How Can I Index My Thousands of Photos Effectively and Automatically? An Unsupervised Feature Selection Approach

Juhua Hu* Jian Pei† Jie Tang‡

Abstract

Given a large photo collection without domain knowledge (e.g., tourism photos, conference photos, event photos, images wrapped from webpages), it is not easy for human beings to organize or only view them within a reasonable time. In this paper, we propose to automatically extract meaningful semantics from a photo collection named “dimensions” to help people view, search and organize photos conveniently and efficiently. However, due to the lack of additional domain knowledge or content information, existing image retrieval techniques are not applicable. To tackle the problem, we first propose a simple strategy to extract all meaningful semantics from original photos/images as candidate dimensions, and then propose an efficient unsupervised feature/dimension selection method to select a sufficient dimension subset to uniquely index each photo within this collection. Our experiments on several real-world photo/image collections validate both the efficiency and effectiveness of our proposed method.

1 Introduction

You just came back from a wonderful vacation at your favorite place. Your digital camera, equipped with only a 64GB flash memory card, records your lovely memory in thousands of pictures. In your computer, you have tens of thousands of photos taken in the last 5 years. You want to share those photos online with your friends, but you face a challenge. Many of your friends unlikely have the time and patience to browse thousands of photos. Ideally, you would like to organize your thousands of photos using a small number of meaningful features, such as “places”, “my kids”, “our pets”, and “classmate reunion”, so that every photo can be uniquely identified and retrieved using a combination of those features. In other words, the features have to be discriminative and independent. We call such a small number of meaningful features a *multidimensional index* (or *index* for short) of the photos. Here, an index is for human users instead of for software search engines. It should be easy to understand and manipulate.

Creating an index of thousands of photos manually is time consuming. Moreover, a manually created index without a careful design may not be able to facilitate search and retrieval effectively. For example, using too many features may overwhelm users. Some photos may need a combination of many features to be retrieved, and thus are deeply hidden. Many photos may not be uniquely identified by feature combinations.

Most existing image retrieval techniques [19] rely heavily on manually or algorithmically annotated images, which is often expensive and unavailable in our problem setting. Fortunately, by applying the state-of-the-art methods, we can extract a large set of meaningful semantics from the original photos as “dimensions”. Concretely, each photo may contain multiple semantic dimensions, such as “chair”, “computer”, “sunset”, and “ocean”. Inspired by the idea of multi-instance multi-label learning [14, 25], where each data object is represented by multiple instances, all meaningful regions for each image can be roughly detected by image segmentation, and then each region can be presented as an instance. Consequently, each instance can be treated as a semantically meaningful dimension for people to search, view and organize these images. For example, given the photo collection from the conference KDD’12, the dimension “stand” as red line circled in Figure 1 could be used to indicate whether an image has the content of “stand”.

However, such semantic annotation methods typically generate many features. One may wonder whether the more features, the better a large set of photos are indexed. If one uses all features as dimensions, many dimensions may be redundant, since different images may contain similar content. A dimension existing in all or many images is not discriminative and not useful for indexing photos. For example, image feature “football” for a collection of football game photos is not informative, since most photos contain footballs.

Thus, although extracting semantic features from photos is highly feasible, the real challenge of selecting effective features to index a large number of objects heavily remains. Ideally, given a collection of photos, we want to find a minimum set of dimensions such that each image in the collection can be uniquely identified by some of those selected dimensions.

One may wonder if this is just an instance of the well studied dimensionality reduction problem [5]. It

*Simon Fraser University, juhuah@sfu.ca.
†Simon Fraser University, jpei@cs.sfu.ca.
‡Tsinghua University, jietang@tsinghua.edu.cn.

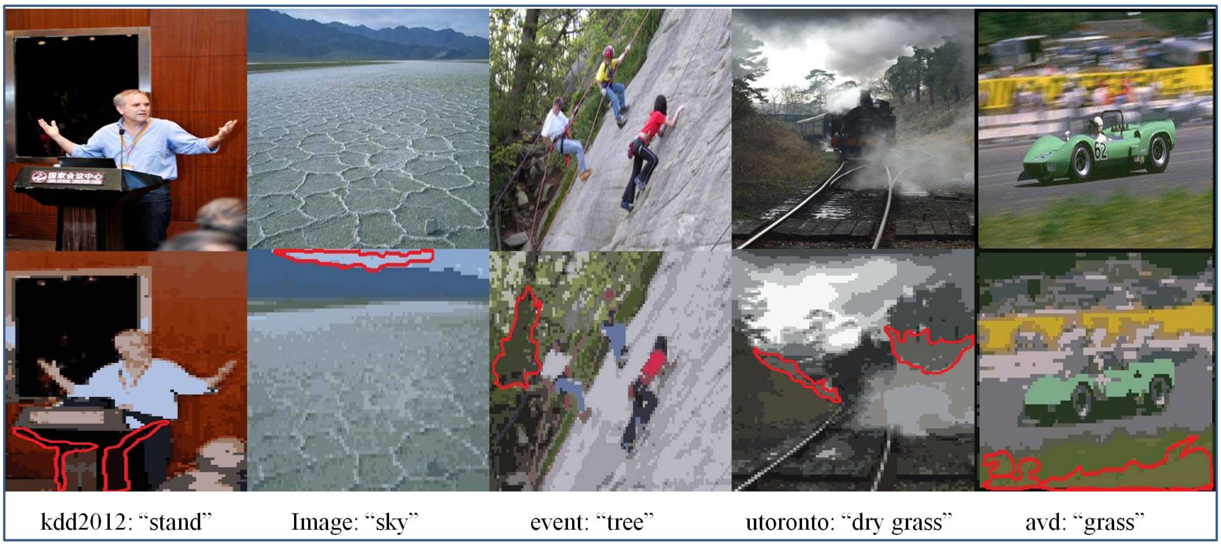


Figure 1: The 1st dimension selected for some photo collections with manual label when $\theta = 0.70$

is well known that dimensionality reduction removes irrelevant and redundant features. Dimensionality reduction methods can be categorized mainly into two groups, namely feature extraction and feature selection. Feature extraction approaches, such as PCA, LDA, SVD and patch alignment [24], are not applicable in our case, since they project features into a new space where new features are usually not physically interpretable as those extracted dimensions in Figure 1. At the same time, the lack of label and class information makes some popular feature selection methods, such as Information Gain, Relief, Fisher Score, and Lasso, not applicable here, either. Some unsupervised feature selection methods have been proposed for clustering [1]. However, those methods usually try to select features that can improve the clustering quality, and assume that similar objects should have similar feature values according to the similarity graph structure. Since our goal is to index each photo uniquely, the unsupervised feature selection methods for clustering are not reliable for our problem, either. Therefore, although our problem is a kind of dimensionality reduction, it cannot be tackled well using the existing methods.

In this paper, we formulate the novel problem of unsupervised minimum feature selection for high dimensional object indexing. This problem is very challenging, and we prove that this optimization problem is submodular [11]. Therefore, we propose an efficient greedy algorithm. Our experiments on various real-world photo and image collections validate both the efficiency and effectiveness of our proposed method.

The rest of the paper is organized as follows. Section 2 briefly reviews the related work on image annotation and feature selection. Section 3 formulates our problem. Section 4 presents our methods. Section 5 reports an empirical study. Section 6 concludes the paper. In the rest of the paper, we use the terms “photo” and “image” interchangeably.

2 Related Work

Our study is mainly related to the existing work on image annotation and feature selection. We review the state-of-the-art literature briefly in this section. A thorough survey is far beyond the capacity of this paper.

2.1 Image Annotation Automatic image annotation has been widely studied. Many approaches have been proposed. The earliest methods use segmentation and translation. For instance, Mori *et al.* [17] introduced a co-occurrence model that uniformly divides an image into regions with keywords and correlates each image to a set of keywords. Duygulu *et al.* [4] used a machine translation approach that translates from a vocabulary of blobs in an image to a vocabulary of words. They segmented an image into regions that are mapped to keywords.

The early image annotation approaches generally evaluate regions individually and overlook the correlation between different regions. This has been improved by some later methods. Jeon *et al.* [7] proposed the fixed annotation-based cross-media relevance model (FACMRM), which takes advantage of the joint distribution of words and blobs. The experiments show that the FACMRM performs almost six times better than a word-blob co-occurrence model and two times better than a model based on machine translation.

More recently, Cao *et al.* [2] suggested to enhance the annotation performance by first finding high-confidence annotation labels for certain images and then propagating to the remaining images according to the similarity of time, location, and visual context.

In real-world applications, more often than not we are facing a huge photo collection without any domain knowledge, let alone mentioning training annotations or image labels. Our paper addresses this challenge.

Our study builds on top of some image annotation techniques. Specifically, as will be described later, we apply some state-of-the-art image annotation techniques to extract features from a large set of images. However, our goal is not on image annotation. Instead, we want to select a minimum set of features so that each image can be indexed uniquely using the combination of features.

2.2 Feature selection Feature selection has been widely studied in the context of supervised and unsupervised learning, and also in improving image annotation. For instance, Carneiro *et al.* [3] introduced the supervised multiclass labeling (SML) method that combines the advantages of supervised and unsupervised learning. The main idea is to let different classes compete for an image in annotation. The competition results in a natural ordering of semantic labels.

Supervised learning often comes with high computational cost. Some studies try to reduce the computational cost. For example, Ma *et al.* [15] proposed structural feature selection with sparsity (SFSS), which jointly selects the most relevant features from all data points using a sparsity-based model and exploits both labeled and unlabeled data to learn a manifold structure.

Feature selection has also been investigated in unsupervised learning, also known as clustering. When applying feature selection in unsupervised learning, one primary interest is to determine what attributes of data can help to obtain better clustering results. The problem of unsupervised feature selection is more challenging because the number of clusters is often unknown.

Law *et al.* [10] presented an expectation-maximization (EM) algorithm for unsupervised clustering with feature selection. This algorithm estimates the salience of features and the optimal number of clusters. Later, Yen *et al.* [22] proposed to use an eigen-decomposition method to find dependent features by choosing a set of coefficients with the linear combination of features that are close to zero, and remove the dependent features with the largest absolute coefficient.

Many studies focus on reducing the size of a feature set. Velayutham and Thangavel [20] presented a novel unsupervised entropy based feature set reduction algorithm using rough set theory, which can be used to discover data dependencies and reduce the number of attributes in a data set without requiring any additional information. Moreover, Jothi and Inbarani [8] proposed to apply a soft set based algorithm for unsupervised feature reduction.

At the highest level, our study also tries to select a set of features, and thus belongs to the big area of feature selection. However, the objective in our study is critically different from those in the existing work. While we will present the formal problem formulation in the next section, instead of aiming at classification or clustering, our study focuses on finding a minimum set of features so that every image can be uniquely indexed with a combination of features.

3 Feature Extraction and Problem Formulation

In this section, we first describe how we can extract candidate features from a set of images without any domain knowledge. Then, we present our problem of feature selection.

3.1 Candidate Feature Extraction Given a large set of photos, without any additional information (i.e., in an unsupervised manner), how can we extract meaningful features/dimesions automatically?

As mentioned above, each photo may contain multiple meaningful semantics. Each semantically meaningful region can be roughly detected by existing image segmentation techniques. For instance, Wang *et al.* [21] proposed a segmentation method that partitions each image into blocks with 4×4 pixels and extracts a feature vector for each block (color feature and texture feature). Then, the k-means algorithm [6] is used to cluster the feature vectors into several clusters such that every cluster contains multiple blocks and form a meaningful region.

We borrow the idea of MIML (Multi-Instance Multi-Label learning) [14, 25], which represent each data object by multiple instances. After segmentation, we represent each image as a bag with multiple regions, that is, $I_i = \{\mathbf{x}_{ij} | j = 1, \cdots, N_i\}$, where each region is described as an *instance* $\mathbf{x}_{ij}$ which is the average feature vector over all its blocks, and N_i is the total number of instances in image I_i. $X = \{\mathbf{x}_{11}, \cdots, \mathbf{x}_{1N_1}, \cdots, \mathbf{x}_{n1}, \cdots, \mathbf{x}_{nN_n}\}$ denotes the set of instances collected from all n images. Therefore, X can be treated as the set of extracted candidate dimensions, further denoted as $D = \{\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_{|X|}\}$, where each $\mathbf{d}_j$ has a meaningful semantic.

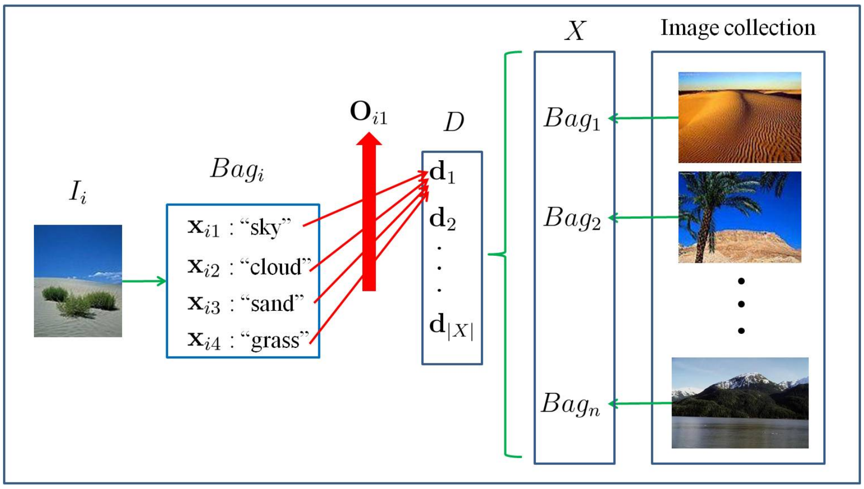


Figure 2: Feature Extraction

We then map each image onto the extracted dimension set D by finding the minimum Euclidean distance from its instances $I_i = \{\mathbf{x}_{ij} | j = 1, \cdots, N_i\}$ to each dimension in D. Formally, each image is represented by the distance feature vector as $[\phi_1, \cdots, \phi_{|D|}]$, where the q-th feature value is calculated by

$$\phi_q(I_i) = \min_{j=1,\cdots,N_i} ((\mathbf{x}_{ij} - \mathbf{d}_q)^\top (\mathbf{x}_{ij} - \mathbf{d}_q))^{\frac{1}{2}}.$$

Thereafter, by comparing the feature value with a predefined distance threshold θ, we determine if an image contains the corresponding dimension. Specifically, if $\phi_{iq} \leq \theta$, we set $\mathbf{O}_{iq} = 1$, which means image I_i has the q-th semantic content, otherwise 0. As such, a new feature space $\mathbf{O} \in \{0, 1\}^{n \times |D|}$ is generated. The whole process is illustrated in Figure 2.

3.2 Problem Formulation Given the candidate dimension set D and the corresponding presentation for all images $\mathbf{O} \in \{0, 1\}^{n \times |D|}$, suppose that each image can be uniquely represented by $\mathbf{O}$ (cases beyond this assumption are discussed in the next section), which means $\forall p, q \in [1, n], p \neq q, \mathbf{o}_p \neq \mathbf{o}_q$, our goal is to select a minimum subset of dimensions $D' \subseteq D$, such that each image can still be uniquely represented by $\mathbf{O}' \in \{0, 1\}^{n \times |D'|}$. The problem can be formulated as the following optimization problem.

$$\begin{aligned} & \arg\min_{D' \subseteq D} && |D'| \\ & \quad s.t. && \mathbf{o}'_p \neq \mathbf{o}'_q, \forall p, q \in [1, n], p \neq q \\ \text{(3.1)} & && \mathbf{o}'_i \neq \mathbf{0}, \forall i \in [1, n], \end{aligned}$$

where the second constraint requires that every image should be covered by selected dimensions.

	d_1	d_2	d_3	d_4
$\mathbf{o}_1$	0	1	1	1
$\mathbf{o}_2$	1	1	0	0
$\mathbf{o}_3$	0	1	1	0

Table 1: a photo collection with 3 images on 4 dimensions

The optimization problem in Equation 3.1 is equivalent to the following optimization problem.

$$\begin{aligned} & \arg\max_{D' \subseteq D} && N_{dif}^{D'} \\ & \quad s.t. && N_{dif}^{D^0} < N_{dif}^{D'}, \forall D^0 \subseteq D, |D^0| < |D'|, \\ \text{(3.2)} & && \mathbf{o}'_i \neq \mathbf{0}, \forall i \in [1, n], \end{aligned}$$

where $N_{dif}^{D'}$ indicates the number of different pairs of photos generated upon the dimension set D'. Taking the photo collection in Table 1 as an example, dimension set $\{d_1, d_2\}$ generates $N_{dif}^{\{d_1,d_2\}} = 2$ different pairs $\langle \mathbf{o}_1, \mathbf{o}_2 \rangle$ and $\langle \mathbf{o}_2, \mathbf{o}_3 \rangle$. Both $\{d_1, d_4\}$ and $\{d_1, d_3, d_4\}$ generate 3 different pairs which is the maximum number, however $\mathbf{o}_3$ is not covered by any dimension selected in $\{d_1, d_4\}$.

Theorem 3.1. *The optimization problem in Equation 3.1 is equivalent to the optimization problem in Equation 3.2.*

Proof. As stated, we assume that each image is uniquely represented by $\mathbf{O} \in \{0, 1\}^{n \times |D|}$, which means $\forall p, q \in [1, n], p \neq q, \mathbf{o_p} \neq \mathbf{o}_q$. Suppose the solution set for Equation 3.1 is $\mathcal{A}$ and the solution set for Equation 3.2 is $\mathcal{B}$.

1. For any solution $A \in \mathcal{A}$, it provides a minimum subset that every image is uniquely represented. The number of different pairs generated by A is $\binom{n}{2}$, which is the maximum. The size of $|A|$ is minimum, and thus $A \in \mathcal{B}$. Therefore, $\mathcal{A} \subseteq \mathcal{B}$.

2. For any solution $B \in \mathcal{B}$, since we assume that each image is uniquely represented by $\mathbf{O} \in \{0,1\}^{n\times|D|}$, $N^B_{dif} = \binom{n}{2}$. Moreover, there exists no other solution B' such that $|B'| < |B|$. That means B is a minimum subset of D that can uniquely identify each image, and thus $B \in \mathcal{A}$. Therefore, $\mathcal{B} \subseteq \mathcal{A}$.

In summary, $\mathcal{A} = \mathcal{B}$.

4 Submodularity and a Greedy Method

Now we prove the optimization problem in Equation 3.2 is *submodular* [11], that is, it exhibits a diminishing returns property: adding a dimension when there are only a few dimensions generates more different pairs than adding it after gathering many dimensions. Let the number of different pairs generated upon dimension set D be $\mathcal{R}(D) = N^D_{dif}$. We first give the following lemma.

Lemma 4.1. *For all dimension sets $A \subseteq B \subseteq D$ and dimension $d \in D \setminus B$,*

$$\mathcal{R}(A \cup \{d\}) - \mathcal{R}(A) \geq \mathcal{R}(B \cup \{d\}) - \mathcal{R}(B).$$

Proof. Given a new dimension d are determined, we can calculate the possible different pairs generated by it. For instance, d_3 in Table 1 will generate two different pairs: $\langle \mathbf{o}_1, \mathbf{o}_2\rangle$ and $\langle \mathbf{o}_2, \mathbf{o}_3\rangle$.

Let the set of different pairs generated by dimension set $\{d\}$ be $S_{\{d\}}$ and that of D be S_D. Obviously, $\mathcal{R}(A\cup\{d\}) - \mathcal{R}(A) = |S_{\{d\}} - S_A \cap S_{\{d\}}|$ and $\mathcal{R}(B \cup \{d\}) - \mathcal{R}(B) = |S_{\{d\}} - S_B \cap S_{\{d\}}| = |S_{\{d\}} - (S_A \cup S_{B-A}) \cap S_{\{d\}}| = |S_{\{d\}} - S_A \cap S_{\{d\}} - S_{B-A} \cap S_{\{d\}}| \leq |S_{\{d\}} - S_A \cap S_{\{d\}}| = \mathcal{R}(A \cup \{d\}) - \mathcal{R}(A)$.

Using Lemma 4.1, we have the following result.

Theorem 4.1. *$\mathcal{R}(D)$ is submodular.*

Proof.
1. $\mathcal{R}(\emptyset) = 0$, which is trivial because all images carry the same dimension values without adding any dimension;

2. Since adding more dimensions does not reduce the number of different pairs, $\mathcal{R}$ is nondecreasing, i.e., $\mathcal{R}(A) \leq \mathcal{R}(B)$ for all $A \subseteq B \subseteq D$;

3. According to Lemma 4.1.

Therefore, $N^D_{dif} = \mathcal{R}(D)$ is submodular.

Algorithm 1 Greedy Algorithm

```
Input: $D$: candidate dimension set
    $\mathbf{O} \in \{0,1\}^{n\times|D|}$: Image representations over $D$
Output: $D'$: selected dimensions
$D' = \emptyset$
Initialize root as $\{1, 2, \cdots, n\}$
while any leaf has more than one image do
    for each $d \in D \setminus D'$ do
        calculate added pairs by $\mathcal{R}(D' \cup \{d\}) - \mathcal{R}(D')$
    end for
    $d_q = \arg\max_{d\in D\setminus D'} \mathcal{R}(D' \cup \{d\}) - \mathcal{R}(D')$
    for each leaf do
        move each image $i$ with $\mathbf{o}_{iq} = 0$ to the left child
        move each image $j$ with $\mathbf{o}_{jq} = 1$ to the right child
    end for
    $D' \leftarrow D' \cup \{d_q\}$
end while
if the most left image $i$ is represented as $\mathbf{o}' = \mathbf{0}$ then
    randomly pick $d_q \in D \setminus D'$ with $\mathbf{o}_{iq} = 1$
    $D' \leftarrow D' \cup \{d_q\}$
end if
```

Maximizing submodular functions in general is NP-hard [9]. A commonly used heuristic is *greedy algorithms*. In our case, the greedy algorithm begins with the empty dimension set $D_0 = \emptyset$, and iteratively, in step t, adds dimension d_t that maximizes the increased number of different pairs

$$d_t = \arg\max_{d\in D\setminus D'_{t-1}} \mathcal{R}(D'_{t-1} \cup \{d_t\}) - \mathcal{R}(D'_{t-1}).$$

Since all dimensions are equally considered, according to the theorem stated in [18], if the greedy algorithm stops when M dimensions are selected, $\mathcal{R}(D_M) \geq (1 - 1/e)\max_{D'\subseteq D, |D'|=M} \mathcal{R}(D')$, which means this simple greedy algorithm is near-optimal.

Therefore, we propose a greedy algorithm summarized in Algorithm 1.

4.1 Analysis Let the height of the binary tree formed by Algorithm 1 be T. We have $\log_2 n \leq T \ll n$, where n is the total number of images. For each level h, there will be at most $|D|$ dimensions to be selected and for each dimension, we can calculate how many pairs will be added for each tree node in this level within one step. The total number of nodes except leaves in the tree is at most $2^0 + 2^1 + 2^2 + \cdots + 2^{T-1} = 2^T - 1$. Therefore, in the worst case, the total running time is $O(|D| \cdot 2^T)$. In each level there are at most n nodes. Therefore, the total running time is between $O(n|D|)$ and $O(n^2|D|)$.

4.2 Extensions For real-world image collections, it is possible that all extracted dimensions cannot uniquely identify each image. In such a case, we change the objective to finding a minimum subset of dimensions that at most α images have the same dimension values, which is formulated as

$$\begin{aligned} \arg\min_{D' \subseteq D} \quad & |D'| \\ s.t. \quad & |S|_{max} \leq \alpha, S = \{\mathbf{o}' | \forall i, j, \mathbf{o}'_i = \mathbf{o}'_j\} \\ & \mathbf{o}'_i \neq \mathbf{0}, \forall i \in [1, n]. \end{aligned}$$

Algorithm 1 can be adopted while Line 3 changes to "any leaf has more than α images".

Moreover, the minimum number of dimensions needed for Problem 3.1 is roughly $\log_2 n$, where a complete binary tree with hight $\log_2 n$ is formed. For each level/dimension, $n/2$ photos have the corresponding dimension value. Some people may like to search photos in this way, e.g., "give me all photos contain the sunset". If the content *sunset* is contained in $n/2$ photos while n is very large, it is still not convenient for people to view all output photos. In such a scenario, we can constrain that for each dimension at most δ photos contain it and δ should be within the capacity of a user. The problem will be changed to find a minimum subset of dimensions such that no two images have the same dimension values and each dimension is contained in at most δ images. That is

$$\begin{aligned} \arg\min_{D' \subseteq D} \quad & |D'| \\ s.t. \quad & \mathbf{o}'_p \neq \mathbf{o}'_q, \forall p, q \in [1, n], p \neq q \\ & N_j \leq \delta, \forall j \in [1, |D'|] \\ & \mathbf{o}'_i \neq \mathbf{0}, \forall i \in [1, n], \end{aligned}$$

where N_j denotes the number of images that contain the j-th dimension. Algorithm 1 can be adopted after a preprocessing: scan all candidate dimensions and delete those contained in more than δ images.

5 Experiments

To validate the effectiveness and efficiency of our proposed method, we conduct experiments on several real-world photo/image collections: the *kdd2012* conference photo collection[1], which has 1,503 images in total, the *Image* [23] data set with 2,000 images, the *event* [13] data set with 1,579 images, *utoronto* [16], a subset of the ImageNet data set with 1,998 useful images, and *avd*, a subset of COREL data set with animals, vehicles and distractors [12] including 3,979 images.

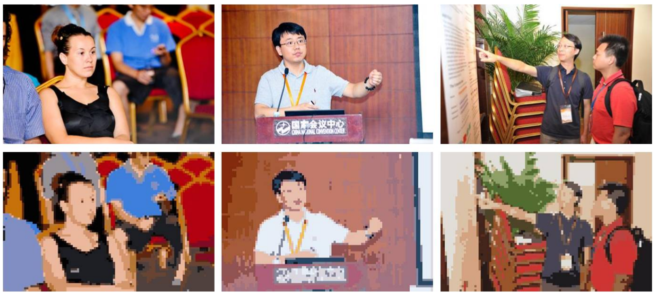


Figure 3: Segmentation example on dataset *kdd2012*

5.1 Setup All our experiments are conducted on a 64bit-Windows sever with a 2.5Hz CPU and 8G main memory. First, we resize all original images into smaller ones with size no larger than 400 × 400 pixels. It takes about ten minutes to resize 2,000 images. Following the image segmentation technique described in [21], each image is segmented into at least 2 and at most 16 regions. Note that 6 features are used for segmentation: 3 LUV color features averaged over the 4×4 block and 3 wavelet texture features in the block. It takes about 1.5 hours to segment 2,000 images. Samples are shown in Figure 3, where the upper row shows the original images and the lower row shows the segmented images with each region represented by the same color.

After segmentation, each region may contain multiple 4×4 blocks. We represent each region as an instance by 25 features, 6 features of the cluster center found in the segmentation process, 3 RGB features averaged over all its blocks, 3 averaged HSV color features, 6 averaged color moment features, 3 averaged wavelet texture features, and the total number of blocks within this region. Therefore, each image is represented as a bag of at most 16 instances, each with 25 features.

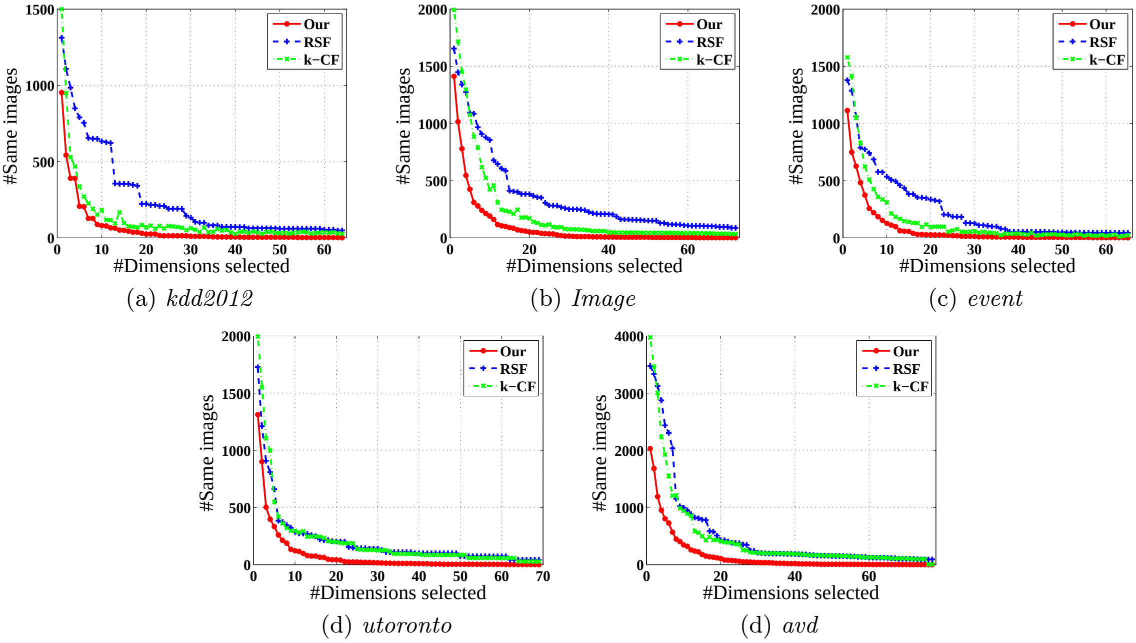
Then, we map each image onto dimensions collected from all images by finding the minimum Euclidean distance from its own instances to each dimension. For convenience, instead of using the distance value, each image is represented by a weight vector $[\psi_1, \cdots, \psi_{|D|}]$, where the q-th value is calculated by $\psi_q = \exp(-\phi_q(I_i)^2/\sigma^2)$, and σ is the mean distance following the method in [26]. In the experiments, if $\psi_q \geq \theta$, where θ is a threshold parameter and $\theta \in \{0.95, 0.90, 0.85, 0.80, 0.75, 0.70\}$, the image contains the q-th dimension and we set $\mathbf{o}_{iq} = 1$, otherwise 0.

To the best of our knowledge, there is no existing method that is exactly for our problem. We propose some simple baselines for comparison. Two baselines are searching by random selection. RSF (Random Selection Forward searching) begins with an empty set and randomly selects a feature at each iteration until each

[1] http://www.flickr.com/photos/sigkdd2012

Data set	#Images	#Candidates	Method			CPU time (sec)		
			Our	RSF	RSB	Our	RSF	RSB
kdd2012	1, 503	20, 251	64	935	20, 098	54.08	61.55	211.17
Image	2, 000	24, 923	72	612	24, 757	105.28	119.25	420.07
event	1, 579	21, 992	65	463	21, 829	64.89	70.18	293.73
utoronto	1, 998	25, 116	69	812	24, 976	125.37	203.34	314.07
avd	3, 979	56, 900	77	1, 834	56, 561	569.43	613.27	2,021.5

Table 2: Performance comparison when $\theta = 0.70$



(a) *kdd2012* (b) *Image* (c) *event*

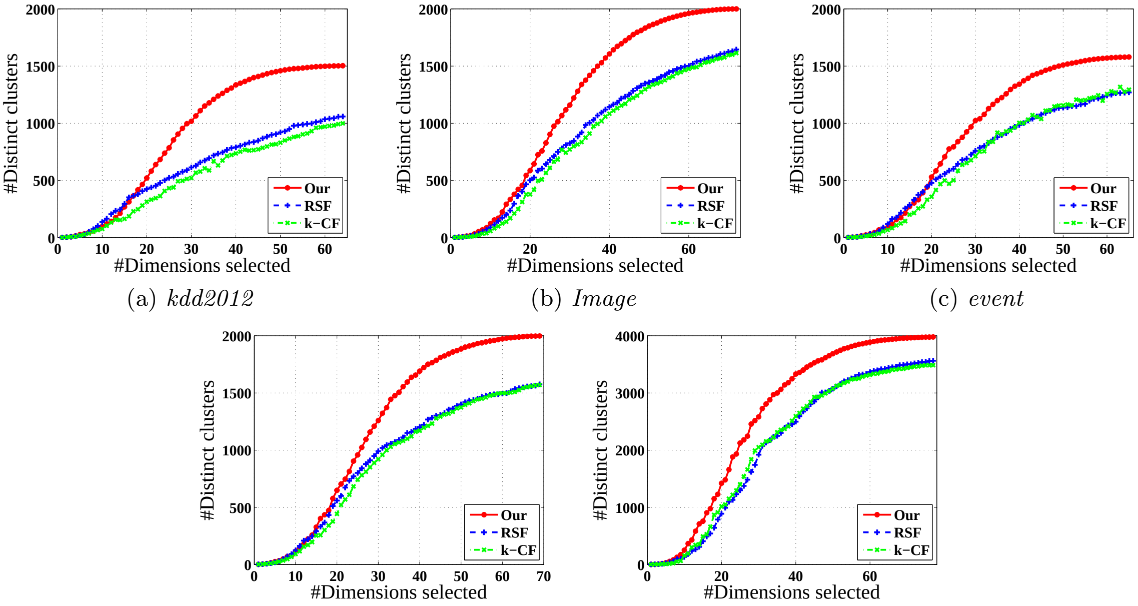
(d) *utoronto* (d) *avd*

Figure 4: Maximum #images with same dimension values as selected dimensions increases when $\theta = 0.70$

image is uniquely represented and RSB (Random Selection Backward searching) begins with the whole set and randomly removes a feature at each iteration until at least one image cannot be uniquely identified. The other baseline is k-CF (k-means Clustering Forward searching) that adopts the k-means clustering algorithm [6] to do clustering over all features and k is increased by 1 in each iteration from $k_0 = 1$ until each image can be uniquely identified.

5.2 Performance In this section, we set the parameter $\theta = 0.70$ and compare the performance of our method to all proposed baselines in four aspects 1) the total number of features selected; 2) efficiency (CPU time); 3) the maximum number of images that have the same dimension values as the number of dimensions selected increases (only for forward searching method); 4) the number of different image sets (a set contains images that have the same dimension values on current selected dimensions) generated as the number of dimensions selected increases (only for forward searching method). Note that k-means clustering is very slow especially when k increases to a large number due to the huge number of candidate features as the CPU time shown in Table 2. For example, it takes about 37.5 hours for k-CF to select only up to 75 features for data set *event*. Therefore, we compare to k-CF only in the 3rd and 4th aspects. For RSF and RSB, the best result of 10 trials for each data set is reported.

The total number of candidate dimensions extracted for each data set is summarized in Table 2, from which we can see that the total number of features selected by our method is much less than *RSF* and *RSB*. At the same time, Table 2 demonstrates that the efficiency of our proposed feature selection method is even slightly better than random searching.



(a) *kdd2012* (b) *Image* (c) *event*

(d) *utoronto* (d) *avd*

Figure 5: #Distinct image sets as selected dimensions increases when $\theta = 0.70$

Data Method	*kdd2012* Our	 RSF	*Image* Our	 RSF	*event* Our	 RSF	*utoronto* Our	 RSF	*avd* Our	 RSF
$\theta = 0.75$	71	675	87	1,735	76	531	79	876	100	3,230
$\theta = 0.80$	81	936	110	3,529	98	1,344	96	1,255	133	4,612
$\theta = 0.85$	100	1,899	175	10,002	144	4,022	129	4,455	206	8,112
$\theta = 0.90$	158	3,385	-	-	264	6,605	-	-	391	16,433
$\theta = 0.95$	367	8344	862	15,750	659	11,014	540	14,003	1,062	38,319

Table 3: Effect of θ ('-' means no solution)

Figure 4 shows the maximum number of images that have the same dimension values on current selected dimensions as the number of dimensions selected increases and Figure 5 shows the number of different image sets generated as the number of selected dimensions increases. The best result for RSF is plotted too. Our method clearly outperforms the baselines.

5.3 Effect of parameter θ In this section, we study the effect of parameter θ. Since RSB is much worse than RBF due to the possible reason that it may randomly remove some essential features at the very beginning, we only compare our method with RSF in this subsection. The larger θ, the less 1's generated in $\mathbf{O}$. Therefore, a larger number of features selected as θ increases, as shown in Table 3. Our method is much better than RSF for different values of θ.

5.4 Illustration of selected dimensions For selected dimensions, we can easily map them back to their specific bag instances and then locate the corresponding regions in the original images, which, for example, can be shown as circled in red line in Figure 1 with semantic labels. With less than 100 dimensions selected, it is much easier for people to give manual labels compared with labeling millions of images.

6 Conclusions

To automatically index a large photo collection without domain knowledge, we first propose a simple strategy to automatically extract meaningful semantics from original photos as dimensions. Upon an even larger set of dimensions collected, we then propose an efficient unsupervised feature/dimension selection method to select a sufficient dimension subset to represent each photo within the collection uniquely for indexing. Experiments on a board of variety real-world photo/image collections demonstrate both the efficiency and effectiveness of our proposed method.

7 Acknowledgements

This research is supported in part by an NSERC Discovery Grant and a BCFRST NRAS Endowment Research Team Program Project. All opinions, findings, conclusions and recommendations in this paper are those of the author and do not necessarily reflect the views of the funding agencies.

References

[1] S. Alelyani, J. Tang, and H. Liu. chapter Feature Selection for Clustering: A Review. CRC Press, 2013.

[2] L. Cao, L. Luo, and T. S. Huang. Annotating photo collections by label propagation according to multiple similarity cues. In *ACM MM*, pages 121–130, Vancouver, Canada, 2008.

[3] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE PAMI*, 29(3):394–410, 2007.

[4] P. Duygulu, K. Barnard, J. de Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *ECCV*, pages 97–112, Copenhagen, Denmark, 2002.

[5] I. Fodor. A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, Center for Applied Scientific Computing, Lawrence Livermore National, 2002.

[6] J. A. Hartigan and M. A. Wong. Algorithm asi36: a k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

[7] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, pages 119–126, Toronto, Canada, 2003.

[8] G. Jothi and H. H. Inbarani. Soft set based quick reduct approach for unsupervised feature selection. In *ICACCCT*, pages 277–281, Ramanathapuram, India, 2012.

[9] S. Khuller, A. Moss, and J. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

[10] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain. Simultaneous feature selection and clustering using mixture models. *IEEE PAMI*, 26(9):1154–1166, 2004.

[11] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *ACM SIGKDD*, pages 420–429, San Jose, CA, 2007.

[12] F.-F. Li, R. VanRullen, C. Koch, and P. Perona. Rapid natural scene categorization in the near absence of attention. *PNAS*, 99(14):9596–9601, 2002.

[13] L.-J. Li and F.-F. Li. What, where and who? classifying event by scene and object recognition. In *ICCV*, 2007.

[14] Y.-F. Li, J.-H. Hu, Y. Jiang, and Z.-H. Zhou. Towards discovering what patterns trigger what labels. In *AAAI*, pages 1012–1018, Toronto, Canada, 2012.

[15] Z. Ma, Y. Yang, F. Nie, J. Uijlings, and N. Sebe. Exploiting the entire feature space with sparsity for automatic image annotation. In *ACM MM*, pages 283–292, Scottsdale, AZ, 2011.

[16] W. May, S. Fidler, A. Fazly, S. Dickinson, and S. Stevenson. Unsupervised disambiguation of image captions. In *SEM*, pages 85–89, Montreal, Canada, 2012.

[17] Y. Mori, H. Takahashi, and R. Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *MISRM*, pages 405–409, Orlando, FL, 1999.

[18] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14, 1978.

[19] D. Ritendra, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[20] C. Velayutham and K. Thangavel. A novel entropy based unsupervised feature selection algorithm using rough set theory. In *ICAESM*, pages 156–161, Tamil Nadu, India, 2012.

[21] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE PAMI*, 23:947–963, 2001.

[22] C. C. Yen, L. C. Chen, and S. D. Lin. Unsupervised feature selection: minimize information redundancy of features. In *TAAI*, pages 247–254, Hsinchu, Taiwan, 2010.

[23] M.-L. Zhang and Z.-H. Zhou. Ml-knn: a lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007.

[24] T. Zhang, D. Tao, X. Li, and J. Yang. Patch alignment for dimensionality reduction. *IEEE TKDE*, 21(9):1299–1313, 2009.

[25] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.

[26] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919, Washington, DC, 2003.