Online Visual Vocabulary Pruning Using Pairwise Constraints

Pavan K. Mallapragada[1], Rong Jin[1] and Anil K. Jain[1,2]
[1]Dept. Computer Science and Engineering, Michigan St. Univ., East Lansing, MI-48824.
[2]Dept. Brain and Cognitive Engineering, Korea Univ.,Anam-dong, Seoul 136-713, Korea.
{pavanm,rongjin,jain}@cse.msu.edu

Abstract

Given a pair of images represented using bag-of-visual-words and a label corresponding to whether the images are "related"(must-link constraint) or "unrelated" (cannot-link constraint), we address the problem of selecting a subset of visual words that are salient in explaining the relation between the image pair. In particular, a subset of features is selected such that the distance computed using these features satisfies the given pairwise constraints. An efficient online feature selection algorithm is presented based on the dual-gradient descent approach. Side information in the form of pair-wise constraints is incorporated into the feature selection stage, providing the user with flexibility to use an unsupervised or semi-supervised algorithm at a later stage. Correlated subsets of visual words, usually resulting from hierarchical quantization process (called groups), are exploited to select a significantly smaller vocabulary. A group-LASSO regularizer is used to drive as many feature weights to zero as possible. We evaluate the quality of the pruned vocabulary by clustering the data using the resulting feature subset. Experiments on PASCAL VOC 2007 dataset using 5000 visual keywords, resulted in around 80% reduction in the number of keywords, with little or no loss in performance.

1. Introduction

Large amounts of multimedia data are being generated every day. An image hosting website such as Flickr receives around 10 images per second, or nearly 38K images per hour. It is computationally prohibitive to perform batch mode learning even on a week's collection of data, let alone running the learning algorithm every time new data is available. Online algorithms provide an efficient way to continuously learn from examples as and when they become available. In this paper, we address the problem of feature selection in an online setting. In particular, we aim to perform feature selection on images represented as a *bag of visual words*.

Representing images using a bag of visual words [4] has received significant attention. In this approach, each image is represented as a distribution over a set of visual vocabulary. The vocabulary itself is a set of prototypes obtained by clustering the set of key points (e.g., using SIFT operator) pooled from a collection of training images. Several applications such as image clustering [1], large scale image [14] and video retrieval [16] have shown this method to be promising in both performance and scalability.

Recent studies have shown that the choice of vocabulary size can have a significant impact on the performance of learning algorithms [18]. A small vocabulary size may result in a feature space not rich enough to capture the variability in the images, while a large vocabulary may cause two keypoints that are similar to be mapped to two different visual words leading to suboptimal performance. Further, a large number of visual words results in the well known problems of curse of dimensionality, complex hypothesis spaces and large computational requirements. Feature selection, or vocabulary pruning, is an important step in text retrieval that retains only a few important words needed for subsequent classification or clustering [6].

Visual vocabularies are usually constructed using recursive partitional clustering algorithms such as bisecting K-means, resulting in a cluster hierarchy [14, 13]. This causes the visual words at the leaf nodes that are children of a common parent to be similar to each other. If one of the visual words is not informative, it is an indication that its siblings may not be informative as well. One of the basic premises of this work is to exploit what we call *visual synonyms* for feature selection. Visual synonyms are identified as the visual words sharing a common parent in the cluster hierarchy.

We propose to use pairwise constraints to encode the relationship between images. The pairwise constraints are of two types: *must-link* and *cannot-link*. A pairwise constraint is a natural way to encode a user's perceived visual similarity between a pair of images. It is easier to specify a constraint between two images than labeling them explicitly with all the objects present. Figure 1 illustrates the goal of

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

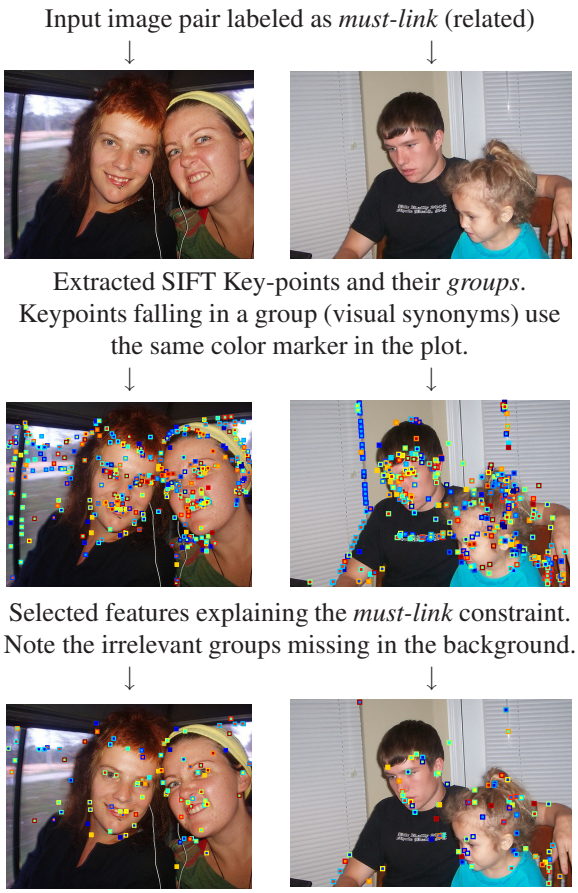

Figure 1. Illustration of SIFT key points, visual synonyms and feature selection at a group level. The first row shows a pair of images input for feature selection. Note that the key points occur in groups. Same colored marker is used for key points belonging to a group. Feature selection by proposed algorithm acts at a group level by removing the entire group of unrelated features.

the proposed approach using an image pair labeled as *must-link*. Loosely speaking, the common key points between a pair of images need to be discarded if it is a cannot-link pair, and need to be retained if they are a must-link pair.

In this paper, we propose an efficient online algorithm that takes in a set of images and the associated pairwise constraints, and selects a subset of visual words. Since each key point in an image is mapped to one of the visual words, pruning the vocabulary results in a reduction in the number of key-points in an image. The feature group information obtained from the cluster hierarchy is exploited to shrink the feature weights at a group level. The quality of the selected features is evaluated using an image clustering application.

2. Related work

Feature or variable selection is a classical problem in multivariate statistics and pattern recognition. All disciplines of learning, i.e. supervised, unsupervised, and semi-supervised usually perform some sort of feature selection. An introduction to feature or variable selection can be found in [3, 6, 8]. Feature extraction, in contrast with feature selection, results in a (non)linear combination of existing features. In applications requiring interpretability, feature selection is preferred to feature extraction.

Feature selection methods can be broadly classified into *search based* methods (e.g. Floating Search [8]), feature ranking, and *shrinkage* methods such as LASSO [17] and Group LASSO [19]. Feature selection by ranking sorts the features based on a score, such as correlation coefficient or mutual information, computed between the feature and the class labels. While feature ranking is commonly used as a baseline, features that are correlated with the labels are possibly correlated among themselves as well, resulting in the selection of a set of redundant features [6].

Search based methods are further classified into *filter* and *wrapper* methods. They operate by incrementally modifying a selected set of features by adding or deleting features one by one. These approaches are greedy in nature, and are affected by the order of adding/deleting features to/from the set. Moreover, they are computationally expensive as the learning algorithm is run every time the selected feature set is modified. Branch and bound algorithms tend to be more accurate, but are limited in their ability to handle only a small set of features due to computational reasons. Search based algorithms are batch mode, and require all the labeled data examples be present before they can be used, and are not applicable to an online setting.

Shrinkage methods are widely used for variable selection in multivariate regression. These tend to be more principled, and amenable to theoretical analysis with a predictable behavior. In general, supervised learners such as SVM, learn the weights of features. Feature selection, however differs from feature weighting. Shrinkage methods such as LASSO perform feature selection by driving as many weights to zero as possible. In a supervised setting, several algorithms such as 1-norm SVMs [21], F_∞ SVM[22] and Lasso Boosting [20], ridge regression employ shrinkage strategy. To the best of our knowledge, there is no feature selection method proposed in the literature that employs LASSO shrinkage with pairwise constraints.

Distance metric learning (DML) is another related area where the features weights are learnt from labeled examples [15, 7]. DML methods learn a quadratic distance function parameterized using a $d \times d$ weight matrix, where d is the dimensionality of the data. Online DML algorithms such as POLA [15] involve a projection step to ensure positive definiteness of the feature matrices, and are computationally expensive. Even using a diagonal weight matrix, they tend to prefer uniform feature weights, contrary to our goal. The proposed algorithm can be shown to be a general-

ization of the POLA algorithm with diagonal weight matrix, when all visual words are put in a single group.

3. Problem formulation

Let $\mathcal{D} = \{\mathcal{I}_1, \ldots, \mathcal{I}_n\}$ be the given collection of n images represented as a distribution over the visual vocabulary $\mathcal{V} = (v_1, \ldots, v_m)$ containing m visual words. Since the visual words are often generated by a recursively bisecting K-means algorithm, we can derive a group structure for the visual words. In particular, we assume the visual words are divided into s groups. Let $\mathbf{v}^g = (v_{m_{g-1}+1}, \ldots, v_{m_g})$ be the collection of visual words belonging to the g-th group, for $g = 1, \cdots, s$. Note that even when no group structure is available, our method is still applicable where each feature forms its own group, and $s = m$. Given the visual words, each image $\mathcal{I}_i$ is represented by a vector of visual word histogram, denoted by $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,m})$. Further, let $\mathbf{x}_i^g$ denote the feature sub-vector of image $\mathcal{I}_i$ corresponding to the vocabulary $\mathbf{v}^g$. Let $\mathbf{w} = (w_1, \ldots, w_m)$ denote the weights for visual words. The squared distance between two visual word histograms $\mathbf{x}$ and $\mathbf{x}'$ given the feature weights $\mathbf{w}$, denoted by $|\mathbf{x} - \mathbf{x}'|^2_{\mathbf{w}}$, is computed as

$$|\mathbf{x} - \mathbf{x}'|^2_{\mathbf{w}} = \sum_{i=1}^{m} w_i (x_i - x'_i)^2. \tag{1}$$

It is necessary that the weights are positive, $w_j \geq 0$, $j = 1, \ldots, m$, for Eq (1) to be a metric. The visual similarity between a pair of images is provided in the form of a pairwise constraint – a must-link constraint indicates two images are visually similar whereas a cannot-link constraint indicates two images are visually different. Let $\mathcal{T} = \{(\mathbf{x}_t, \mathbf{x}'_t, y_t), t = 1, \ldots, T\}$ denote the collection of pairwise constraints that will be used for learning the weights, where $\mathbf{x}_t$ and $\mathbf{x}'_t$ are visual word histograms corresponding to two images, and $y_t = \pm 1$, where $+1$ indicates the two images are visually similar and -1 otherwise.

The goal is to learn weights $\mathbf{w}$ for the visual words such that the following criteria are met:

1. The distance between the two images computed using Eq (1) reflects the visual similarity between the images.

2. Select a small subset of features by driving as many entries in the vector $\mathbf{w}$ to 0 as possible.

For a given a pairwise constraint $(\mathbf{x}_t, \mathbf{x}'_t, y_t)$, if $y_t = 1$, the distance between $\mathbf{x}_t$ and $\mathbf{x}'_t$ must be less than a threshold b (which can either be learnt, or specified by the user). On the other hand, if $y_t = -1$, the distance computed using the selected features must be greater than b. We define a loss function measuring the error made by a weight vector $\mathbf{w}$ on an example pair $\mathbf{x}_t, \mathbf{x}'_t$ with true label y_t as follows:

$$\ell(\mathbf{w}; \mathbf{x}_t, \mathbf{x}'_t, y_t) = \max\left(0, 1 - y_t(b - |\mathbf{x}_t - \mathbf{x}'_t|^2_{\mathbf{w}})\right). \tag{2}$$

In order to encode the hierarchical structure among visual words, we introduce a *mixed norm* for weight vector w, denoted by $\|w\|_{1,2}$, that is defined as follows:

$$\|\mathbf{w}\|_{1,2} = \sum_{g=1}^{s} \sqrt{\sum_{j=m_{g-1}+1}^{m_g} w_j^2} \tag{3}$$

where $m_{g-1} + 1$ is the index of the first element in the g-th group. The above norm is introduced to enforce feature selection at a group level, i.e., if multiple visual words within a group are assigned small weight, the entire group of visual words may be deemed irrelevant and can be discarded. This mixed norm is often referred to as group-lasso or the $L_{1,2}$ norm and is widely used for feature selection [11].

Using the norm defined in Eq(3) as the regularizer and the loss defined in Eq (2), the feature weights can be learnt by minimizing the following objective function:

$$\min_{w \in \mathbb{R}^m_+} \|\mathbf{w}\|^2_{1,2} + \lambda \sum_{t=1}^{T} \ell(\mathbf{w}; \mathbf{x}_t, \mathbf{x}'_t, y_t) \tag{4}$$

where $b > 0$ is a predefined constant. The goal of this work is to present an online algorithm to minimize Eq (4). Online algorithms are computationally efficient since they learn with only one example at each time.

4. Online algorithm using projections

Our online feature selection algorithm is presented in Section 4.1, followed by a theoretical analysis in Section 4.2. For conciseness, we define $\Delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}'_t$ and use the notation $\ell_t(\mathbf{w})$ to denote the loss $\ell(\mathbf{w}; \mathbf{x}_t, \mathbf{x}'_t, y_t)$ at the t-th round of learning. Algorithm 1 summarizes the general online feature selection framework.

4.1. Algorithm

Step 1. Given a pair of images $\mathbf{x}_t$, $\mathbf{x}'_t$, predict whether they are in the same cluster using the existing weight vector $\mathbf{w}_t$ and Eq (1). Observe the true output y_t, and compute the loss $\ell_t(\mathbf{w})$.

Step 2. For convenience, define temporary weights $\boldsymbol{\theta}_t = (\boldsymbol{\theta}_t^1, \boldsymbol{\theta}_t^2, \cdots, \boldsymbol{\theta}_t^s)$, where $\boldsymbol{\theta}^g$ is the subvector corresponding to group g, as follows:

$$\boldsymbol{\theta}_t^g = \|\mathbf{w}_t\|_{1,2} \frac{\mathbf{w}_t^g}{\|\mathbf{w}_t^g\|_2}, g = 1, \cdots s. \tag{5}$$

Step 3. Since the gradient of the loss function $\nabla_{\mathbf{w}} \ell_t(\mathbf{w})$ indicates the direction for updating weights, the temporary weights are updated using the following rule

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \lambda \nabla_{\mathbf{w}} \ell_t(\mathbf{w}_t) = \boldsymbol{\theta}_t - \lambda y_t \Delta\mathbf{x}_t \tag{6}$$

where λ a prespecified stepsize or the learning rate.

Step 4. To perform group level feature selection, each group is weighted by a factor that depends on the norm of the feature weights within the group. In particular, we compute the weight of each group using a soft-max function. That is, the weight of group g, q^g is obtained as,

$$q^g = \frac{\exp(\|\boldsymbol{\theta}^g\|_2^2/2\mu)}{\sum_{k=1}^{s}\exp(\|\boldsymbol{\theta}^k\|_2^2/2\mu)}. \quad (7)$$

Note that due to normalization, smaller values of $\|\theta^g\|_2^2$ result in near zero q^g values. The smoothing parameter μ in the softmax function controls the distribution of group weights. For a large μ all groups are weighted equally irrespective of their utility, and as μ goes to zero, only one group whose weights have the largest norm is selected.

Step 5. Since the weights for the features must be positive, replace all the negative elements in $\boldsymbol{\theta}$ with 0. Compute the weight vector $\mathbf{w}_{t+1}$from the temporary weights $\boldsymbol{\theta}$ as follows:

$$\mathbf{w}_{t+1}^g = q^g \frac{\boldsymbol{\theta}_t^g}{\|\boldsymbol{\theta}_t^g\|_2^2}, \quad g = 1, \cdots, s. \quad (8)$$

where $\mathbf{w}_{t+1}^g$ is the g-th subvector of $\mathbf{w}$ corresponding to the vocabulary of g-th group, $\mathbf{v}^g$.

Eq (8) gives the solution for the weight vector $\mathbf{w}_{t+1}$ for the next iteration. Steps 1-5 are repeated as each example pair becomes available. The features corresponding to non-zero weights in $\mathbf{w}$ are considered relevant, and form the selected subset of features.

4.2. Theoretical analysis

Potential based gradient descent [2, Chapters 2,11] is an online learning framework that generalizes several classical algorithms like Widrow-Hoff, Winnow and the recent Exponentiated Gradient (EG) algorithm [2]. However, classical analysis presented in [2] is applicable only to potential functions that are strictly convex. The potential generating the $L_{1,2}$ norm considered in the proposed approach is not strictly convex. In this section, we propose a smooth approximation to the mixed norm $\|\cdot\|_{1,2}^2$ for weight estimation. We begin with the following lemma that allows us to rewrite $\|w\|_{1,2}^2$ as a variational minimization problem.

Lemma 1. *The group-LASSO norm can be shown to be the exact minimizer of the variational problem*

$$\frac{1}{2}\|\mathbf{w}\|_{1,2}^2 = \frac{1}{2}\min_{p\in\mathbb{R}_+^s}\left\{\sum_{g=1}^{s}\frac{\|\mathbf{w}^g\|_2^2}{p_g} : \sum_{g=1}^{s} p_g = 1\right\} \quad (9)$$

Proof. See [10]. □

A smoothing term is now introduced to ensure that the norm $\|\cdot\|_{1,2}^2$ is strictly convex. The smooth norm $\Phi(\mathbf{x},\mu)$ is defined as follows:

$$\Phi(\mathbf{w};\mu) = \min_{p\in\mathbb{R}_+^s}\left\{\sum_{g=1}^{s}\frac{\|\mathbf{w}^g\|_2^2}{2p_g} - \mu H(p) : \sum_{g=1}^{s} p_g = 1\right\} \quad (10)$$

where $H(p)$ is the Shannon entropy defined as $H(p) = -\sum_{g=1}^{s} p_g \ln p_g$, and μ is the *smoothness* parameter. Also, we have $\frac{1}{2}\|\mathbf{w}\|_{1,2}^2 - \mu\ln s \le \Phi(\mathbf{w};\mu) \le \frac{1}{2}\|\mathbf{w}\|_{1,2}^2$.

Lemma 2. *The approximate potential function $\Phi(\mathbf{w},\mu)$ is a strictly convex function.*

Proof. See [10]. □

The following lemma shows the convex conjugate of the smooth norm, which is shown to be strictly convex in the subsequent lemma.

Lemma 3. *The convex conjugate of the smooth norm $\Phi(\mathbf{w},\mu)$, denoted by $\Phi^*(w,\mu)$ is computed as*

$$\Phi^*(\mathbf{w},\mu) = \mu\ln\left(\sum_{g=1}^{s}\exp\left[\frac{\|\mathbf{w}^g\|_2^2}{2\mu}\right]\right)$$

Proof. See [10]. □

Note that as μ goes to zero, $\Phi^*(\mathbf{w},\mu)$ becomes $\max_{1\le g\le s}|\mathbf{w}^g|_2^2$, which is the square of the mixture of the L_∞ and L_2 norm. This is interesting since L_∞ norm is the dual of L_1 norm. Lemma 4 below shows that $\Phi^*(\mathbf{w},\mu)$ is a strict convex function

Lemma 4. *The Hessian matrix of $\Phi^*(\mathbf{w},\mu)$, denoted by $H^*(\mathbf{w},\mu)$, is positive definite i.e.,$H^*(\mathbf{w},\mu) \succ 0$. Furthermore, if $\|\mathbf{w}\|_2 \le R$, we have $H^*(\mathbf{w},\mu) \preceq (1 + R^2/\mu)I$*

Proof. See [10]. □

Given that both potential $\Phi(\mathbf{w},\mu)$, and its convex conjugate $\Phi^*(\mathbf{w},\mu)$ are strictly convex functions, the potential based gradient descent algorithm presented in [2, Chapter 11] can be used. The algorithm is described in Algorithm 2, where $\Omega = \{\mathbf{w}\in\mathbb{R}_+^m : |\mathbf{w}|_2 \le R\}$ is the domain for feature weights and $R\in\mathbb{R}$ is a predefined constant. Step 4 involves a projection of an estimate of weight vector $\mathbf{w}'_{t+1}$ into Ω, such that the Bregman divergence generated by the potential function Φ, denoted by $D_\Phi(\mathbf{w}_{t+1},\mathbf{w}_t)$ is minimized.

An online learning algorithm performs a weight update whenever it makes a mistake in its prediction. Online learning algorithms are characterized by mistake bounds [2], which bound the number of mistakes made by an algorithm compared to those made by the knowledge of optimal weight vector in retrospect. The following theorem shows the mistake bound for the above online algorithm.

Algorithm 1 OnlineFeatureSelection

Initialize $\mathbf{w} \leftarrow 0, t \leftarrow 0$
for each round $t = 1, 2, \cdots$ **do**
  Observe $(\mathbf{x}_t, \mathbf{x}'_t)$ and Predict $d_t \leftarrow |\mathbf{x}_t - \mathbf{x}'_t|_{\mathbf{w}}$
  if $y_t(b - d_t) \leq 0$ **then**
    $\mathbf{w}_t \leftarrow$ DualGradientDescentStep$(\mathbf{x}_t, \mathbf{x}'_t, \mathbf{w}_{t-1})$
  end if
  $t \leftarrow t + 1$
end for

Algorithm 2 DualGradientDescentStep($\mathbf{w}_t$)

1. $\theta_t \leftarrow \nabla\Phi(\mathbf{w}_t, \mu)$
2. $\theta'_{t+1} \leftarrow \theta_t - \lambda\nabla\ell_t(\mathbf{w}_t)$
3. $\mathbf{w}'_{t+1} \leftarrow \nabla\Phi^*(\theta'_{t+1}, \mu)$
4. $\mathbf{w}_{t+1} \leftarrow \pi_\Omega(\mathbf{w}'_{t+1}, \Phi) = \arg\min_{\mathbf{w}\in\Omega} D_\Phi(\mathbf{w}, \mathbf{w}'_{t+1})$

Theorem 1. *For any convex loss function ℓ, learning rate λ, and $X_\infty = \max_t \|\Delta\mathbf{x}_t\|$ where $\Delta\mathbf{x}_t = \mathbf{x}_t - \mathbf{x}'_t$, let $\kappa = (1 + R^2/\mu)$, and $\lambda = \epsilon/(\kappa X_\infty^2)$. For all $\mathbf{u} \in \Omega$, the number of mistakes M made by the proposed algorithm is bounded as follows:*

$$M \leq \frac{1}{1-\epsilon}\left(\frac{\kappa X_\infty^2\left(\|\mathbf{u}\|^2 + \mu \ln s\right)}{2\epsilon} + \sum_{t=1}^{T}\ell_t(\mathbf{u})\right) \quad (11)$$

Proof. See [10]. □

For $\epsilon = 0.5$, the above theorem shows that the number of mistakes M made by $\mathbf{w}$ is no more than twice that of the optimal weight vector $\mathbf{u}$, and a constant depending on $\mathbf{u}$, the smoothing parameter μ and the logarithm of the number of groups s.

4.3. Implementation details

For the potential function defined in Eq (10), steps 3 and 4 of Algorithm 2 are computationally complex. In particular, the computation of $\nabla\Phi(\mathbf{w}_t, \mu)$ involves solving a non-linear optimization problem defined in Eq (10). To avoid this, we use the original $L_{1,2}$ norm instead of the smooth norm. Further, the projection step 4 in Algorithm 2 is difficult. The projection in $L_{1,2}$ is performed approximately by projecting weights in each group $\|\mathbf{w}^g\|$ into a unit ball using an L_2 norm. This results in significant computational gains, with negligible difference in the empirical evaluation. This choice results in a normalized weight vector, fixing the value of $R = 1$. The solution is given in Eq (8), and the detailed derivations of the solution are presented in [10].

5. Experimental evaluation and results

Datasets: The proposed algorithm is evaluated using the PASCAL VOC challenge 2007 dataset [5]. This dataset has 9,963 images labeled using 20 classes of objects. The training and validation set contains 5,011 images. A detailed description of the data including the number of images per class is provided in [5]. The images in the dataset have multiple labels, and hence it is not directly suitable for evaluating clustering. We ignore infrequent objects and consider only the images containing one of the 6 most popular classes in the dataset, namely, `bicycle` (243), `bird` (330), `car` (713), `cat` (337), `chair` (445), and `person` (2008). The number of samples in each class is shown in brackets. For objects with multiple labels, one of the labels is chosen randomly.

5.1. Feature extraction

SIFT (Version 4) key points [9] are extracted from each image. Each key-point is represented using a 128-dimensional feature vector. The key-points extracted from images in the training set are pooled together resulting in around 4.5 million key points. These key-points are clustered into 5,000 clusters using approximate hierarchical K-means algorithm from the FLANN library [12], with a branching factor of 20, resulting in a visual vocabulary of size 5000. Key point histograms are computed for each image in the training set. The group information of the visual vocabulary is obtained during the clustering phase by identifying all the visual words with common parents.

Experimental setup: Group-LASSO is a general norm which can be specialized to both L_2 or L_1 using appropriate group definition. If the number of groups is equal to the number of features, then Group-LASSO is equivalent to performing feature selection using an L1 norm. If all the features are put in a single group, the proposed algorithm is equivalent to the online distance metric learning algorithm POLA [15], which uses an L_2 norm as a regularizer. The performance of proposed algorithm with and without group structure ($L_{1,2}$ and L_1) is evaluated. The proposed algorithm is compared with the L_2 distance metric learning algorithm POLA. To compare the performance of the online algorithm with the batch mode algorithms, the classical Best First Search algorithm is used. However, note that batch mode algorithms assume that all examples are available a priori, and therefore usually have better performance.

For each pair of classes from the PASCAL VOC dataset, 300 randomly selected pairwise constraints are specified. The online learning algorithm is run for 10 epochs with the same 300 constraints shuffled each time. The number of constraints considered in our algorithm are orders of magnitude smaller than those considered by other approaches [15, 7], which use around 10,000 constraints.

K-means algorithm is used to cluster the images with the selected features. Different sub-tasks from the PASCAL VOC dataset are chosen based on their class labels.

Task	Classes			Proposed		Online Baseline	Batch Baseline
#	c_1	c_2	K-means	$L_{1,2}$	L_1	POLA(L_2)	BestFirst
1	bird	cat	34.25	54.77+	51.18+	41.89−	56.40+
2	bird	bicycle	46.88	45.79−	49.30+	46.55	48.83+
3	bird	chair	57.51	57.97	60.22+	50.55−	61.10+
4	bird	car	55.74	63.24+	66.99+	58.32+	66.01+
5	bird	person	79.34	78.78	76.54−	75.34−	73.47−
6	cat	bicycle	42.55	53.81+	61.73+	53.00+	59.73+
7	cat	chair	41.85	46.16+	48.04+	47.18+	55.24+
8	cat	car	55.37	55.10	55.72	55.50	55.15
9	cat	person	78.98	78.45	73.48−	74.92−	66.47−
10	bicycle	chair	62.83	64.18+	64.58+	60.73−	56.85−
11	bicycle	car	66.25	67.78+	68.97+	65.69−	66.76
12	bicycle	person	84.09	83.76	78.44−	79.96−	84.10
13	chair	car	50.35	51.03	52.02+	53.51+	55.73+
14	chair	person	73.67	76.68+	68.84−	71.87−	64.91−
15	car	person	62.65	62.73	59.97−	63.74+	57.03−
	Summary			8+/1−	9+/5−	5+/8−	7+/5−

Table 1. Performance of the proposed algorithm measured using pairwise-F1 measure. The first two columns show the target clusters, subsequent columns show the mean pairwise F_1 measure, expressed as percentage. Significant differences (paired t-test at 95% confidence) compared to the K-means algorithm are indicated by a $+$ or a $-$.

The pairwise constraints provided to the proposed feature selection algorithm are derived from the true labels of the examples. To alleviate the variability due to local minima in the K-means algorithm, it is run with ten different initializations. The cluster labels corresponding to the run with lowest value of objective function are used for evaluation. Pairwise-F measure is used as the evaluation metric.

Parameters: The proposed algorithm has two parameters – the learning rate (or step size in the sequential gradient descent) λ and the norm-smoothness parameter μ. We set $\lambda = \frac{s}{2}$ where s is the number of groups in the visual words. The value of μ is set to 1. The value of b is chosen empirically to be 4. Ideally, if $\mathbf{w}$ is unconstrained, the value of b does not matter since it compensates for a scale factor in $\mathbf{w}$. The approximation used for Step 4 of Algorithm 2 (see Section 4.3) results in $R = 1$ constraining the domain of $\mathbf{w}$ to the unit $L_{1,2}$ ball. In this case, for $b > X_\infty$, there is no $\mathbf{w}$ that satisfies any of the cannot link constraints. Therefore a choice of b must satisfy $0 < b < X_\infty$. The domain size R is not a parameter, and need not be specified.

The values of the parameter are selected using cross validation on one of the clustering tasks (`bird` vs `cat`), which are then used for all the tasks. The range of values for these parameters to perform cross validation was motivated by Theorem 1. It may appear that selecting μ close to 0 would reduce the $\mu \ln s$ term in the mistake bound in Eq (11). However, setting μ to be small results in a small λ small, rendering the updates insignificant. Moreover, too small or too large a value for λ increases the the bound significantly resulting in poor learning, and hence is not recommended.

5.2. Results and discussion

Figure 2 illustrates the features selected by the proposed algorithm on six example images from the VOC 2007 dataset. The left image in each pair shows the original set of key points extracted by the SIFT algorithm with its default settings. The right image in the pair shows the key points corresponding to the visual words, selected by the proposed algorithm. Note that in almost all the images, the key points in the background are drastically reduced . However, in the examples containing `bird` in Figure 2, the key points corresponding to the tree branches are also retained by the feature selection algorithm. In a large fraction of `bird` images in the dataset, branches co-occur with a bird. Unless a sufficient number of cannot-link constraints are given between images containing `birds` and tree-branches, corresponding key points would not be eliminated. Such cases did not occur frequently in the dataset considered.

Table 1 shows the performance of the K-means clustering algorithm on 15 clustering tasks created from the VOC dataset. Table 2 shows the mean and standard deviation of the visual words selected by the proposed algorithm and baselines. Group-LASSO based feature selection always resulted in the least number of features, followed by LASSO. The variance of Group-LASSO is higher since the features are discarded in groups of large size. In most cases, the performance drop is not significant (using paired t-test at 95% confidence). The cases where there is a significant differ-

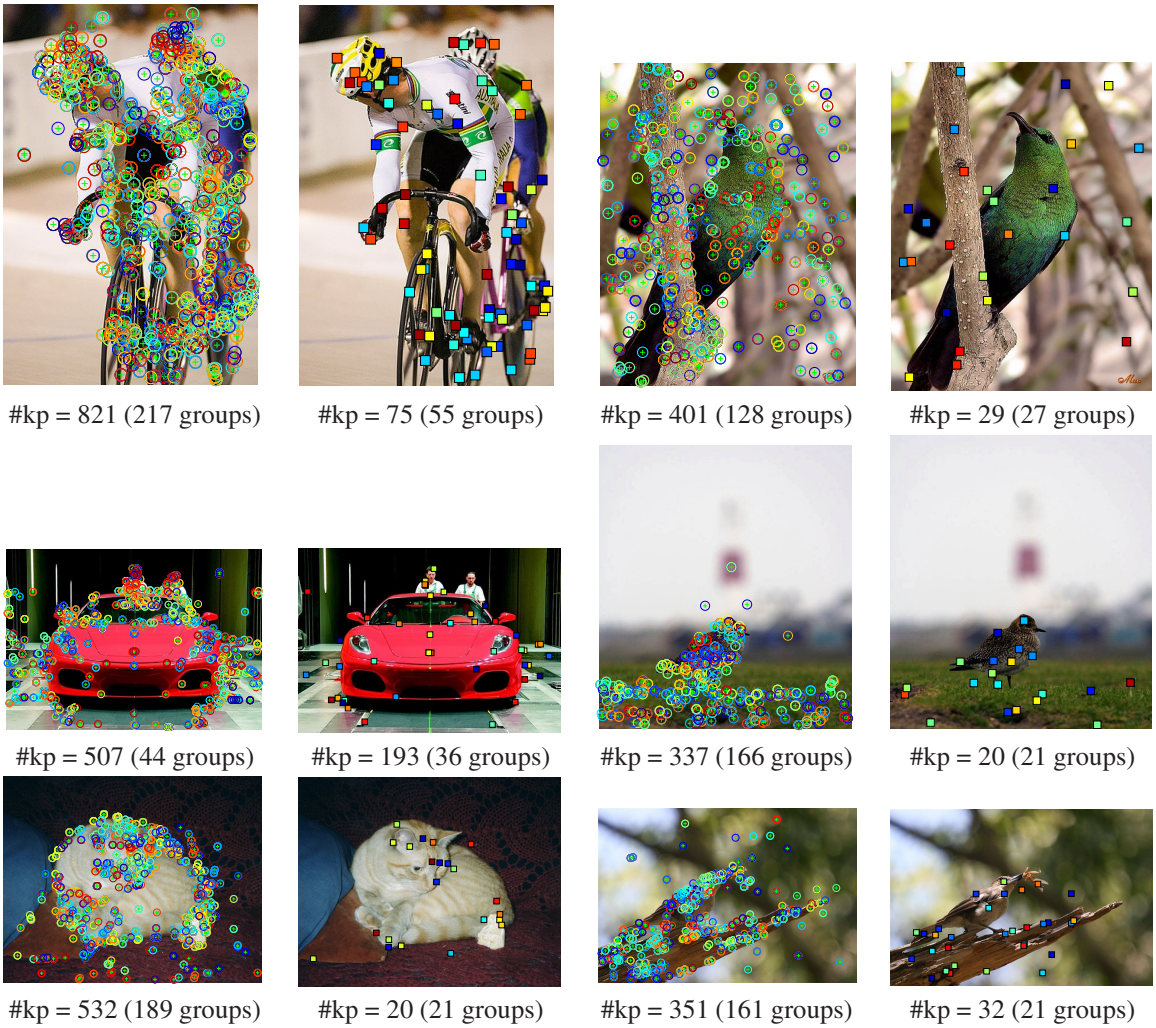

Figure 2. Feature selection using group-LASSO. In each pair of images, the left image shows the key points extracted from the original image and the right image shows the selected key points using the proposed algorithm. The number below each image indicates the number of key points (kp), and the number of groups are shown in brackets.

ence in performance are marked by $+$ or $-$ accordingly.

In three out of the five clustering tasks involving the `person` class, the performance after feature selection is lower than that of K-means. This is attributed to the large difference in the number of samples in each class in the dataset. The degradation of the proposed method however, is less severe compared to the baselines. The class `person` is not only most frequent bust also frequently co-occurs with the other classes in the dataset. This imbalance in the number of samples results in a large bias towards positive or negative constraints, resulting in relatively poor feature selection. This can be alleviated by balancing the number of positive and negative constraints.

Overall, the proposed feature selection method, using both group-LASSO and LASSO, results in a vocabulary pruning of about 75-80%, on average for two-class problems. Larger number of classes may retain larger fraction of key-points. Since the key-points are clustered using a larger number of images than those considered for each clustering task, one might observe that there are naturally irrelevant key points for each task. However, that is not the case. In almost all the clustering tasks, most of all the visual keywords are observed.

6. Summary and conclusions

An online algorithm is presented for pruning the vocabulary in image analysis tasks that use bag of words representation. Online algorithms are computationally efficient since they learn incrementally. Vocabulary pruning aids in representing images using smaller feature vectors, which naturally reduces the computation time required for sub-

Task	Proposed				Baseline	
#	Group LASSO		LASSO(L_1)		POLA(L_2)	
1	1148	509	1263	91	4929	187
2	986	449	1082	113	4982	30
3	1133	602	1204	310	4995	2
4	816	372	1170	82	4994	2
5	682	536	943	64	4834	473
6	1134	363	1156	124	4991	4
7	1283	537	1268	102	4996	2
8	1050	446	1213	124	4799	616
9	682	377	971	100	4943	163
10	1118	435	1092	45	4994	2
11	790	336	1025	92	4985	23
12	495	198	847	245	4921	215
13	999	377	1180	92	4978	34
14	729	391	940	55	4992	9
15	665	347	969	84	4982	37

Table 2. Mean and standard deviation of the number of visual words (from a total of 5,000) selected by the proposed LASSO and Group-LASSO method vs the L_2 DML algorithm. POLA is not a feature selection technique, and hence learns the weights for all the features. The batch mode forward search algorithm always selected 150 features, and hence is not reported in the table. The tasks are defined in Table 1

.

sequent clustering, classification or retrieval. The quality of pruned vocabulary is evaluated using a clustering task, and is comparable to that of batch learning algorithms. A controlled study was performed to evaluate the merits of the proposed algorithm. Although the proposed algorithm is evaluated on visual vocabulary pruning task, it is applicable to other feature selection tasks as well. Real world applications may derive the pairwise constraints automatically from auxiliary information (e.g. text in web pages), where one may also be able to exploit the degree of relation between images.

7. Acknowledgements

This research was partly supported by the ONR grant no. N000140710225 and the NSF grant no IIS-0643494.

References

[1] K. Barnard, P. Duygulu, and D. Forsyth. Clustering art. In *CVPR*, volume 2, 2001.

[2] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge Univ. Press, 2006.

[3] M. Dash and H. Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156, 1997.

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

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL VOC Challenge 2007 (VOC2007) Results.

[6] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *JMLR*, 3:1157–1182, 2003.

[7] P. Jain, B. Kulis, I. Dhillon, and K. Grauman. Online metric learning and fast similarity search. In *NIPS*, 2008.

[8] H. Liu and H. Motoda. *Computational methods of feature selection*. Chapman & Hall/CRC, 2008.

[9] D. Lowe. Object recognition from local scale-invariant features. In *ICCV*, volume 2, pages 1150–1157, 1999.

[10] P. K. Mallapragada, R. Jin, and A. K. Jain. Online feature selection using group-LASSO. Technical Report MSU-CSE-10-8, Michigan State University, 2010.

[11] L. Meier, S. van de Geer, and P. Buhlmann. The group lasso for logistic regression. *J.Royal Stat Soc. B*, 70(1):53, 2008.

[12] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

[13] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, volume 5, 2006.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, volume 3613, pages 1575–1589, 2007.

[15] S. Shalev-Shwartz, Y. Singer, and A. Ng. Online and batch learning of pseudo-metrics. In *ICML*, 2004.

[16] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, 2003.

[17] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soc. B*, pages 267–288, 1996.

[18] J. Yang, Y. Jiang, A. Hauptmann, and C. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *ACM Workshop on MIR*, page 206, 2007.

[19] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J.Royal Stat Soc. B*, 68(1):49, 2006.

[20] P. Zhao and B. Yu. Boosted lasso. *JMLR*, 2004.

[21] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani. 1-norm support vector machines. In *NIPS 16*, page 49, 2004.

[22] H. Zou and M. Yuan. The F-∞-norm support vector machine. *Statistica Sinica*, 2006.