Weighted Substructure Mining for Image Analysis

Sebastian Nowozin, Koji Tsuda
Max Planck Institute for Biological Cybernetics
Spemannstrasse 38, 72076 Tübingen, Germany
{sebastian.nowozin,koji.tsuda}@tuebingen.mpg.de

Takeaki Uno
National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430, Japan
uno@nii.jp

Taku Kudo
Google Japan Inc., Cerulean Tower 6F, 26-1 Sakuragaoka-cho, Shibuya-ku, Tokyo, 150-8512, Japan
taku@google.com

Gökhan Bakır
Google GmbH, Freigutstrasse 12, 8002 Zurich, Switzerland
ghb@google.com

Abstract

In web-related applications of image categorization, it is desirable to derive an interpretable classification rule with high accuracy. Using the bag-of-words representation and the linear support vector machine, one can partly fulfill the goal, but the accuracy of linear classifiers is not high and the obtained features are not informative for users. We propose to combine item set mining and large margin classifiers to select features from the power set of all visual words. Our resulting classification rule is easier to browse and simpler to understand, because each feature has richer information.

As a next step, each image is represented as a graph where nodes correspond to local image features and edges encode geometric relations between features. Combining graph mining and boosting, we can obtain a classification rule based on subgraph features that contain more information than the set features. We evaluate our algorithm in a web-retrieval ranking task where the goal is to reject outliers from a set of images returned for a keyword query. Furthermore, it is evaluated on the supervised classification tasks with the challenging VOC2005 data set. Our approach yields excellent accuracy in the unsupervised ranking task compared to a recently proposed probabilistic model and competitive results in the supervised classification task.

1. Introduction

In the last decade the development of inexpensive digital cameras and the growth of the Internet produced a wealth of easily accessible digital image content. Despite some research progress, the ability of programs to automatically interpret the content of images, or even just assisting a human user to sort or group images is still rather limited, both in terms of accuracy and scalability. For example, thousands of images are easily retrieved using keyword searching by, *e.g.*, Google Image. Typically they contain *outlier images* that do not semantically match what the user really wanted. Therefore it would be helpful if a machine learning algorithm can identify those outliers automatically [6]. Another attractive task would be to classify retrieved images automatically into folders. This is basically a supervised multiclass image classification problem.

Images can be modeled as collections of parts, as is done *e.g.* in the popular bag-of-words model. Given such representation we can try to identify *patterns* that frequently appear among the images. Cheng et al. [1] recently showed under moderate assumptions that frequent patterns are more likely to be discriminative for telling apart samples of different classes. Motivated by this result, we propose to use *frequent item set mining* on the bag-of-words representation in order to identify likely discriminative patterns in a collection of images. By combining this mining step with a 1-class linear ν-SVM we obtain a function to rank each image by the similarity to the overall collection of images. We

apply the algorithm in a purely unsupervised setting to detect outlier images in a web-retrieval task where images are the result to a keyword query.

In supervised classification where we have known training class labels we can directly search for the most discriminative patterns instead of assuming frequent patterns to be the most discriminative ones. By using *weighted substructure mining* algorithms the patterns maximizing a score function can be sought efficiently. We obtain a classification function using these patterns by combining the weighted mining with linear programming boosting [2] to a new classifier termed *item set boosting*. This new classifier allows us to iteratively build an optimal classification rule as a linear combination of simple and *interpretable* hypothesis functions, where each hypothesis function checks for the presence of a combination of visual codewords.

The proposed approach is flexible and works for other image representations as well. In this paper, we demonstrate this by replacing the bag-of-words representation with one based on labeled connected graphs. In the graph each vertex represents a local image feature and the geometric relationships between the features are encoded as edge attributes. Instead of item sets, the discriminative patterns are subgraphs, which capture co-occurrence of multiple features as well as their geometric relationships.

An advantage of using efficient weighted substructure mining algorithms is that we can exhaustively search very large pattern spaces. For the bag-of-words representation with n codewords we effectively search a pattern space containing potentially $O(2^n)$ patterns. For subgraphs the pattern space is an order of magnitude larger, yet efficient graph mining techniques still allow us to search the space exhaustively.

We start with a short overview of recent approaches to object classification in the presence of clutter. In Section 2 we discuss our approach to unsupervised image ranking and evaluate it experimentally. In Section 3 we describe our approach to supervised object classification. The main components, generalized weighted substructure mining and the LPBoost algorithm, are described in detail and are experimentally validated. Finally, Section 4 contains a summary of the results.

1.1. Literature Review

Some researchers have concentrated on evaluating and improving the features used for object classification [22, 15, 8]. If the features are expressive enough, simple classifiers such as Support Vector Machines can be used successfully for learning. Adapted versions of SVMs and Boosting have also been used [21, 17].

Quack *et al*. [11] use frequent item set mining on interest points extracted for each video frame in a video sequence. Each frame becomes a set of interest points and frequent spatial configurations corresponding to individual objects are found from these sets. An earlier similar work is Sivic and Zisserman [16].

Russell *et al*. [13] discover object classes in an unsupervised setting. Given many images, for each image a set of feature points are extracted. Additionally each image is segmented into regions multiple times, varying the segmentation parameters. Each segment covers a set of interest points and the problem of identifying object classes is posed as the problem of finding segments that consistently share similar interest points in one region. They demonstrate that natural object classes are recovered.

2. Unsupervised image ranking

Consider the task of keyword based image retrieval, where for a given keyword a set of images $x_1, x_2, \ldots, x_N$ is retrieved. Most of the images will be related to the keyword, but there will be a small fraction of images that do not relate to the keyword. Patterns which consistently appear in the samples are likely to be discriminative for sorting out the outliers.[1]

For this ranking task we use the *bag-of-words* representation commonly used in computer vision based on local image features. Local image features are a sparse representation for natural images. Modern local feature extraction methods work in two steps. First, an *interest point operator* defined on the image domain extracts a set of *interest points* [9]. Second, the image information in the neighborhood around each interest point is used to build a fixed-length descriptor such that invariance and robustness against common image transformations is obtained [10]. We assume that each extracted interest point p_i has image coordinates $p_i.coords \in \mathbb{R}^2$ and relative scale information $p_i.scale \in \mathbb{R}^+$. For each image, a set of local image features is extracted and the feature descriptors are projected onto a set of discrete "visual words" from a codebook. The codebook is created a-priori by k-means clustering. The discretization is carried out using nearest-neighbor matching to the codebook vectors.

2.1. Method

Denote by d the number of visual words in the codebook. An image is represented as a set of visual words, equivalently d-dimensional binary vector $\boldsymbol{x} \in \{0,1\}^d$. In item set mining, a pattern $\boldsymbol{t} \in \{0,1\}^d$ is also defined as a set of visual words. A pattern $\boldsymbol{t}$ is included in $\boldsymbol{x}$, i.e., $\boldsymbol{t} \subseteq \boldsymbol{x}$, if for all $\boldsymbol{t}_i = 1$ we also have $\boldsymbol{x}_i = 1$. Denote by $\mathcal{T}$ the set of all possible patterns, whose number of non-zero elements is between t_{min} and t_{max}. Denote by $I(\cdot)$ the indicator function that is 1 if the condition inside is satisfied and 0

[1] The relationship between frequency of a pattern and its discriminative information has recently been examined by Cheng et al. [1].

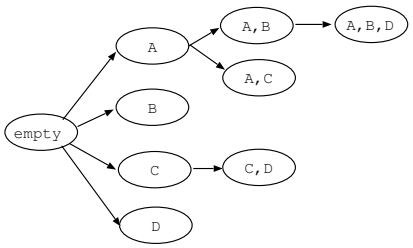

Figure 1. Search tree for frequent item set mining. Not all combinations are visited due to pruning.

otherwise. Given a set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, the *support* of a pattern $\boldsymbol{t}$ in a set X is the number of times it appears in the elements of X, such that $\text{support}(\boldsymbol{t}; X) = \sum_n I(\boldsymbol{t} \subseteq \boldsymbol{x}_n)$. Frequent substructure mining is defined as follows

Problem 1 (Frequent substructure mining) *Given a minimum support threshold τ, find the complete set of patterns $T = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{|T|}\}$, $\boldsymbol{t}_i \in \mathcal{T}$, such that support$(\boldsymbol{t}; X) \geq \tau$.*

We used the *Linear time Closed item set Miner* (LCM) algorithm of Uno et al. [18]. There are two types of item set mining algorithms, *apriori* and *backtracking*. LCM is an instance of the latter and frequent item sets are found in a search tree as in Figure 1. The key idea of efficient mining is to exploit the *anti-monotonicity*, namely the frequency of a pattern is always smaller than or equal to that of its subpattern,

$$\boldsymbol{t} \subseteq \boldsymbol{t}' \;\Rightarrow\; \text{support}(\boldsymbol{t}; X) \geq \text{support}(\boldsymbol{t}', X).$$

The tree is generated from the root with an empty pattern, and the pattern of a child node is made by adding one item. As the pattern gets larger, the frequency decreases monotonically. If the frequency of the generated pattern $\boldsymbol{t}$ is less than m, it is guaranteed that the frequency of any superpattern of $\boldsymbol{t}' \supseteq \boldsymbol{t}$ is also less than m. Therefore, the exploration is stopped there (i.e., *tree pruning*). By repeating node generation until all possibilities are checked, all frequent patterns are enumerated.

The runtime complexity of LCM is *output linear*, namely the time is linear to the number of visited nodes in the search space. So if the minimum support threshold is large and the search space is limited well, LCM finishes in a short time. It is hard to theoretically relate the complexity to the number of examples ℓ, but practically LCM shows linear time growth (see Figure 2). In our experience, item set boosting was much faster than the support vector machine taking $O(\ell^2)$ time with a nonlinear kernel. We set the threshold τ such that the k most frequent patterns $T = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_k\}$ are extracted. For each image $\boldsymbol{x}_i$, a binary vector is built as $V_i = [I(\boldsymbol{t}_1 \subseteq \boldsymbol{x}_i), \ldots, I(\boldsymbol{t}_k \subseteq \boldsymbol{x}_i)]$, $V_i \in \{0,1\}^k$, which is the new representation for each image. On this new representation we train a 1-class ν-SVM classifier with a linear kernel [14]. The ν-SVM has a high response for "prototypical" samples. Thus, we establish a ranking by ordering the samples according to the output of the trained classifier. Samples on top of the list are more likely to contain what the user is looking for.

2.2. Experiments and Results

To evaluate the performance of our ranking approach we use the dataset of 5245 images from Fritz and Schiele [6]. The images were obtained by keyword searches for "motorbikes" and vary widely in shape, orientation, type and background scene. All images were assigned labels based on whether they actually show a motorbike or not; the labels are only used to measure the performance of the method and are never used in the training phase. Of all downloaded images, 194 images do not contain motorbikes and our task is to identify these outliers.

While model selection in unsupervised settings can be difficult, our algorithm has only two free parameters, the number of patterns k to mine and the SVM ν-parameter. The SVM ν parameter is an upper bound on the fraction of outliers the final classifier will produce and thus can be chosen to be roughly equal to the expected number of outliers. As we only consider the relative ranking of the images, the choice of ν turned out to be not critical and the results are consistent over a large range of values. The number of patterns k is fixed to $k = 128$ across all runs because initial experiments have shown this value is a good tradeoff between the extreme cases of having too few patterns – some samples contain no selected pattern at all – and too many patterns, most of which are not very discriminative.

The results for the unsupervised ranking are shown in Table 1.[2] The ROC curve produced by our approach is shown in Figure 3, the top 100 and bottom 100 images are shown in Figure 6. The interpretability of the most influential patterns is shown in Figure 5, the distribution of weights for the individual frequent patterns is given by Figure 4. The "baseline histogram" results are obtained by normalizing the bag-of-words vector and applying a linear 1-class ν-SVM. For the 80 Harris-Laplace and the dense 300 Hessian-Laplace bags we used a codebook size of 64 and 192 words, respectively. This choice is rather arbitrary, but the performance is comparable over large variations of the codebook sizes.

The runtime of the item set ranking is dominated by the feature extraction and training of the 1-class classifier. The mining step is very fast, as shown on Figure 2.[3]

[2] The ROC Equal Error Rate (EER) score shown for the results from Fritz and Schiele [6] is measured from their ROC curve, as the numerical value is not explicitly given.

[3] The measurements were taken on a P4 2.4 GHz system.

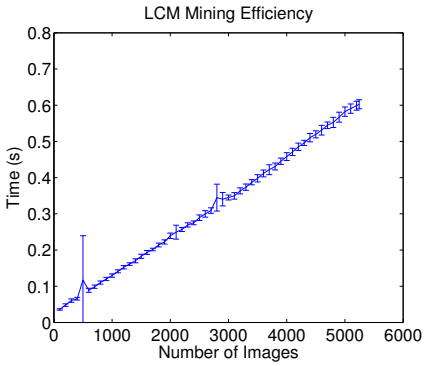

Figure 2. Runtimes of the LCM item set mining, averaged over 10 runs. The top 20 frequent item sets are returned for different number of input images, each with 300 discrete elements.

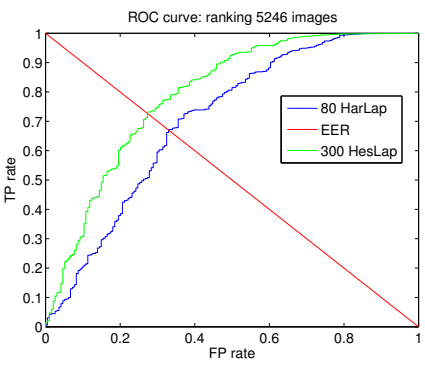

Figure 3. ROC curve of the unsupervised ranking approach on the Fritz and Schiele [6] data set.

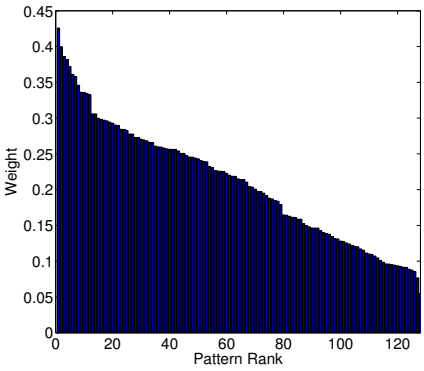

Figure 4. Distribution of weights assigned by the 1-class ν-SVM to the 128 frequent item set patterns (line 4 in table 1).

Figure 5. Top five most influential patterns. Each row shows the response of the global top five and global bottom five samples for one item set pattern, where each pattern carries a positive weight. In each image the features responding to the pattern are drawn in blue. As in the experiment $k = 128$ has been used and the top five images all have the same total sum as they all contain *all* frequent patterns. The obvious mistake in image 5243 is due to particularly low resolution of the image (77x54 pixels). The patterns are interpretable, for example pattern #19 (second row) captures a wheel and handlebar in the shown images; pattern #100 (third row) captures features on the wheels.

Method	EER
Fritz and Schiele [6], appearance	0.66
Fritz and Schiele [6], appearance+struct.	0.71
Item Set based, appearance, 80 HarLap	0.6701
Item Set based, appearance, 300 HesLap	0.7268
baseline histogram, 80 HarLap	0.5514
baseline histogram, 300 HesLap	0.5914

Table 1. Results for the unsupervised image ranking experiment. The scores are ROC equal error rates (EER).

2.3. Discussion

The baseline results clearly demonstrate that a combination of individual features alone is not informative enough for ranking, whereas the item set representation and its use of *combinations* of features is achieving state-of-the-art results. The interpretability of the results is difficult to evaluate quantitatively but clearly responses on background clutter and proper hits on object features and their quantitative influence can be identified quickly as is illustrated in Figure 5.

3. Supervised object classification

For supervised two-class object classification we are given a set of images with known class labels. For the training phase we assume each image is member of only one class.

Figure 6. Top 100 vs. bottom 100 ranked samples. The top 100 samples relevant for most web retrieval tasks do not contain any outlier images. The bottom 100 also contain a considerable number of motorcycles, but this is often due to small resolution images.

3.1. Classification rule

Our classification function is a linear combination of simple classification stumps $h(\boldsymbol{x}; \boldsymbol{t}, \omega)$ and has the form

$$f(\boldsymbol{x}) = \sum_{(\boldsymbol{t},\omega)\in\mathcal{T}\times\Omega} \alpha_{\boldsymbol{t},\omega} h(\boldsymbol{x}; \boldsymbol{t}, \omega). \tag{1}$$

The individual stumps $h(\cdot; \boldsymbol{t}, \omega)$ are parametrized by the pattern $\boldsymbol{t}$ and additional parameters $\omega \in \Omega$. We will use $\Omega = \{-1, 1\}$ and $h(\boldsymbol{x}; \boldsymbol{t}, \omega) = \omega(2I(\boldsymbol{t} \subseteq \boldsymbol{x}) - 1)$. Also $\alpha_{\boldsymbol{t},\omega}$ is a weight for pattern $\boldsymbol{t}$ and parameters ω such that $\sum_{(\boldsymbol{t},\omega)\in\mathcal{T}\times\Omega} \alpha_{\boldsymbol{t},\omega} = 1$ and $\alpha_{\boldsymbol{t},\omega} \geq 0$. This is a linear discriminant function in an intractably large dimensional space. To obtain an interpretable rule, we need to obtain a *sparse* weight vector $\boldsymbol{\alpha}$, where only a few weights are nonzero. In the following, we will present a linear programming approach for efficiently capturing patterns with non-zero weights.

3.2. LPBoost 2-class formulation.

To obtain a sparse weight vector, we use the formulation of LPBoost [2]. Given the training images $\{\boldsymbol{x}_n, y_n\}_{n=1}^{\ell}, y_n \in \{-1, 1\}$, the training problem is formulated as

$$\min_{\substack{\boldsymbol{\alpha},\\ \boldsymbol{\xi}\in\mathbb{R}_+^\ell,\\ \rho\in\mathbb{R}}} \quad -\rho + D\sum_{n=1}^{\ell} \xi_n \tag{2}$$

$$\text{sb.t.} \quad \sum_{(\boldsymbol{t},\omega)\in\mathcal{T}\times\Omega} y_n \alpha_{\boldsymbol{t},\omega} h(\boldsymbol{x}_n; \boldsymbol{t}, \omega) + \xi_n \geq \rho, \quad n = 1, \ldots, \ell \tag{3}$$

$$\sum_{(\boldsymbol{t},\omega)\in\mathcal{T}\times\Omega} \alpha_{\boldsymbol{t},\omega} = 1,$$

where ρ is the soft-margin, separating negative from positive examples, $D = \frac{1}{\nu\ell}$, $\nu \in (0, 1)$ is a parameter controlling the cost of misclassification which has to be found using model selection techniques, such as cross-validation. Solving the optimization problem (2) is very hard, due to the large number of variables in $\boldsymbol{\alpha}$. So we solve the following *equivalent* dual problem instead.

$$\min_{\substack{\boldsymbol{\lambda}\in\mathbb{R}_+^\ell,\\ \gamma\in\mathbb{R}}} \quad \gamma \tag{4}$$

$$\text{sb.t.} \quad \sum_{n=1}^{\ell} \lambda_n y_n h(\boldsymbol{x}_n; \boldsymbol{t}, \omega) \leq \gamma, \ (\boldsymbol{t}, \omega) \in \mathcal{T} \times \Omega$$

$$\sum_{n=1}^{\ell} \lambda_n = 1$$

$$0 \leq \lambda_n \leq D, \ n = 1, \ldots, \ell$$

After solving the dual problem, the primal solution $\boldsymbol{\alpha}$ is obtained from the Lagrange multipliers [2]. The dual problem has a limited number of variables, but a huge number of constraints. Such a linear program can be efficiently solved by *column generation* techniques: Starting with an empty pattern set, the pattern whose corresponding constraint is violated the most is identified and added iteratively. Each time a pattern is added, the optimal solution is updated by solving the reduced dual problem.

After one iteration has been completed, the objective function value γ at the solution of the dual is used to check for convergence in the next iteration. The next iteration's optimal $\hat{h}$ has to satisfy $\sum_{n=1}^{\ell} y_n \lambda_n \hat{h}(\boldsymbol{x}_n) > \gamma + \theta$, where θ is a convergence threshold. For $\theta = 0$, if $\hat{h}$ does not satisfy the inequality we converged to the globally optimal solution and stop iterating.

3.3. Weighted substructure mining

In each iteration of LPBoost we add one constraint to the linear program. Selecting the constraint to add is the computationally most expensive step. Therefore the efficiency of LPBoost depends on how efficiently the pattern corresponding to the *most violated constraint* of the

dual (4) can be found. The search problem is formulated as $(\hat{\boldsymbol{t}}, \hat{\omega}) = \underset{(\boldsymbol{t},\omega)\in\mathcal{T}\times\Omega}{\operatorname{argmax}}\ gain(\boldsymbol{t}, \omega)$, where

$$\text{gain}(\boldsymbol{t}, \omega) = \sum_{n=1}^{\ell} \lambda_n y_n h(\boldsymbol{x}_n; \boldsymbol{t}, \omega). \tag{5}$$

Problem (5) can be solved using the following generalization of frequent substructure mining in which each sample $\boldsymbol{x}_n$ is assigned a weight $\lambda_n \in \mathbb{R}$.

Problem 2 (Weighted substructure mining) *Given a set $X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $\boldsymbol{x}_i \in \mathcal{T}$, associated weights $L = \{\lambda_1, \ldots, \lambda_N\}$, $\lambda_i \in \mathbb{R}$, and a threshold τ, find the complete set of patterns and parameters $T_w = \{(\boldsymbol{t}_1, \omega_1), \ldots, (\boldsymbol{t}_{|T|}, \omega_{|T|})\}$, $\boldsymbol{t}_i \in \mathcal{T}$, $\omega_i \in \Omega$ such that for all $\boldsymbol{t}_i$ holds: $\sum_n \lambda_n h(\boldsymbol{x}_n; \boldsymbol{t}_i, \omega_i) \geq \tau$.*

This problem is more difficult than the frequent substructure mining, because we need to search with respect to a non-monotonic score function (5). So we use the following monotonic bound, $\text{gain}(\boldsymbol{t}, \omega) \leq \mu(\boldsymbol{t})$,

$$\mu(\boldsymbol{t}) = \max\{ \begin{array}{l} 2\sum_{\{n|y_n=+1, \boldsymbol{t}\subseteq\boldsymbol{x}_n\}} \lambda_n - \sum_{n=1}^{\ell} y_n\lambda_n, \\ 2\sum_{\{n|y_n=-1, \boldsymbol{t}\subseteq\boldsymbol{x}_n\}} \lambda_n + \sum_{n=1}^{\ell} y_n\lambda_n \ \} . \end{array}$$

The pattern set T_w is enumerated by branch-and-bound procedure based on this bound. In the enumeration problem (Problem 2), the threshold τ is fixed in the whole search process. To obtain the best pattern, the algorithm should be slightly modified such that τ is always updated to the current best value [7].

Like linear SVM, LPBoost is a very efficient method. The main computation is to solve a linear program where the number of variables is equal to the number of images and the number of constraints is kept small by the column generation technique. Experimentally, Derimiz et al. [2] observed a linear time growth to the number of examples in many benchmark datasets.

3.4. Graph Boosting

The bag-of-words model used in item set boosting discards all information about geometric relationships between the individual local features. Using graphs of local image features we can additionally model the geometry. For graphs the weighted substructure mining problem is tractable using graph mining algorithms; therefore our framework stays the same and we only replace item sets with graphs and weighted item set mining with weighted graph mining. We now define the graphs we use and specify how geometry is encoded; then briefly discuss which graph mining algorithm is used.

We define graphs on images as follows. Each interest point is represented by one vertex and its descriptor becomes the corresponding vertex label. We connect all vertices by undirected edges to obtain a completely connected graph. For each edge a temporary continuous-valued edge label vector $A = [a_1, a_2, a_3]$ is derived from its adjacent vertices' interest points p_1, p_2, where

$$\begin{aligned} a_1 &= \left|\log\left(p_1.scale/p_2.scale\right)\right|, \\ a_2 &= \frac{\|p_1.coords - p_2.coords\|}{\min\{p_1.scale, p_2.scale\}}, \\ a_3 &= \left|\sin\left(\max\left\{\text{atan2}\left(V, W\right), \text{atan2}\left(-V, -W\right)\right\}\right)\right| \\ V &= p_1.coords.y - p_2.coords.y \\ W &= p_1.coords.x - p_2.coords.x \end{aligned}$$

These attributes are used to encode the ratio of scales, a normalized distance and a horizontal orientation measure, respectively. For a set of graphs, each individual attribute is normalized across the set to zero mean and unit variance. The continuous attributes will be discretized.

Graph discretization. A per-class codebook is created from the continuous vertex and edge attributes using k-means clustering. The codebooks are concatenated such that one global vertex codebook and one global edge codebook is obtained. For each vertex, its continuous attribute is discretized by searching the nearest neighbor codeword in the vertex codebook. Let $d_i(x)$ denote the Euclidean distance of the i'th nearest neighbor codeword to x. If $d_1(x)/d_2(x) \geq \sigma$, we also assign the label of the second codeword to the node. For all our experiments $\sigma = 0.95$. The above is done analogously to discretize edge attributes.

Weighted graph mining. Yan and Han [19] describe *gSpan*, an efficient algorithm to mine frequent subgraphs for a given set of labeled connected graphs. We use the extended gSpan version of Kudo et al. [7] to solve Problem 2 for arbitrary weights.

3.5. 1.5-class Classification

For a discriminative classification task, the 2-class LPBoost formulation (2) is a good choice. But in a general object detection and classification setting on natural images we can have multiple objects within one image, such that the sample has multiple output labels. Additionally our training data might contain multiple objects per image or background clutter.

For such weakly- and multi-labeled data it is more appropriate to describe each object class using a one-class decision function. In a pure one-class setting a function $f : \mathcal{X} \to \mathbb{R}_+$ is trained using positive examples from one class such that $f(\boldsymbol{x})$ has a high output where $\boldsymbol{x}$ belongs to the positive class and a low output otherwise. One-class classifiers based on LPBoost, such as the formulation of Rätsch et al. [12] do not make use of known negative samples. Modifications of one-class classifiers incorporating

negative samples are known in the literature as "one class with negative examples" and rejection-classification problem [20]; here we call them "1.5-class" classifiers.

The boosting formulation (2) can be changed into a 1.5-class formulation by constraining the base hypotheses $h(\cdot; \boldsymbol{t}, \omega)$ to positive outputs. This way, the classification stumps cannot reward the absence of a feature as an indicator for a positive class decision. We set $\Omega = \emptyset$ such that the new stumps have the form

$$h(\boldsymbol{x}; \boldsymbol{t}) = \begin{cases} 1 & \boldsymbol{t} \subseteq \boldsymbol{x} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

The detailed derivation of our novel modification to the LPBoost algorithm is included in the supplementary materials.

3.6. Experiments and Results

We evaluate our supervised classification approach using the VOC2005 dataset [4]. We perform model selection by training on the training set ("train") and minimizing the test error on the validation set ("val"). The optimal parameters are then used to train one classifier on the combined ("train+val") set. This classifier is evaluated on the two test sets ("test1" and "test2"). We use the same experimental procedure as used in the official VOC challenge report [3].

To extract local image features we use the Harris-Laplace interest point operator and the SIFT feature descriptor [9, 10], extracted by the binaries provided by Mikolajczyk[4]. The interest point operator threshold is adjusted for each image such that a fixed number of features is extracted ($n = 80$, for all our experiments). The exact same features are used for all experiments. Often, a much larger number is used in the object classification literature. However, we want to focus on the relative classification accuracy of the methods and thus use a small number of features, such that we can still apply the graph mining exhaustively.

Relative comparison. To compare the relative performance of our method with another approach we also implemented the method of Zhang et. al [22], which performed best in the VOC2005 object classification challenge. In Zhang's approach each image is represented by a histogram of local image feature descriptors. The descriptors are assigned to histogram bins by means of a codebook generated using clustering, as described in Section 3.4. One Support Vector Machine is trained per class in a one-vs-rest fashion. The kernel function used is the χ^2-kernel [5] between two N-element normalized histograms h, h'.[5] Here, A is a normalization constant set to the mean of the χ^2-distances between all training samples.

[4] http://www.robots.ox.ac.uk/~vgg/research/affine/

[5] $K(h, h') = \exp\left(-\frac{1}{A}\left[\frac{1}{2}\sum_{\{n:h_n+h'_n>0\}}\frac{(h_n-h'_n)^2}{h_n+h'_n}\right]\right)$.

Results. The results for five different methods on the supervised classification experiment are shown in Table 2. The first, χ^2-SVM uses histogram features and has a codebook size of 1024 words. The 1.5-class graph boosting method uses $\nu = 0.4$ and a codebook of size 64 and 192 for nodes and edges, respectively; the 2-class graph boosting has $\nu = 0.15$ and codebook sizes of 48 and 96. The two item set boosting results use codebooks of size 512 and 384 and $\nu = 0.4$ and $\nu = 0.2$, respectively.

The results in the supervised object classification task are competitive to the χ^2-SVM approach. Generally and as expected the graph boosting is doing better than the item set boosting approach. Also the 1.5-class formulation is on average doing better than the 2-class LPBoosting. For the easier test set ("test1"), our boosting approach is consistently a bit worse than the χ^2-SVM one, but for the more difficult test set ("test2") the equal error rates of our 1.5-class graph boosting approach are on-par with the χ^2-SVM.

Analyzing the resulting classifiers, in the 1.5-class formulation, for both item set and graph boosting the number of active patterns in each one-vs-rest classifier, measured by $\sum_{\boldsymbol{t}} I(\alpha_{\boldsymbol{t}} \geq 10^{-6})$, is between 25-45.

3.7. Discussion

Considering the results of our supervised approach it is particularly interesting that we achieve respectable results using only a linear combination a small number of simple features. This is the case because our formulation is able to explicitly select discriminative features from a very large feature space. The theoretical number of possible item sets is $O(2^n)$, where n is the number of codewords, whereas the limit for the theoretical number of possible subgraphs is even larger.

The absolute EER results of the χ^2-SVM approach are below the ones reported in the VOC report [3], but we use a more sparse representation of only 80 features per image in order to make a relative comparison of the approaches; in comparison, for the results reported in [3] Zhang et al. [22] used an average of over 3000 features per image. For such a large number of features, the resulting completely-connected graphs are too large to be mined exhaustively with current graph mining techniques.

4. Conclusion

Object classification in natural images, supervised or unsupervised, is a remarkably difficult task. The best performing approaches from the literature based on non-linear SVMs or sophisticated probabilistic models do not offer an accessible interpretation of the model parameters. In this paper we proposed a way to bridge the gap between high prediction performance and interpretability.

The contribution of this paper is threefold; first, we have

	ROC Equal Error Rates							
	Test set 1, classes				Test set 2, classes			
Method	1	2	3	4	1	2	3	4
χ^2 SVM	0.829	0.728	0.738	0.858	0.665	0.631	0.599	0.687
Graph Boosting, 1.5-class	0.806	0.690	0.668	0.807	0.662	0.621	0.621	0.634
Graph Boosting, 2-class	0.764	0.663	0.679	0.775	0.643	0.564	0.596	0.649
Item Set Boosting, 1.5-class	0.764	0.649	0.655	0.786	0.643	0.569	0.584	0.673
Item Set Boosting, 2-class	0.745	0.647	0.679	0.812	0.615	0.574	0.601	0.667

Table 2. Results for the VOC 2005 data set. The classes are motorbike (1), bike (2), person (3) and car (4). Test set 1 is from the same distribution as the training set, test set 2 is much more difficult. For all results above, 80 Harris-Laplace features per image have been used.

shown that weighted pattern mining algorithms are well suited for cluttered image data because they are able to ignore non-discriminate or non-frequent parts. For suitable patterns they are both efficient and allow interpretability. Second, we derived and validated two practical methods for unsupervised ranking and supervised object classification based on the LPBoost formulation. Third, we introduced and experimentally validated a 1.5-class generalization to 1-class ν-LPBoosting.

In the future, we plan to overcome the limitations of a sparse image representation by using a densely sampled representation efficiently with weighted item set mining in order to scale the approach to more images and video data.

Acknowledgments. We would like to thank Mario Fritz for providing the keyword web-query image ranking data set. We are thankful for the helpful comments by all reviewers. This work is funded in part by the EU CLASS project, IST 027978.

References

[1] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *23rd International Conference on Data Engineering*, 2007.

[2] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Journal of Machine Learning*, 46:225–254, 2002.

[3] M. Everingham, L. V. Gool, C. Williams, and A. Zisserman. Pascal visual object classes challenge results. Technical report, 2005.

[4] M. Everingham, A. Zisserman, C. K. I. Williams, and L. J. V. G. *et al*. The 2005 PASCAL visual object classes challenge. In J. Q. Candela, I. Dagan, B. Magnini, and F. d'Alché Buc, editors, *MLCW*, volume 3944 of *Lecture Notes in Computer Science*, pages 117–176. Springer, 2005.

[5] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nyström method. *PAMI*, 26(2):214–225, 2004.

[6] M. Fritz and B. Schiele. Towards unsupervised discovery of visual categories. 2006. DAGM, Berlin.

[7] T. Kudo, E. Maeda, and Y. Matsumoto. An application of boosting to graph classification. In *NIPS*, 2004.

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[9] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, Oct. 2004.

[10] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, Oct. 2005.

[11] T. Quack, V. Ferrari, and L. J. V. Gool. Video mining with frequent itemset configurations. In H. Sundaram, M. R. Naphade, J. R. Smith, and Y. Rui, editors, *CIVR*, volume 4071 of *Lecture Notes in Computer Science*, pages 360–369. Springer, 2006.

[12] G. Rätsch, B. Schölkopf, S. Mika, and K.-R. Müller. SVM and boosting: One class. Technical report, 2000.

[13] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, pages 1605–1614. IEEE Computer Society, 2006.

[14] B. Schölkopf and A. J. Smola. *Learning with Kernels, 2nd Edition*. MIT Press, 2002. ISBN 0-262-19475-9.

[15] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, pages 994–1000, 2005.

[16] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *CVPR*, pages 488–495, 2004.

[17] A. B. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *CVPR*, pages 762–769, 2004.

[18] T. Uno, T. Asai, Y. Uchida, and H. Arimura. LCM: An efficient algorithm for enumerating frequent closed item sets. In B. Goethals and M. J. Zaki, editors, *FIMI*, volume 90 of *CEUR Workshop Proceedings*, 2003.

[19] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *ICDM*, pages 721–724, 2002.

[20] C. Yuan and D. Casasent. A novel support vector classifier with better rejection performance. In *CVPR*, pages 419–424, 2003.

[21] 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.

[22] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: An in-depth study. In *INRIA*, 2005.