Scene Discovery by Matrix Factorization

Nicolas Loeff and Ali Farhadi

University of Illinois at Urbana-Champaign,
Urbana, IL, 61801
{loeff,afarhad2}@uiuc.edu

Abstract. What constitutes a scene? Defining a meaningful vocabulary for scene discovery is a challenging problem that has important consequences for object recognition. We consider scenes to depict correlated objects and present visual similarity. We introduce a max-margin factorization model that finds a low dimensional subspace with high discriminative power for correlated annotations. We postulate this space should allow us to discover a large number of scenes in unsupervised data; we show scene discrimination results on par with supervised approaches. This model also produces state of the art word prediction results including good annotation completion.

1 Introduction

Classification of scenes has useful applications in content-based image indexing and retrieval and as an aid to object recognition (improving retrieval performance by removing irrelevant images). Even though a significant amount of research has been devoted to the topic, the questions of what constitutes a scene has not been addressed. The task is ambiguous because of the diversity and variability of scenes but also mainly due to the subjectivity of the task. Just like in other areas of computer vision such as activity recognition, it is not simple to define the vocabulary to label scenes. Thus, most approaches have used the physical setting where the image was taken to define the scene (e. g. beach, mountain, forest, etc.).

Previous work is focused on supervised approaches. It is common to use techniques that do not share knowledge between scene types. For instance, In [12] Lazebnik proposes a pyramid match kernel on top of SIFT features to measure image similarity and applies it to classification of scenes using an SVM. Chapelle et al. [6] use global color histograms and an SVM classifier.

Therefore other models build intermediate representations, usually as a bag of features, in order to perform classification. Internal representations let classifiers share features between scene classes. Quelhas and Odobez [19] propose a scene representation using mixtures of local features. Fei-Fei and Perona [13] use a modified Latent Dirichlet Allocation model on bags of patches to create a topic representation of scenes. Scenes are also directly labeled during training. Liu and Shah [14] use maximization of mutual information between bags of features and intermediate concepts to create an internal representation. These intermediate concepts are purely appearance based. On top of it, they run a *supervised* SVM classifier. Bosch et al. [3] uses a pLSA model on

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part IV, LNCS 5305, pp. 451–464, 2008.

top of bags of features to discover intermediate visual representations and a *supervised* KNN classifier to identify scenes.

Other approaches first manually define a vocabulary for the internal representation and then try to learn it. J. C. van Gemert et al. [22] describe scenes using "proto-concepts" like vegetation, sky and water, and learning using image statistics and context. Vogel and Schiele [24] manually label 9 different intermediate "concepts" (e. g. water, sky, foliage) and learn a KNN classifier on top of this representation. Oliva and Torralba [17] use global "gist" features and local spatial constraints, plus human labeled *intermediate* properties (such as "roughness" or "openness") as an intermediate representation.

We propose a different strategy. First, we aim to find scenes without supervision. Second, we treat the building of the internal representation not as separate from a classification task, but as interdependent processes that must be learnt together.

What is a scene? In current methods, visual similarity is used to classify scenes into a known set of types. We expect there are many types of scene, so that it will be hard to write down a list of types in a straightforward way. We should like to build a vocabulary of scene types from data. We believe that two images depict the same scene category if:

1. Objects that appear in one image could likely appear in the other
2. The images look similar under an appropriate metric.

This means one should be able to identify scenes by predicting the objects that are likely to be in the image, or that tend to co-occur with objects that are in the image. Thus, if we could estimate a list of all the annotations that could reasonably be attached to the image, we could cluster using that list of annotations. The objects in this list of annotations don't actually have to be present – not all kitchens contain coffee makers – but they need to be plausible hypotheses. We would like to predict hundreds of words for each of thousands of images. To do so, we need stable features and it is useful to exploit the fact that annotating words are correlated.

All this suggests a procedure akin to collaborative filtering. We should build a set of classifiers, that, from a set of image features, can predict a set of word annotations that are like the original annotations. For each image, the predicted annotations will include words that annotators may have omitted, and we can cluster on the completed set of annotations to obtain scenes. We show that, by exploiting natural regularization of this problem, we obtain image features that are stable and good at word prediction. Clustering with an appropriate metric in this space is equivalent to clustering on completed annotations; and the clusters are scenes.

We will achieve this goal by using matrix factorization [21,1] to learn a word classifier. Let Y be a matrix of word annotations per image, X the matrix of image features per image, and W a linear classifier matrix, we will look for W to minimize

$$J(W) = \text{regularization}(W) + \text{loss}(Y, W^t X) \tag{1}$$

The regularization term will be constructed to minimize the rank of W, in order to improve generalization by forcing word classifiers to **share a low dimensional representation**. As the name "matrix factorization" indicates, W is represented as the product

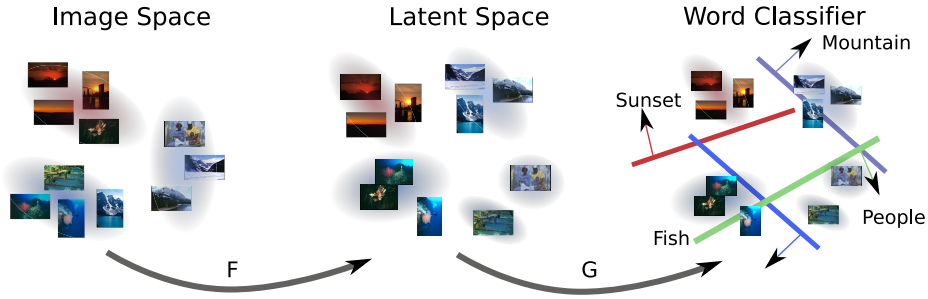

Fig. 1. Matrix factorization for word prediction. Our proxy goal is to find a word classifier W on image features X. W factorizes into the product $W = FG$. We regularize with the rank of W; this makes $F^t X$ a **low-dimensional feature space** that maximizes word **predictive** power. In this space, where correlated words are mapped close, we learn the classifiers G.

between two matrices FG. This factorization learns a feature mapping (F) with shared characteristics between the different words. This latent representation should be a good space to learn correlated word classifiers G (see figure 1).

Our problem is related to multi-task learning as clearly the problem of assigning one word to an image is correlated with the other words. In a related approach [2] Ando and Zhang learn multiple classifiers with a shared structure, alternating fixing the structure and learning SVM classifiers and fixing the classifiers to learn the structure using SVD. Ando and Zhang propose an interesting insight into the problem: instead of doing dimensionality reduction on the data space (like PCA), they do it in the classifier space. This means the algorithm looks for low-dimensional structures with good predictive, rather than descriptive, power. This leads to an internal representation where the tasks are easier to learn. This is a big conceptual difference with respect to approaches like [14,3]. It is also different from the CRF framework of [20], where pairwise co-occurrence frequencies are modeled.

Quattoni et al. [18] proposed a method for supervised classification of topics using auxiliary tasks, following [2]. In contrast, our model we discover scenes without supervision. We also differ in that [18] first learns word classifiers, fixes them, and then finds the space for the topic (scene) prediction. We learn both the internal structure and the classifiers simultaneously, in a convex formulation. Thus our algorithm is able to use correlation between words not only for the scene classification task but also for word prediction. This results in improved word prediction performance. In section **??** we show the model also produces better results than [18] for the scene task, even without having the scene labels!

2 A Max-Margin Factorization Model

Consider a set of N images $\{x_i\}$, each represented by a d-dimensional vector, and M learning tasks which consist in predicting the word $y_i^m \in \{-1, 1\}$ for each image using a linear classifier $w_m^t x_i$. This can be represented as $Y \sim W^t X$ for a matrix

$Y \in \{\pm 1\}^{M \times N}$ where each column is an image and each row a word, $W \in \mathbb{R}^{d \times M}$ is the classifier matrix and $X \in \mathbb{R}^{d \times N}$ the observation matrix. We will initially consider that the words are decoupled (as in regular SVMs), and use the L_2 regularization $\sum_m ||w_m||_2^2 = ||W||_F^2$ (known as the Frobenius norm of W). A suitable loss for a max-margin formulation is the hinge function $h(z) = \max(0, 1 - z)$. The problem can then be stated as

$$\min_W \frac{1}{2} ||W||_F^2 + C \sum_{i=1}^{N} \sum_{m=1}^{M} \Delta(y_i^m) h(y_i^m \cdot (w_m^t x_i)) \tag{2}$$

where C is the trade-off constant between data loss and regularization, and Δ is a slack re-scaling term we introduce to penalize errors differently: false negatives $\Delta(1) = 1$ and false positives $\Delta(-1) = \epsilon < 1$. The rationale is that missing word annotations are much more common than wrong annotation for this problem.

Our word prediction formulation of the loss is different from [21] (a pure collaborative filtering model) and [1] (a multi-class classifier), even though our tracenorm regularization term is similar to theirs. Our formulation is, to the best of our knowledge, the first application of the tracenorm regularization to a problem of these characteristics. From [1] we took the optimization framework, although we are using different losses and approximations and we are using BFGS to perform the minimization. Finally, we introduce a unsupervised model on top of the internal representation this formulation produces to discover scenes.

Matrix Factorization: In order to exploit correlations in the words, an alternative problem is to factor the matrix $W = FG$ where $F \in \mathbb{R}^{d \times k}$ can be interpreted as a *mapping* of the features X into a k dimensional *latent* space and $G \in \mathbb{R}^{k \times M}$ is a linear classifier on this space (i. e. $Y \sim G^t(F^t X)$). Regularization is provided by constraining the dimensionality of the latent space (k) and penalizing the Frobenius norm of F and G [21]. The minimization in F and G is unfortunately non-convex, and Rennie suggested using the tracenorm (the minimum of the possible sum of Frobenius norms so that $W = FG$) as an alternative regularization. As the tracenorm may also be written as $||W||_\Sigma = \sum_l |\gamma_l|$ (where γ_l is the $l-$th singular value), tracenorm minimization can be seen as minimizing the L_1 norm of the singular values of W. This **leads to a low-rank solution**, in which correlated words share features, while the Frobenius norm of W (which minimizes the L_2 norm of the singular values) assumes the words are independent.

Minimization is now with respect to W directly, and the problem is convex. Moreover, the dimensionality k doesn't have to be provided.

$$\min_W \frac{1}{2} ||W||_\Sigma + C \sum_{i=1}^{N} \sum_{m=1}^{M} \Delta(y_i^m) h(y_i^m \cdot (w_m^t x_i)) \tag{3}$$

Rennie [21] showed (3) can be recast as a Semidefinite Program (SDP). Unfortunately, SDPs don't scale nicely with the number of dimensions of the problem, making any decent size problem intractable. Instead, he proposed gradient descent optimization.

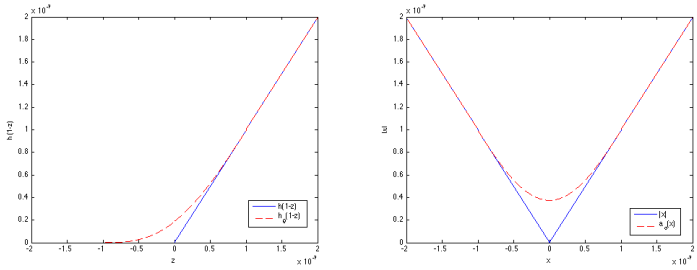

Fig. 2. Smooth approximations of the hinge function (left) and absolute value function (right), used in the gradient descent optimization

2.1 Gradient Based Optimization

Equation 3 is not differentiable due to the hinge loss and the tracenorm, but the equation can be approximated to arbitrary precision by a smoothed version. This allows to perform gradient based optimization. We will consider a smooth approximation $h_\rho(z)$ of the hinge loss $h(z)$ that is exact for $|1-z| \geq \rho$, and is twice differentiable everywhere:

$$h(1-z) \approx h_\rho(1-z) = \begin{cases} z & z > \rho \\ \frac{-z^4}{16\rho^3} + \frac{3z^2}{8\rho} + \frac{3z}{2} + \frac{3\sigma}{16} & |z| \leq \rho \\ 0 & z < -\rho \end{cases} \tag{4}$$

For the tracenorm we use $||W||_\Sigma \approx ||W||_S = \sum_l a_\sigma(\gamma_l)$, where the smoothed absolute value is again exact for $|x| \geq \sigma$ and is twice differentiable everywhere,

$$a_\sigma(x) = \begin{cases} |x| & |x| > \sigma \\ \frac{-z^4}{8\sigma^3} + \frac{z^2}{4\sigma} + \frac{3\sigma}{8} & |x| \leq \sigma \end{cases} \tag{5}$$

In our experiments we use $\rho = \sigma = 10^{-7}$. Plots for both approximation are depicted in figure 2.

We will then consider the smooth cost

$$J(W; Y, X, \sigma, \rho) = J_R(W; \sigma) + C \cdot J_D(W; Y, X, \rho) \tag{6}$$

where the regularization cost is

$$J_R(W, \sigma) = ||W||_S \tag{7}$$

and the data loss term is

$$J_D(W; Y, X, \rho) = \sum_{i=1}^{N} \sum_{m=1}^{M} \Delta(y_i^m) h_\rho(y_i^m \cdot (w_m^t x_i)) \tag{8}$$

Using the SVD decomposition $W = UDV^t$,

$$\frac{\partial J_R}{\partial W} = U a'_\sigma(D) V^t \tag{9}$$

The gradient of the data loss term is

$$\frac{\partial J_D}{\partial W} = -(\Delta(Y) \cdot h'_\rho(Y \cdot W^t X))^t (Y \cdot X) \tag{10}$$

where $(A \cdot B)$ is the Hadamard or element-wise product: $(A \cdot B)_{ij} = a_{ij} b_{ij}$. Exact second order Newton methods cannot be used because of the size of the Hessian, so we use limited-memory BFGS for minimization.

2.2 Kernelization

A interesting feature of problem 3 is that it admits a solution when high dimensional features X are not available but instead the Gram matrix $K = X^t X$ is provided. Theorem 1 in [1] can be applied with small modifications to prove that there exists a matrix $\alpha \in \mathbb{R}^{M \times N}$ so that the minimizer of (3) is $W = X\alpha$. But instead of solving the dual Lagrangian problem we will use this representation of W to minimize the primal problem (actually, it's smoothed version) using gradient descent. The derivatives in terms of K and α only become

$$\frac{\partial J_R}{\partial \alpha} = \frac{\partial \, ||X\alpha||_S}{\partial \alpha} = \frac{X^t \partial \, ||X\alpha||_S}{\partial X\alpha} = K\alpha V D^{-1} a'_\sigma(D) V^t \tag{11}$$

using that $D(VV^t)D^{-1} = I$, $X\alpha = UDV^t$, and that $K = X^t X$. The gradient of the data loss term is

$$\frac{\partial J_D}{\partial W} = -K * (\Delta(Y) \cdot h'_\rho(\alpha^t K \alpha) \cdot Y) \tag{12}$$

3 Scene Discovery – Analysing the Latent Representation

Section 2.1 introduced a smooth approximation to the convex problem 3. After convergence we obtain the classification matrix W. The solution does not provide the factorization $W = FG$. Moreover, any decomposition $W = FG$ is not unique as a full rank transformation $\tilde{F} = FA$, $\tilde{G} = A^{-1}G$ will produce the same W.

What is a good factorization then? As discussed in the section 1 clustering in the latent space should be similar to clustering the word predictions. Since we define scenes as having correlated words, a good factorization of W should maximally transfer the correlation between the predicted words $\left((W^t X)^t (W^t X)\right)$ to the correlation in the latent space $\left((A^t F^t X)^t (A^t F^t X)\right)$. Identifying terms, $A = (GG^t)^{1/2}$. In this space $(A^t F^t X)$, images with correlated words (i. e. belonging to the same scene category) should cluster naturally.

For the factorization of W we will use a truncated SVD decomposition and then we will use this A. We will measure their similarity of images in this space using the cosine distance.

4 Experiments

To demonstrate the performance of our scene discovery model we need a dataset with multiple object labels per image. We chose the standard subset of the Corel image collection [7] as our benchmark dataset. This subset has been extensively used and

consists of 5000 images grouped in 50 different sets (CDs). These images are separated into 4500 training and 500 test images. The vocabulary size of this dataset is 374, out of which 371 appear in train and 263 in test set. The annotation length varies from 1 to 5 words per image.

We employ features used in the PicSOM [23] image content analysis framework. These features convey image information using 10 different, but not necessarily uncorrelated, feature extraction methods. Feature vector components include: DCT coefficients of average color in 20x20 grid (analogous to MPEG-7 ColorLayout feature), CIE LAB color coordinates of two dominant color clusters, 16×16 FFT of Sobel edge image, MPEG-7 EdgeHistogram descriptor, Haar transform of quantised HSV color histogram, three first central moments of color distribution in CIE LAB color space, average CIE LAB color, co-occurence matrix of four Sobel edge directions, histogram of four Sobel edge directions and texture feature based on relative brightness of neighboring pixels.

The final image descriptor is a 682 dimensonal vector. We append a constant value 1 to each vector to learn a threshold for our linear classifiers.

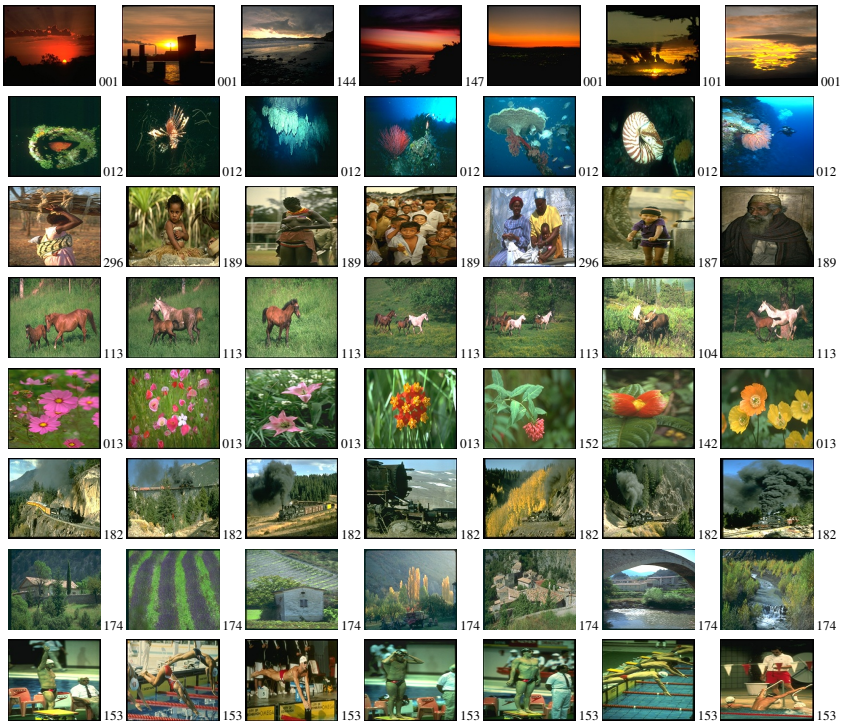

Fig. 3. Example **clustering results** on the Corel training set. Each row consists of the closest images to the centroid of a different cluster. The number on the right of each image is the Corel CD label. The algorithm is able to discover scenes even when there is high visual variability in the images (e. g. *people* cluster, *swimmers*, CD-174 cluster). Some of the scenes (e. g. *sunsets*, *people*) clearly depict scenes, even if the images are come from different CDs. (For display purposes, portrait images were resized)

Scene discovery. First, we explore the latent space described in section 3. As mentioned there, the cosine distance is natural to represent dissimilarity in this space. To be able to use it for clustering we will employ graph-based methods. We expect scene clusters to be compact and thus use complete link clustering. We look initially for many more clusters than scene categories, and then remove clusters with a small number of images allocated to them. We reassign those images to the remaining clusters using the closest 5 nearest neighbors. This produced approximately 1.5 clusters per CD label. For the test set we use again the 5 nearest neighbors to assign images to the train clusters. As shown in figure 3, the algorithm found highly plausible scene clusters, even in the presence of

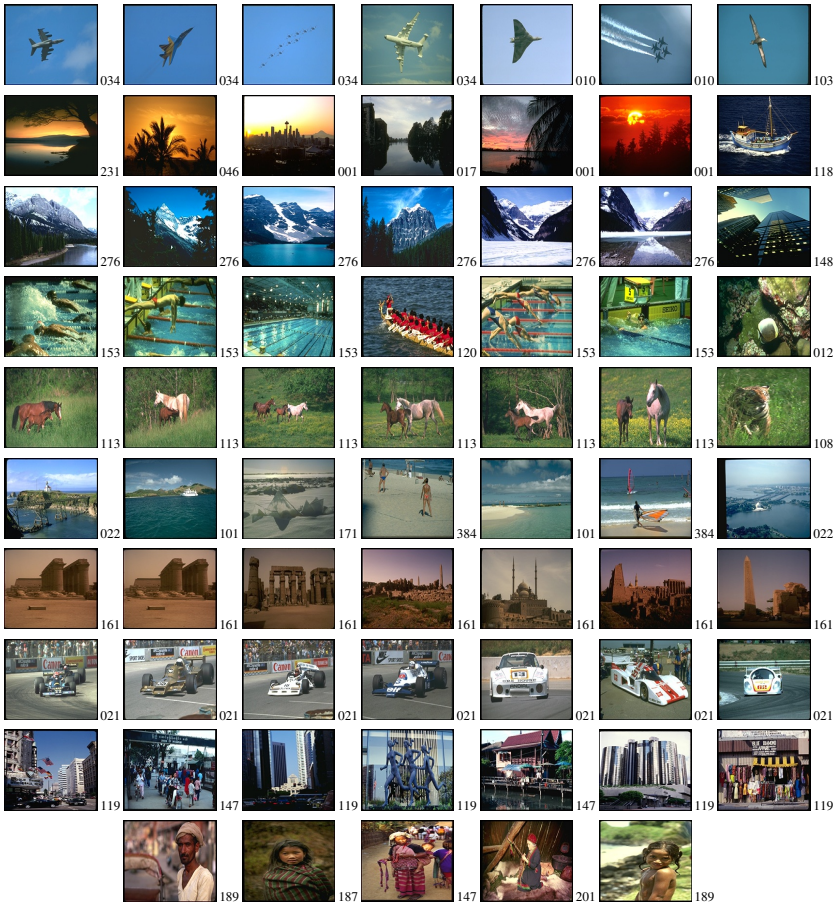

Fig. 4. Example **results on the Corel test set**. Each row consists of the closest 7 **test** images to each centroid found on the **training** set. The number on the right of each image is the Corel CD label. Rows correspond to scenes, which would be hard to discover with pure visual clustering. Because our method is able to predict word annotations while clustering scenes, it is able to discount large but irrelevant visual differences. Despite this, some of mistakes are due to visual similarity (e. g. the bird in the last image of the *plane* cluster, or the skyscraper in the last image of the *mountain* cluster). (For displaying purposes, portrait images were resized).

large visual variability. This is due to the fact that these images depict objects that tend to appear together. The algorithm also generalizes well: when the clusters were transfered to the test set it still produced a good output (see figure 4).

Word prediction. Our approach to scene discovery is based on the internal representation of the word classifier, so these promising results suggest a good word annotation prediction performance. Table 1 shows the precision, recall and F1-measure of our word prediction model is competitive with the best state-of-the-art methods using this dataset. Changing the value of ϵ in equation 3 traces out the precision-recall curve; we show the equal error rate ($P = R$) result. It is remarkable that the kernelized classifier does not provide a substantial improvement over the linear classifier. The reason for this may lie in the high dimensionality of the feature space, in which all points are roughly at the same distance. In fact, using a standard RBF kernel produced significantly lower results; thus the sigmoid kernel, with a broarder support, performed much better. Because to this and the higher computational complexity of the kernelized classifier, we will use the linear classifier for the rest of the experiments.

The **influence of the tracenorm regularization** is clear when the results are compared to independent linear SVMs on the same features (that corresponds to using the Frobenius norm regularization, equation 2). The difference in performance indicates

Table 1. Comparison of the performance of our word annotation prediction method with that of Co-occurance model (Co-occ), Translation Model (Trans), Cross-Media Relevance Model (CMRM), Text space to image space (TSIS), Maximum Entropy model (MaxEnt), Continuous Relevance Model (CRM), 3×3 grid of color and texture moments (CT-3×3), Inference Network (InfNet), Multiple Bernoulli Relevance Models (MBRM), Mixture Hierarchies model (MixHier), PicSOM with global features, and linear independent SVMs on the same features. The performance of our model is provided for the linear and kernelized (sigmoid) classifiers.* *Note: the results of the PicSOM method are not directly comparable as they limit the annotation length to be at most five (we do not place this limit as we aim to complete the annotations for each image).*

Method	P	R	F1	Ref
Co-occ	0.03	0.02	0.02	[16]
Trans	0.06	0.04	0.05	[7]
CMRM	0.10	0.09	0.10	[9]
TSIS	0.10	0.09	0.10	[5]
MaxEnt	0.09	0.12	0.10	[10]
CRM	0.16	0.19	0.17	[11]
CT-3×3	0.18	0.21	0.19	[25]
CRM-rect	0.22	0.23	0.23	[8]
InfNet	0.17	0.24	0.23	[15]
Independent SVMs	0.22	0.25	0.23	
MBRM	0.24	0.25	0.25	[8]
MixHier	0.23	0.29	0.26	[4]
This work (Linear)	0.27	0.27	0.27	
This work (Kernel)	0.29	0.29	0.29	
PicSOM	0.35^*	0.35^*	0.35^*	[23]

Fig. 5. Example **word completion results**. Correctly predicted words are below each image in blue, predicted words not in the annotations ("False Positives") are *italic red*, and words not predicted but annotated ("False Negatives") are in green. Missing annotations are not uncommon in the Corel dataset. Our algorithm performs scene clustering by predicting all the words that *should* be present on an image, as it learns correlated words (e. g. images with *sun* and *plane* usually contain *sky*, and images with *sand* and *water* commonly depict *beaches*). Completed word annotations are a good guide to scene categories while original annotations might not be; this indicates visual information really matters.

the sharing of features among the word classifiers is beneficial. This is specially true for words that are less common.

Annotation completion. The promising performance of the approach results from its generalization ability; this in turn lets the algorithm predict words that are not annotated in the training set but *should* have been. Figure 5 shows some examples of word completion results. It should be noted that performance evaluation in the Corel dataset is delicate, as missing words in the annotation are not uncommon.

Discriminative scene prediction. The Corel dataset is divided into sets (CDs) that do not necessarily depict different scenes. As it can be observed in figure 3, some correctly clustered scenes are spread among different CD labels (e. g. *sunsets, people*). In order to evaluate our unsupervised scene discovery, we selected a subset of 10 out of the 50 CDs from the dataset so that the CD number can be used as a reliable proxy for scene labels. The subset consists of CDs: 1 (sunsets), 21 (race cars), 34 (flying airplanes), 130 (african animals), 153 (swimming), 161 (egyptian ruins), 163 (birds and nests), 182 (trains), 276 (mountains and snow) and 384 (beaches). This subset has visually very disimlar pictures with the same labels and visually similar images (but depicting different objects) with different labels. The train/test split of [7] was preserved.

To evaluate the performance of the *unsupervised* scene discovery method, we label each cluster with the most common CD label in the **training** set and then evaluate the scene detection performance in the **test** set. We compare our results with the same clustering thechnique on the image features directly. In this space the cosine distance losses

Table 2. Comparison of the performance of our scene discovery on the latent space with another unsupervised method and four supervised methods on image features directly. Our model produced significantly better results that the unsupervised method on the image features, and is only surpassed by the supervised kernelized SVM. For both unsupervised methods, clustering is done on the train set and performance is measured on the test set (see text for details).

Method	Accuracy
Unsupervised Latent space (this work)	0.848
Unsupervised Image features clustering	0.697
Supervised Image features KNN	0.848
Supervised Image features SVM (linear)	0.798
Supervised Image features SVM (kernel)	0.948
Supervised "structural learning" [2,18]	0.818

its meaning and thus we use the euclidean distance. We also computed the performance of two *supervised* approaches on the image features: k nearest neighbors (KNN), support vector machines (SVM), and "structural learning" (introduced in [2] and used in a vision application -Reuters image classification- in [18]). We use a one-vs-all approach for the SVMs. Table 2 show the the latent space is indeed a suitable space for scene detection: it clearly outperforms clustering on the original space, and only the supervised SVM using a kernel provides an improvement over the performance of our method.

The difference with [18] deserves further exploration. Their algorithm classifies *topics* (in our case scenes) by first learning a classification of *auxiliary tasks* (in this case words), based in the framework introduced in [2]. [18] starts by building *independent*

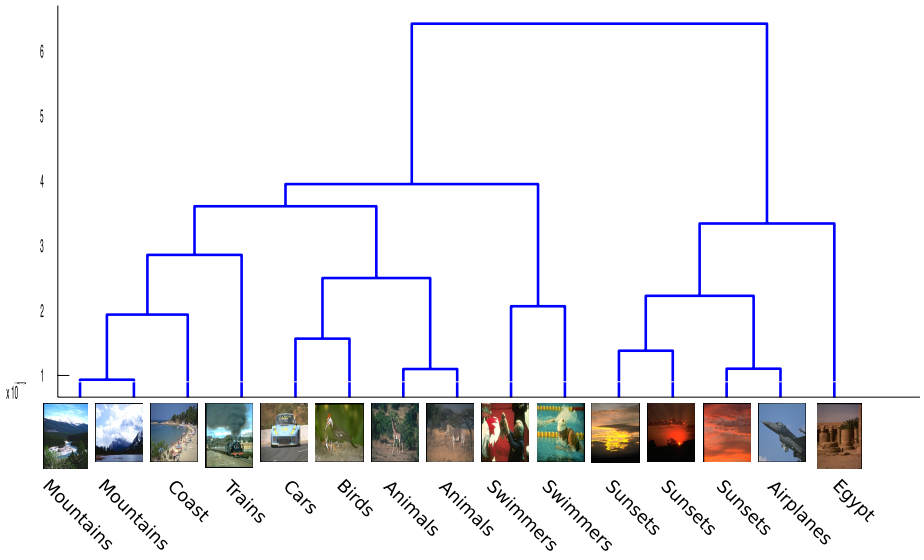

Fig. 6. Dendrogram for our clustering method. Our scene discovery model produces 1.5 *proto-scenes* per scene. Clusters belonging to the same scene are among the first to be merged

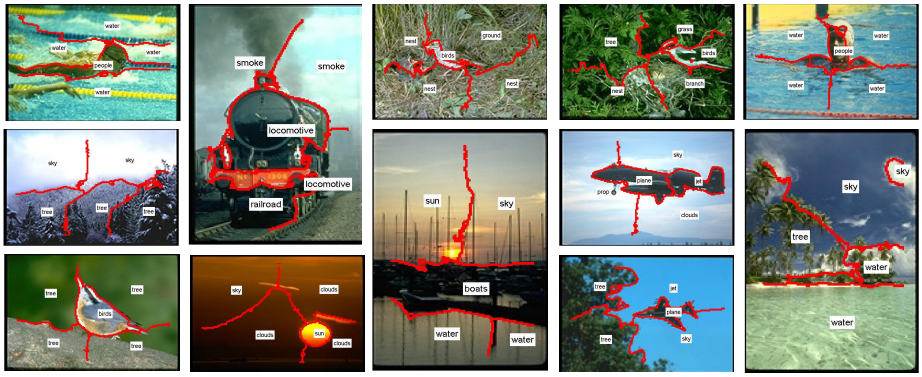

Fig. 7. Future work includes unsupervised **region annotation**. Example images show promising results for region labeling. Images are presegmented using normalized cuts (red lines), features are computed in each region and fed to our classifier as if they were whole image features.

SVM classifiers on the auxiliary tasks/words. As we showed in table 1, this leads to lower performance in word classification when compared to our *correlated* classifiers. On top of this [18] runs an SVD to correlate the output of the classifiers. It is remarkable that our algorithm provides a slight performance advantage despite the fact [18] is supervised and learns the topic classifier directly, whereas our formulation is unsupervised and does not use topic labels.

Figure 4 depicts a dendrogram of the complete-link clustering method applied to the clusters found by our scene discovery algorithm. As expected clusters belonging to the same scene are among the first to be merged together. The exception is a *sunset* cluster that is merged with an *airplane* cluster before being merged with the rest of the *sunset* clusters. The reason for this is that both cluster basically depict images where the sky occupies most of the image. Is is pleasing that "scenery" clusters depicting mountains and beaches are merged together with the train cluster (also depicts panoramic views); the *birds* and *animals* clusters are also merged together.

5 Conclusions

Scene discovery and classification is an important and challenging task that has important applications in object recognition. We have introduced a principled way of defining a meaningful vocabulary of what constitutes a scene. We consider scenes to depict correlated objects and present visual similarity. We introduced a max-margin factorization model to learn these correlations. The algorithm allows for scene discovery on par with supervised approaches even without explicitly labeling scenes, producing highly plausible scene clusters. This model also produced state of the art word annotation prediction results including good annotation completion.

Future work will include using our classifier for weakly supervised region annotation/labeling. For a given image, we use normalized cuts to produce a segmentation.

Using our classifier, we know what words describe the image. We then *restrict* our classifier to these word subsets and to the features in each of the regions. Figure 7 depicts examples of such annotations. These are promising preliminary results; since quantitative evaluation of this procedure requires having a ground truth labels for each segment, we only show qualitative results.

Acknowledgements

The authors would like to thank David Forsyth for helpful discussions.

This work was supported in part by the National Science Foundation under IIS - 0534837 and in part by the Office of Naval Research under N00014-01-1-0890 as part of the MURI program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation or the Office of Naval Research.

References

1. Amit, Y., Fink, M., Srebro, N., Ullman, S.: Uncovering shared structures in multiclass classification. In: ICML, pp. 17–24 (2007)
2. Ando, R.K., Zhang, T.: A high-performance semi-supervised learning method for text chunking. In: ACL (2005)
3. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via plsa. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
4. Carneiro, G., Vasconcelos, N.: Formulating semantic image annotation as a supervised learning problem. In: CVPR, vol. 2, pp. 163–168 (2005)
5. Celebi, E., Alpkocak, A.: Combining textual and visual clusters for semantic image retrieval and auto-annotation. In: 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology, 30 November - 1 December 2005, pp. 219–225 (2005)
6. Chapelle, O., Haffner, P., Vapnik, V.: SVMs for histogram-based image classification. IEEE Transactions on Neural Networks, special issue on Support Vectors (1999)
7. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
8. Feng, S.L., Manmatha, R., Lavrenko, V.: Multiple bernoulli relevance models for image and video annotation. In: CVPR, vol. 02, pp. 1002–1009 (2004)
9. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: SIGIR, pp. 119–126 (2003)
10. Jeon, J., Manmatha, R.: Using maximum entropy for automatic image annotation. In: Enser, P.G.B., Kompatsiaris, Y., O'Connor, N.E., Smeaton, A.F., Smeulders, A.W.M. (eds.) CIVR 2004. LNCS, vol. 3115, pp. 24–32. Springer, Heidelberg (2004)
11. Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: NIPS (2003)
12. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
13. Li, F.-F., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR, vol. 2, pp. 524–531 (2005)

14. Liu, J., Shah, M.: Scene modeling using co-clustering. In: ICCV (2007)
15. Metzler, D., Manmatha, R.: An inference network approach to image retrieval. In: Enser, P.G.B., Kompatsiaris, Y., O'Connor, N.E., Smeaton, A.F., Smeulders, A.W.M. (eds.) CIVR 2004. LNCS, vol. 3115, pp. 42–50. Springer, Heidelberg (2004)
16. Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: Proc. of the First International Workshop on Multimedia Intelligent Storage and Retrieval Management (1999)
17. Oliva, A., Torralba, A.B.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision 42(3), 145–175 (2001)
18. Quattoni, A., Collins, M., Darrell, T.: Learning visual representations using images with captions. In: CVPR (2007)
19. Quelhas, P., Odobez, J.-M.: Natural scene image modeling using color and texture visterms. Technical report, IDIAP (2006)
20. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV (2007)
21. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: ICML, pp. 713–719 (2005)
22. van Gemert, J.C., Geusebroek, J.-M., Veenman, C.J., Snoek, C.G.M., Smeulders, A.W.M.: Robust scene categorization by learning image statistics in context. In: CVPRW Workshop (2006)
23. Viitaniemi, V., Laaksonen, J.: Evaluating the performance in automatic image annotation: Example case by adaptive fusion of global image features. Image Commun. 22(6), 557–568 (2007)
24. Vogel, J., Schiele, B.: Natural scene retrieval based on a semantic modeling step. In: CIVR, pp. 207–215 (2004)
25. Yavlinsky, A., Schofield, E., Rger, S.: Automated image annotation using global features and robust nonparametric density estimation. In: Leow, W.-K., Lew, M., Chua, T.-S., Ma, W.-Y., Chaisorn, L., Bakker, E.M. (eds.) CIVR 2005. LNCS, vol. 3568, pp. 507–517. Springer, Heidelberg (2005)