# Automatic Image Annotation Using Group Sparsity

Shaoting Zhang[†], Junzhou Huang[†], Yuchi Huang[†], Yang Yu[†], Hongsheng Li[§], Dimitris N. Metaxas[†]
[†]Department of Computer Science, Rutgers University, Piscataway, NJ, 08854
[§]Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA, 18015
{shaoting,jzhuang,yuchuang,yyu,dnm}@cs.rutgers.edu, h.li@lehigh.edu

## Abstract

*Automatically assigning relevant text keywords to images is an important problem. Many algorithms have been proposed in the past decade and achieved good performance. Efforts have focused upon model representations of keywords, but properties of features have not been well investigated. In most cases, a group of features is preselected, yet important feature properties are not well used to select features. In this paper, we introduce a regularization based feature selection algorithm to leverage both the sparsity and clustering properties of features, and incorporate it into the image annotation task. A novel approach is also proposed to iteratively obtain similar and dissimilar pairs from both the keyword similarity and the relevance feedback. Thus keyword similarity is modeled in the annotation framework. Numerous experiments are designed to compare the performance between features, feature combinations and regularization based feature selection methods applied on the image annotation task, which gives insight into the properties of features in the image annotation task. The experimental results demonstrate that the group sparsity based method is more accurate and stable than others.*

## 1. Introduction

In the past decade, the number of images online and offline has increased dramatically. Many search engines retrieve relevant images by text-based searching without using any content information. Thus, assigning more relevant keywords is significant and can improve the image search quality. The purpose of image annotation is to automatically assign relevant text keywords to any given image, reflecting its content. It is a difficult task because of the lack of correspondence between keywords and image regions. Recent publications testify that it is an active subject of research [10, 20, 21, 12]. Since the amount of images with available annotations is increasing, many machine learning algorithms have been proposed to solve this problem by making use of ground truth [6]. Progress has been made by evaluations on standardized annotated data sets. In Section 1.1, we review the previous work of image annotation in detail.

The main shortcomings of existing work are twofold. First, features are often preselected, yet the properties of different features and feature combinations are not well investigated in the image annotation task. Second, these predefined features do not equally or positively contribute to the annotation performance. This problem can occur at both of the high and low levels. For instance, histograms of RGB and HSV color space are widely used in image recognition applications. At the high level, the whole histograms from RGB and HSV are correlated. At the low level, the bins in each histogram also have connections. Thus features may have sparse prior and can be pruned or assigned different weights. Furthermore, these features may have a group clustering trend, which means that the nonzero elements exist in the union of subspaces. However, previous works do not use these two properties together to select features to improve the annotation performance.

In this paper we present a group sparsity based method to solve the feature related problems in the image annotation application. Our method includes training and testing procedures. The training part selects low level features (e.g., bins in the feature histogram) using both sparsity and group clustering priors. These priors improve the model's robustness to noise. This method is a variation of regularization problems. It is inspired by the recently proposed group sparsity [33, 17] in the compressive sensing community. Testing part automatically annotates input images by transferring keywords from similar images. In our annotation framework, keyword similarities are also incorporated to find images with both similar keywords and similar visual contents. Numerous experiments are designed to compare the performance among different features and regularization methods applied on the image annotation task, which gives insight into the feature related properties in the image annotation task. From our experiments, the group sparsity based method shows improved accuracy and robustness compared to other regularization methods.

**Our contributions are:** 1) An effective framework is introduced to solve the image annotation problem: it includes a group sparsity based feature selection method and an image pair extraction method. Both of them have not been used in this large-scale problem. 2) We report extensive experiments, designed to give insight into the feature properties and provide in-depth comparisons between regularization methods applied on the annotation problem.

### 1.1. Previous Work on Image Annotation

Many approaches have been proposed to address the annotation task. Three main groups are identified [12]: generative models, discriminative models and nearest neighbor based methods.

**Generative models** can be further categorized as topic models and mixture models. Topic models annotate images as samples from a specific mixture of topics. Each topic is a distribution over image features and annotation words. Examples of topic models include latent Dirichlet allocation [2], probabilistic latent semantic analysis [25], hierarchical Dirichlet processes [31], and machine translation methods [7]. Mixture models define a joint distribution over image features and annotation keywords. Given a new image, these models compute the conditional probability over keywords given the visual features by normalizing the joint likelihood. A fixed number of mixture components over visual features per keyword can be used [4], or a mixture model can be defined by using the training images as components over visual features and keywords [8, 18, 19].

**Discriminative models** for keyword prediction have also been proposed [10, 14]. These methods learn a separate classifier for each keyword, and use them to predict whether the test image belongs to the class of images that are annotated with each particular keyword. However, both generative and discriminative models preselect features and do not analyze the differences within features. Feature selection is not a concern either.

**Nearest neighbor** based methods have become more attractive recently since the amount of training data is rapidly increasing, such as using label diffusion over a similarity graph of labeled and unlabeled images [21, 27], and learning discriminative models in neighborhoods of test images [34]. A nearest-neighbor keyword transfer mechanism was recently introduced [23]. In this method, image annotation is solved as a retrieval problem. Nearest neighbors are determined by the average of several distances (called Joint Equal Contribution, JEC) computed from different visual features. Keywords are then transferred from neighbors to the given image. Elementary color and texture features are tested and compared. Regularization based feature selection is also considered by using keyword similarity. Weights are computed in the "feature level", which means that all histograms within the same feature share one common weight. However, it does not give better results than the equally weighted approach, and sparsity is not produced. TagProp [12] is a new nearest neighbor type model that predicts keywords by taking a weighted combination of the keyword absence and presence among neighbors, showing state-of-the-art performance.

## 2. Feature Selection with Group Sparsity

In this section we introduce the feature selection method for the image annotation task by using group sparsity properties, along with the training and testing procedures.

**Notations:** Consider the data $(x_i, y_i)$ where $x_i = (x_{i1}, ..., x_{ip})$ and $y_i$ are the regressors and response for the $i$th observation. Denote $X$ as the $n \times p$ feature matrix, with row $j$ of $X$ being data $x_j$. Given all responses $Y = (y_1, ..., y_n) \in \mathbb{R}^n$ that are generated from a sparse linear combination of the features plus a stochastic noise vector $\epsilon \in \mathbb{R}^n$: $Y = Xw + \epsilon$, where $w$ is the weight vector and can select features. By far the most popular loss function to calculate $w$ in this regression problem is the least square estimate, which is also named as the minimizer of the residual sum of squared errors:

$$\hat{w} = \arg\min_{w \in \mathbb{R}^p} \|Xw - Y\|_2^2. \tag{1}$$

**Annotation:** In our framework of the image annotation, the regressor $y_i$ represents the similarity within the image pair, where $y_i = 1$ means the sample is positive and the images in $i$th pair are similar, while $y_i = -1$ means the sample is negative and the images are dissimilar. To incorporate the keyword information into the objective function, the definition of similarity should consider the similarity of keywords. In other words, positive samples share keywords, while negative samples do not. The quality of these pairs highly influence the overall performance. However, the ground truth of similarity is generally not available. It is obtained by an iterative method proposed in Section 3.2. Here we assume that it is available. $x_i$ is the vector of the difference between features of the $i$th sample (a pair of images). If $w_j = 0$, the $j$th element in the difference vector $x_i$ does not contribute, which means that the $j$th bin in the feature vectors can be pruned directly. Based on the pruned features and weights, the similarity between the testing image and all training images are obtained. The keywords of images with highest similarity scores are annotated to the test image. Note that a larger result means a higher similarity. Since keyword similarities are incorporated into this learning framework, theoretically it can perform better than JEC, which solely considers visual feature distance when searching the similar images.

**Sparsity prior:** The analytical solution of $w$ in (1) is represented as $(X^T X)^{-1} X^T Y$. However, the matrix $X^T X$ may be singular because of the correlation of variables or

**Training**

**Input:** Feature matrix of training images $F \in \mathbb{R}^{m \times p}$, where $m$ is the number of training images, and each row is a feature vector $f_i \in \mathbb{R}^p$; similar and dissimilar image pair list $L \in \mathbb{R}^{n \times 2}$; step size $t$.

Compute the matrix of feature differences $X \in \mathbb{R}^{n \times p}$ based on $F$ and $L$, where $n$ is the number of pairs.
Generate similarity vector $Y \in \mathbb{R}^n$ based on $L$.
Generate groups $G_1$ to $G_m$ according to feature groups.
Use projected-gradient algorithm [3, 28] (or other optimization methods described in [26, 1]) to minimize (3).
**Output:** weights $w \in \mathbb{R}^p$.

**Testing**

**Input:** Weight vector $w \in \mathbb{R}^p$; feature vector of a test image $t \in \mathbb{R}^p$; feature matrix of training images $F$; keyword ground truth of training images.

Prune $t$, $F$ and $w$ according to the nonzero elements in $w$. Thus the sizes of $t$, $F$ and $w$ have shrunk.
Generate a empty similarity vector $s \in \mathbb{R}^m$.
**Loop:** Iterate over the rows of $F$

- $s_i = (t - f_i) \cdot w$

Find the five largest values in $s$ as the most similar images in terms of keyword similarity.
Transfer keywords according to their local frequency.
**Output:** predicted keywords for the test image.

Table 1. The algorithms of training and testing procedures of our annotation framework.

the fact that $p$ can be larger than $n$, making the model unstable. Regularization approaches are widely used to alleviate this problem, which is written the following format:

$$\hat{w} = \underset{w \in \mathbb{R}^p}{\arg\min} \left[ \frac{1}{n} \|Xw - Y\|_2^2 + \lambda \|w\|_q \right], \qquad (2)$$

where $q = 2$ denotes $L^2$ regularization, and $q = 1$ represents $L^1$ regularization (LASSO [29]). $\lambda\|w\|_2$ can be interpreted as a prior that $w$ should not be too large. Although $L^2$ regularization increases the stability of the model, it does not induce parsimony since $L^2$ norm does not encourage sparsity. When irrelevant features are present in $X$, the weights produced under the sparsity prior can outperform those with $L^2$ constraints. In contrast, $\lambda\|w\|_1$ incorporates a sparsity prior. The solutions produced can be as sparse with $L^0$ regularization in some underdetermined systems. However, as shown in Section 4, using this prior alone can produce over-sparse solutions in our systems, which adversely affect the performance.

**Clustering prior:** In the features of the image annotation task, the clustering prior can also be utilized. Each feature such as RGB and HSV tends to be a group since the histograms of the same feature are not independent. We argue that by taking advantage of the feature group structure, the performance of feature selection models can be improved. Thus the problem is reformulated as:

(a) image (b) saliency (c) image (d) saliency

Figure 1. Automatically detecting salient regions without any prior. Because of the large scale region of the sky, (a) and (c) are not similar in terms of the RGB distance. The difference between (b) and (d) is much smaller.

$$\hat{w} = \underset{w \in \mathbb{R}^p}{\arg\min} \left[ \frac{1}{n} \|Xw - Y\|_2^2 + \lambda \sum_{j=1}^{m} \|w_{G_j}\|_2 \right], \qquad (3)$$

where features are partitioned into $m$ disjoint groups $G_1, G_2, ..., G_m$, and $w_{G_j}$ denotes the group of weights corresponding to the feature group $G_j$. In the image annotation task, the groups are naturally defined as different kinds of features such as RGB, Haar, etc. $\lambda \sum_{j=1}^{m} \|w_{G_j}\|_2$ is the combination of both $L^1$ and $L^2$ norms. $L^2$ norm is used for the weights inside of the same group, while $L^1$ norm is used to sum the results between groups. Using the group structure ensures more robust and accurate weights and still benefits from the sparsity. The algorithm framework is illustrated in Table 1. The weight $w$ is produced during the training stage by leveraging the group sparsity as regularization. In the testing stage, similarities between the test image and training images are calculated. Since the similarity is based on the keyword information, keywords from the most similar images are transferred to annotate the test image.

## 3. Features and Image Pairs

In this section, we introduce features used to evaluate our algorithm, and explain how to obtain similar and dissimilar pairs for when ground truth of similarity is not available.

### 3.1. Color and Texture Features

Color and texture are two important visual cues to represent the image content. To maximally use the color information extracted from the images and investigate their properties, we select features from many color spaces as color descriptors [30], including RGB, HSV, LAB, Opponent, *rg*histogram and transformed color distribution, and represent them as histograms. Texture is another well-studied component of image appearance. Gabor and Haar Wavelets [23] are selected for evaluation. Each image is filtered with Gabor wavelets at four scales and six orientations. The features of magnitude and phase angle at twenty four response images are concatenated into two feature vectors separately. The Haar Wavelet responses are generated in a similar way by convoluting an image with three

$2 \times 2$ edge filters. The information of the response magnitude is used as Haar feature. SIFT [22] and Histogram of Oriented Gradient (HOG) [5] are also tested as local descriptors. SIFT is extracted sparsely for Harris-Laplacian interest points. Images are then represented as a "bag-of-words" histogram. Extracting SIFT densely or incorporating color SIFT [30] can improve the performance, but these approaches generate much larger feature vectors, which increase the computation complexity of both training and testing procedures.

Instead of only using color features from the whole image [23], we also consider the saliency. As shown in Figure 1, (a) and (c) are not similar in terms of the $L^1$ distance of RGB histograms. The averaged distance is $0.11$ in all channels. Considering that many values in RGB histograms are flat zeros, $0.11$ is not small. The reason is that the colors of the large scale regions of the sky are quite different. By leveraging the features from saliency (Figure 1 (b, d)), the problem can be alleviated. The average distance between the salient regions is $0.06$ in this case. As we show later (Figure 3), color information from saliency is a good complement for the global information. The Spectral Residual Model (SRM) [15] is employed to automatically extract the saliency, which is independent of features, categories or other forms of prior knowledge of the object.

### 3.2. Obtaining Image Pairs

The features cannot be used for regularization algorithms directly since the objective function is a similarity function. For training purpose, similar and dissimilar image pairs are required, whose ground truth is not available. Since the goal is to assign relevant keywords, [23] proposed an approach to discover these pairs based on the keyword similarity. Any pair of images in the training set that share more than 4 keywords is considered as a positive training example, while a pair without any common keyword is a negative one. Intuitively, the distances within positive pairs are expected to be smaller than the ones within negative pairs, since images of similar pairs should be much closer with each other than dissimilar ones. However, the approach above induces noise (Figure 2 (a)) because there is no exact correlation between visual similarity and keyword overlap. Distributions of distances within positive and negative pairs are mixed (Figure 2 (b)). Using these image pairs, regularization methods underperform our task.

An iterative framework is proposed to obtain similar and dissimilar pairs with less noise. It is inspired by the relevance feedback algorithms and the expectation-maximization (EM) algorithm. The intuition is that we try to leverage both the distance and keyword similarity, so we can remove noise according to the feedback information (keyword similarity) from the training data, and iteratively improve the performance by finding better pairs and

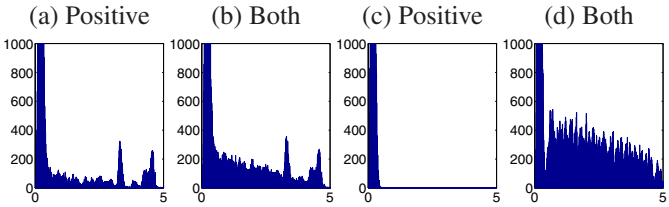


Figure 2. Histograms of distances within positive pairs (a, c), and within both positive pairs and negative pairs (b, d). (a, b) come from the method in [23]. (c, d) are from our method.

weights. Each iteration includes two steps.

**"E" step:** Positive and negative pair candidates are defined as the $k_1$ nearest and $k_2$ farthest neighbors of each image in the training data, measured by the weighted distance. The initial guesses of weights are equally weighted. The candidates are pruned according to the similarity of keywords. For example, the candidates with no common keyword are defined as negative samples, while candidates with more than $75\%$ common keywords are positive ones. In this case the keyword similarity in the candidates is the "feedback". In this application, $k_1$ is chosen four times larger than $k_2$. The reason is that farthest pairs in terms of visual distance usually do not share common keywords. Thus most negative candidates are preserved. However, nearest pairs may not share large set of common keywords. Therefore many positive candidates are pruned.

**"M" step:** These similar and dissimilar pairs selected in the "E" step are used as the image pairs for regularization methods. New weights are computed by minimizing (3), and can be used in "E" step for distance measurements.

Using these updated weights, this method can be iterated. In practice, the pairs from the first iteration are already much better than the previous technique solely relying on keywords. The histogram of distances within positive pairs is shown in Figure 2 (c). Noise is decreased and a small interval can be observed between positive and negative pairs (Figure 2 (d)). Using these new pairs, regularization methods perform much better in this learning problem.

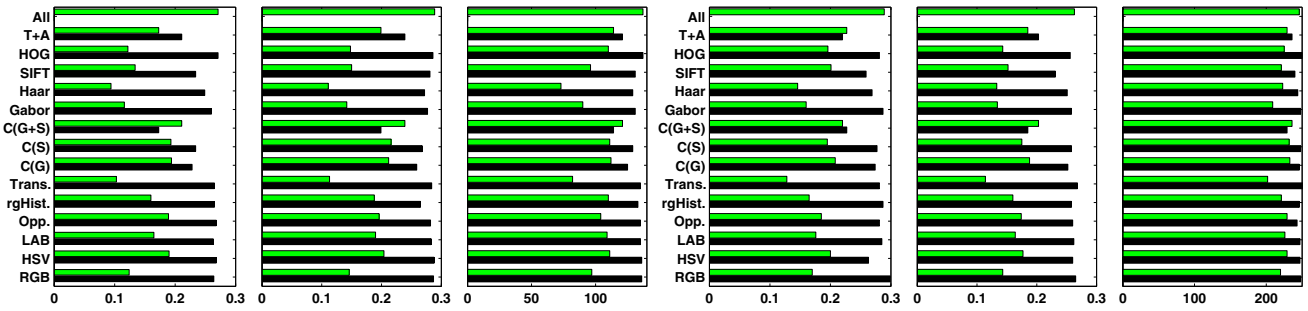
## 4. Experiments

In this section we first describe our experimental protocols. Then all features are evaluated using the image sets. After that, regularization based feature selection methods are evaluated and compared, which validates the benefits of sparsity and group clustering priors.

### 4.1. Experimental Settings

To evaluate the performance and compare with other state-of-the-art annotation algorithms, the proposed framework is experimentally evaluated using two benchmarks (Corel5K [7] and IAPR TC12 [11]).

**Corel5K**: It has become the benchmark for image annotation. As shown in Table 2, many state-of-the-art methods were tested on it. The set contains 5,000 images collected




(a) Corel5K, P (b) Corel5K, R (c) Corel5K, N+ (d) IAPR, P (e) IAPR, R (f) IAPR, N+

Figure 3. Annotation results of different features. The green bar shows the performance using the feature(s) in that row, while the black bar shows "leave-one-out" experiments when all features are used except the one(s) in that row. "C(G)" means that using all color information from the whole image (global), while "(S)" from the saliency. "T+A" includes texture and appearance features. $L^1$ is chosen for HOG, KL-divergence for LAB, and $\chi^2$ for the others. Equal weights are used to combine different features.

from the larger Corel CD set, split into 4,500 training and 500 test examples. The whole set consists of 50 groups. In each group there are 100 similar images, such as beach, aircraft and tiger. The set is annotated from a dictionary of 374 keywords, with each image having been annotated by an average of 3.5 keywords.

**IAPR TC12**: It is a collection of 20,000 images. Its size is suitable for testing the scalability of annotation algorithms. Since the annotations are free-flowing captions, preprocessing is necessary to extract nouns as keywords. We use the same annotation as in [23]. The size of the keyword dictionary is 291 and an average 4.7 for each image.

The annotation results of previously published algorithms are shown in Table 2. The average precision and recall rates as well as the number of total keywords recalled are reported for comparison. Our experiments are based on the same features and settings. Five keywords are annotated for each test image. With more keywords assigned, recall can be increased while precision is decreased. Annotating five keywords is a good compromise between these two measurements. The keyword transfer procedure is similar to [23] except that we do not consider the global frequency of keywords since it may not be available in real word applications.

## 4.2. Evaluation of Features

Annotation results of features are provided in Figure 3 (green bars). Although each feature performs differently on Corel5K and IAPR TC12, similarities can still be observed. Transformed color spaces such as HSV and LAB generally perform better than RGB. Interestingly, using all six color spaces together does not improve much upon the single feature since they are highly correlated. Features from saliency alone perform similar to the global information, but combining global and saliency colors makes an improvement. The reason is that the saliency extracting algorithm is not always accurate. It may fail for images with complex background. When the saliency is correct, these features are dominantly good in some cases (Figure 1), while global information is better when saliency map misinterprets the image. Thus saliency features can be a reasonable complement of global ones. Texture from saliency is not selected since it only performs slightly better than random guess (less than $3\%$ for precision and recall). It may be caused by the irregular shape of saliency. Texture information in IAPR TC12 achieves higher performance than the one in Corel5K, which shows that the importance of features vary in different image sets .

In the "leave-one-out" experiment (Figure 3, black bars), all features are used except for the one(s) in that row. In many cases, removing one set of features does not decrease performance. In fact, the results are even improved by pruning out RGB in IAPR TC12. Note that it does not show that these features are not important. It only validates our assumption of sparsity prior and redundancy of features. Since some features are highly correlated with each other, using all of them induces noise and decrease the performance. Besides, the processing time is also increased when using more features. Intuitively, the annotation performance can benefit from using this sparsity prior to prune features.

## 4.3. Evaluation of Regularization Methods

The first two rows of Figure 4 compare different regularization based feature selection methods applied on the Corel5K and IAPR TC12 respectively. The comparisons include equal weights, non-regularization, $L^2$, $L^1$ and group sparsity based regularization. They are based on the same group of features in Figure 3. The iterative framework is employed to get image pairs. Although regularization techniques are relatively robust to noise, the image pairs still influence the performance. Using pairs from our method performs better than using equal weights, while the similar number of pairs from the previous technique [23] only achieves about $10\%$ in terms of precision and recall in our experiments. In our experiment, the pairs from Corel5K contains less noise than the ones from IAPR TC12, because

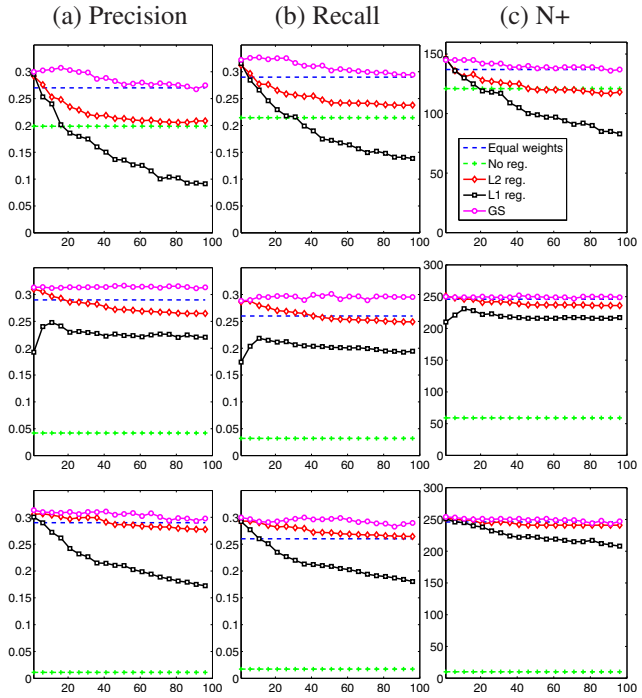



Figure 4. Annotation results of different regularization methods. 1st row: Corel5K; 2nd row: IAPR TC12; 3rd row: using weights calculated from Corel5K to test IAPR TC12. The legend is shown in the third figure of the first row. The horizontal axis represents the tuning parameter $\lambda$, from 1 to 100 with step size 5. The vertical axis represents the performance such as precision and recall.

| | Corel5K | | | IAPR TC12 | | |
|---|---|---|---|---|---|---|
| **Method** | **P** | **R** | **N+** | **P** | **R** | **N+** |
| **CRM**[19] | 0.16 | 0.19 | 107 | – | – | – |
| **InfNet**[24] | 0.17 | 0.24 | 112 | – | – | – |
| **NPDE**[32] | 0.18 | 0.21 | 114 | – | – | – |
| **SML**[4] | 0.23 | 0.29 | 137 | – | – | – |
| **TGLM**[21] | 0.25 | 0.29 | 131 | – | – | – |
| **MBRM**[8] | 0.24 | 0.25 | 122 | 0.24 | 0.23 | 223 |
| **JEC**[23] | 0.27 | 0.32 | 139 | 0.28 | 0.29 | 250 |
| **LASSO**[23] | 0.24 | 0.29 | 127 | 0.28 | 0.29 | 246 |
| **TagProp**[12] | 0.33 | 0.42 | 160 | 0.46 | 0.35 | 266 |
| **GS** | 0.30 | 0.33 | 146 | 0.32 | 0.29 | 252 |

Table 2. Annotation results of previous published algorithms on Corel5K and IAPR TC12. N+ denotes the number of recalled keywords. GS is the proposed method and stands for group sparsity.

Corel5K also has category information, which can be used with the keyword information together as feedback. IAPR TC12 only includes keyword ground truth. Thus weights calculated from Corel5K generally perform better than the ones from IAPR TC12.

The models' sensitivities of the tuning parameter $\lambda$ are investigated by observing a large range of $\lambda$ ($[1, 100]$). Note that equal weights and non-regularization methods do not depend on the tuning parameter. The performance of equal weights with our features are available in Table 2 and Figure 4. Generally it is robust and performs well since sophisticated features are employed. However, it does not use any keyword similarity information to find nearest neighbors. The assumption that images with less feature distance share more common keywords is not universally true. Thus the performance can be improved by using regularization methods with image pairs selected from keyword similarity, and our results validate this point.

Using least square without regularization underperforms the task since the singular system induces numerical errors even using pseudo inverse. It performs worse in IAPR TC12 because IAPR TC12 includes much more testing images and the image pairs contains more noise. $L^2$ regularization highly increases the robustness and performance of the solution. However, neither sparsity nor clustering prior are leveraged. The results are generally not sparse (see Section 4.4). Without the clustering prior, $L^1$ performs even worse than $L^2$ regularization, although sparsity is induced. The reason is that $L^1$ regularization can arbitrarily prune the features without keeping the structure. Thus important features may not be preserved, and the solution is less stable. Generally the starting points of regularization methods are close, since they tend to be the non-regularization problem when $\lambda$ is close to zero.

Our method leverages both sparsity and clustering priors, and it performs better than previous methods. The performance does not decrease much when increasing the tuning parameters since the structure prior contributes to the stability of the model. In fact, within a large scale of $\lambda$ the model achieves similar performance. There are small fluctuations in Corel5K when the tuning parameter is not very large. It may be caused by the discreteness of the keywords transfer mechanism. Since IAPR TC12 includes 2,000 test images, the discreteness problem can be alleviated and the small fluctuations are reduced. When the tuning parameter is larger than a certain level the overfitting problem still exists in Corel5K. The quantitative results of our method are available in Table 2. It outperforms other regularization based methods and all annotation algorithms except recently proposed TagProp [12]. It is reasonable since we focus on the feature selection and do not create models for single keywords.

To demonstrate that the models do not rely on the specific image set and can be used in practical applications, weights calculated from Corel5K are used on the IAPR TC12 directly. The results are shown in the third row of Figure 4. Interestingly, the weights from Corel5K work well and sometimes even better than ones from IAPR TC12 since the image pairs from Corel5K contains less noise.

Table 3 compares predicted keywords and human annotations of images from Corel5K and IAPR TC12. In all of our experiments, five keywords are transferred to each image, unless the keywords from similar images are less than

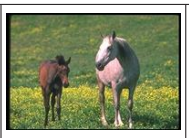
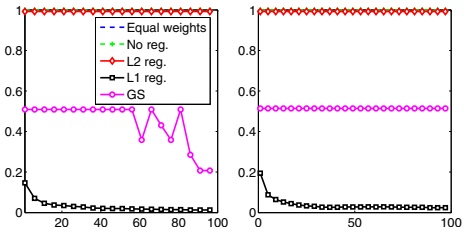
| |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Predicted keywords | flower front group *bike forest* | bed bedside lamp *blanket pillow* | girl hair wall portrait *boy* | ship water sunset *cloud view* | front girl hat wall *bloom* | landscape mountain snow *cloud sky* |
| Human annotation | flower front group | bed bedside lamp | girl hair wall portrait | ship water sunset | front girl hat wall | landscape mountain snow |
| |  |  |  |  |  |  |
| Predicted keywords | sky jet plane smoke *birds* | bridge flowers *steel sky arch* | water pool *sky* swimmers people | wall cars tracks formula *turn* | field horses mare foals *grass* | sky tree garden cottage *roofs* |
| Human annotation | sky jet plane smoke | bridge flowers | water pool swimmers people | wall cars tracks formula | field horses mare foals | sky tree garden cottage |

Table 3. Predicted keywords versus human annotations for images from IAPR TC12 (first row) and Corel5K (second row). The keywords are predicted using our proposed algorithm. The differences in predicted keywords are marked in italic font. Although redundancies may misinterpret the images, some keywords explain them well.




(a) Corel5K (b) IAPR TC12

Figure 5. Sparsity of weights generated from different regularization methods. The legend is shown in the first figure. The horizontal axis represents the tuning parameter $\lambda$, and the vertical axis represents the percentage of nonzero elements in the weights. Note that curves of equal weights, "no reg." and "$L^2$ reg." are overlapped at the top since no sparsity is induced.

five. Thus the average number of predicted keywords is more than that of human annotations. In these cases, some locally frequent keywords in the most similar images can appear in the prediction as redundancies, which may misinterpret images. Interestingly, in many cases these additional keywords also explain the images well, such as "bike" in the first image of Table 3.

### 4.4. Sparsity

Sparsity is preferred since it can alleviate the overfitting problem and increase the robustness of the model. Furthermore, when computing pairwise distances between a test image and all training images, the computation complexity is $O(mn)$, where $n$ is the number of training images and $m$ is the number of features. Thus reducing the number of features decreases the computation complexity. Figure 5 shows the percentage of nonzero elements of weights after regularization with different tuning parameters. From the results, equal weights method puts no constraint on the number of features. Thus all weights are nonzero. When many features are used, the performance may be highly decreased. The result of the least square solution without regularization is similar to that of equal weights method since no parsimony constraint is induced. $L^2$ regularization tends to have relatively fewer large weights with many small ones. Sparsity is not produced. $L^1$ regularization effectively decreases the number of nonzero elements. However, the annotation performance is also reduced (see Section 4.3). Our group sparsity based method not only produces relative sparse solutions, but also does not sacrifice the annotation performance. Its sparsity is generally stable because of the benefit of structure priors.

### 4.5. Summary

The experimental results show the following facts. 1) Performances of many features and feature combinations in the annotation task are reported. The "leave-one-out" test shows that removing some features usually does not decrease the annotation performance much. Since these features may highly correlate with each other, sparsity prior can contribute to the pruning procedure in a certain level. 2) Regularization based feature selection methods are employed to incorporate keyword information. Group clustering prior benefits the performance and stability of the model in the annotation task. It also produces relatively sparse solutions. The weights calculated from a set of images also work well for the other set of images. In all experiments the method using group sparsity priors shows improved performance and stability.

## 5. Conclusions

In this paper, we proposed a group sparsity based feature selection method to solve the annotation task. This algorithm leverages both sparsity and clustering priors to prune the features. Keyword information is also incorporated into the framework by searching similar and dissimilar image pairs from both keyword similarities and relevance feedback. Compared to other regularization methods, it shows higher performance in the image annotation task. In the future we will investigate the importance of different settings in our algorithm, such as grouping methods and feature combinations, since these settings may influence the regularization performance of group sparsity. We also plan to combine our method with the distance learning approaches [13, 9] by incorporating this group sparsity term into their loss functions. Dynamic Group Sparsity (DGS) [16] will also be tested in this application.

## References


[1] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, 2006.

[2] K. Barnard, P. Duygulu, D. Forsyth, N. D. Freitas, D. M. Blei, J. K, T. Hofmann, T. Poggio, and J. Shawe-taylor. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[3] E. Berg, M. Schmidt, M. Friedlander, and K. Murphy. Group sparsity via linear-time projection. In *Technical report, TR-2008-09*, 2008.

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

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[6] R. Datta, D. Joshi, J. Li, James, and Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 39:2007, 2006.

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

[8] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *CVPR*, 2004.

[9] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, pages 513–520, 2004.

[10] D. Grangier and S. Bengio. A discriminative kernel-based approach to rank images from text queries. *PAMI*, 30(8):1371–1384, 2008.

[11] M. Grubinger, P. D. Clough, H. Muller, and T. Deselaers. The iapr tc-12 benchmark - a new evaluation resource for visual information systems. 2006.

[12] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, 2009.

[13] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742, 2006.

[14] T. Hertz, A. Bar-hillel, and D. Weinshall. Learning distance functions for image retrieval. In *CVPR*, pages 570–577, 2004.

[15] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, 2007.

[16] J. Huang, X. Huang, and D. Metaxas. Learning with dynamic group sparsity. In *ICCV*, 2009.

[17] J. Huang and T. Zhang. The benefit of group sparsity. In *Technical Report arXiv:0901.2962, Rutgers University*, 2009.

[18] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, 2003.

[19] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *NIPS*, 2003.

[20] J. Li and J. Wang. Real-time computerized annotation of pictures. *PAMI*, 30(6):985–002, 2008.

[21] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recogn.*, 42(2):218–228, 2009.

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[23] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *ECCV*, pages 316–329, 2008.

[24] D. Metzler and R. Manmatha. An inference network approach to image. In *CIVR*, pages 42–50, 2004.

[25] F. Monay and D. Gatica-Perez. Plsa-based image auto-annotation: Constraining the latent space. In *ACM Multimedia*, pages 348–351, 2004.

[26] G. Obozinski, B.Taskar, and M. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 2009.

[27] J. Pan, H. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery, 2004.

[28] M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. In *CVPR*, 2008.

[29] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society*, 58:267–288, 1994.

[30] K. van de Sande, T. Gevers, and C. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, 99(1).

[31] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical dirichlet process model. In *MDM*, pages 1–7, 2008.

[32] A. Yavlinsky, E. Schofield, and S. Ruger. Automated image annotation using global features and robust nonparametric density estimation. In *CIVR*, pages 507–517, 2005.

[33] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.

[34] H. Zhang, A. C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, pages 2126–2136, 2006.