03	129.72 ± 3.05
I2C [1]	5.08 ± 0.13	7.44 ± 0.61	14.29 ± 0.33
Local NBNN [11]	1.18 ± 0.09	3.04 ± 0.22	3.57 ± 0.15

Table 4. memory usage comparison (MB)

Method	MSRC	VOC2011	MirFlickr
LC2I-L1	**5.91 ± 0.04**	**6.12 ± 0.05**	**5.38 ± 0.08**
LC2I-L2 [5]	31.19 ± 0.51	65.38 ± 2.44	20.34 ± 0.40
C2I-DVW-FS [20]	101.88 ± 1.49	240.25 ± 3.72	254.08 ± 4.13
C2I,I2C [1]	547.88 ± 3.26	1508.94 ± 10.28	1512.23 ± 15.11

Compared to I2C distance, the original C2I distance needs much more expensive computational cost since the number of local features in a class is much more than that in an image, while both methods need the same amount of memory usage to store the whole feature sets of classes for distance calculation. However, with our learning method using L1-norm regularization, only a very small number of local features are preserved in the final feature set for each class. Therefore, the computational cost of our C2I distance can be even lower than I2C distance, and the memory usage is greatly reduced. Even compared to Local NBNN, which is the most efficient method in I2C distance computation so far, our method still needs less running time for distance calculation.

The method of "C2I-DVW-FS" clusters the feature set of each class to 1000 visual words and selects the top 64 most discriminative visual words. Local features belonging to these visual words are selected to make up the final feature set for each class and used in C2I distance. However, the result in Table 1 and Table 2 shows that its recognition performance is even worse than the original C2I distance, probably because label information is not utilized in the clustering and selection procedure. Therefore this method cannot distinguish relevant local features used in C2I distance. Meanwhile, its sparsity is not as good as our learning method. In summary, our learned C2I distance not only achieves better recognition performance, but also greatly reduces the computational cost and memory usage.

4.3 The Influence of Parameter C

We also analyze the impact of the trade-off parameter C in the objective function 5. Table 5 shows the MAP and sparsity under different C values ranging from 10^{-3} to 10^{1} on MSRC dataset. With the increase of C, the size of the feature set is always increased, meaning more local features are preserved and more computational cost and memory usage are required. This can be recognized as the increase of C makes the objective function pay less emphasis on the sparsity of the weight vector and therefore resulting in more local features preserved. However, when C is increasing, the recognition performance is first increased but then decreased, as a large value of C might put too much emphasis on the error term to cause over-fitting. The best MAP is achieved when C is equal to 10^{-2} on MSRC dataset. If we further increase C after this peak point, both the recognition performance and efficiency are getting worse.

Table 5. MAP and sparsity of our method over different C value on MSRC dataset

C	MAP	Sparsity
10^{-3}	0.6027 ± 0.0031	1.73% ± 0.03
10^{-2}	0.6379 ± 0.0020	3.35% ± 0.14
10^{-1}	0.6054 ± 0.0007	8.71% ± 0.23
10^{0}	0.5727 ± 0.0020	14.97% ± 0.51
10^{1}	0.5669 ± 0.0007	20.55% ± 0.95

5 Conclusion

In this paper, we have discussed the drawback of I2C distance in classifying multi-label images and proposed to use C2I distance for dealing with multi-label image classification. To select relevant local features for each class and speed up the C2I distance calculation, we associated a weight for each local feature densely sampled from images, and proposed an optimization algorithm using large margin constraint and L1-norm regularization to learn these weights. We also proposed a gradient descend based solver to efficiently and effectively update these weights in the learning procedure. Experiments on three multi-label datasets have shown that our learning algorithm can not only improve the performance of C2I distance and outperform other related methods, but also significantly reduce the number of local features in the class feature set and greatly improve the efficiency.

However, it should be noted such performance and efficiency improvements in the test phase is a trade-off of the additional training phase for learning the C2I distance. The original C2I distance calculation in the training phase is computation expensive since this is done before the learning procedure. Therefore, when there are a large number of local features in the original feature set, an initial local feature selection before the learning procedure may be required to alleviate the computational cost of distance calculation in the training phase, which would be a possible future direction.

Acknowledgement. This study is done when Zhengxiang Wang and Shenghua Gao were studying in Nanyang Technological University and is supported by Fujitsu Research & Development Center Co., Ltd and the research grant for the Human Sixth Sense Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR).

References

1. Boiman, O., Shechtman, E., Irani, M.: In defense of nearest-neighbor based image classification. In: CVPR (2008)
2. Bradley, P.S., Mangasarian, O.L.: Feature selection via concave minimization and support vector machines. In: ICML (1998)
3. Yuan, G.X., Chang, K.W., Hsieh, C.J., Lin, C.J.: A comparison of optimization methods and software for large-scale L1-regularized linear classification. JMLR 11(52) (2010)
4. Perronnin, F., Liu, Y., Sanchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: CVPR (2010)
5. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: ICCV (October 2007)
6. Frome, A., Singer, Y., Malik, J.: Image retrieval and classification using local distance functions. In: NIPS, vol. 19 (2006)
7. Wang, Z., Hu, Y., Chia, L.-T.: Image-to-Class Distance Metric Learning for Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 706–719. Springer, Heidelberg (2010)
8. Wang, Z., Hu, Y., Chia, L.T.: Improved learning of i2c distance and accelerating the neighborhood search for image classification. Pattern Recognition 44(10-11), 2384–2394 (2011)
9. Behmo, R., Marcombes, P., Dalalyan, A., Prinet, V.: Towards Optimal Naive Bayes Nearest Neighbor. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 171–184. Springer, Heidelberg (2010)
10. Tuytelaars, T., Fritz, M., Saenko, K., Darrell, T.: The nbnn kernel. In: ICCV, pp. 1824–1831 (2011)
11. McCann, S., Lowe, D.G.: Local naive bayes nearest neighbor for image classification. In: CVPR (2012)
12. Wang, H., Nie, F., Huang, H.: Learning instance specific distance for multi-instance classification. In: AAAI (2011)
13. Wang, H., Huang, H., Kamangar, F., Nie, F., Ding, C.: Maximum margin multi-instance learning. In: NIPS, vol. 24 (2011)
14. Verbeek, J., Guillaumin, M., Mensink, T., Schmid, C.: Image annotation with tagprop on the mirflickr set. In: MIR (2010)
15. Lampert, C.H.: Maximum margin multi-label structured prediction. In: NIPS, vol. 24 (2011)
16. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. JMLR 10, 207–244 (2009)
17. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2011 (VOC 2011) (2011) (Results)
18. Huiskes, M.J., Thomee, B., Lew, M.S.: New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. In: MIR, pp. 527–536 (2010)

19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2) (2004)
20. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: CVPR (2007)
21. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: ICCV (2003)
22. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), http://www.vlfeat.org/
23. Timofte, R., Gool, L.V.: Iterative nearest neighbors for classification and dimensionality reduction. In: CVPR (2012)