Int J Comput Vis
DOI 10.1007/s11263-009-0271-8

Unsupervised Object Discovery: A Comparison

Tinne Tuytelaars · Christoph H. Lampert · Matthew B. Blaschko · Wray Buntine

Received: 28 July 2008 / Accepted: 6 July 2009

Abstract The goal of this paper is to evaluate and compare models and methods for learning to recognize basic entities in images in an unsupervised setting. In other words, we want to discover the objects present in the images by analyzing unlabeled data and searching for re-occurring patterns. We experiment with various baseline methods, methods based on latent variable models, as well as spectral clustering methods. The results are presented and compared both on subsets of Caltech256 and MSRC2, data sets that are larger and more challenging and that include more object classes than what has previously been reported in the literature. A rigorous framework for evaluating unsupervised object discovery methods is proposed.

Keywords Object discovery · Unsupervised object recognition · Evaluation

T. Tuytelaars (✉)
ESAT-PSI, K.U. Leuven, Kasteelpark Arenberg 10, bus 2241, Leuven 3001, Belgium
e-mail: Tinne.Tuytelaars@esat.kuleuven.be

C.H. Lampert · M.B. Blaschko
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

C.H. Lampert
e-mail: Christoph.Lampert@tuebingen.mpg.de

M.B. Blaschko
University of Oxford, Oxford, UK
e-mail: blaschko@robots.ox.ac.uk

W. Buntine
NICTA, Canberra, Australia
e-mail: Wray.Buntine@nicta.com.au

W. Buntine
Australian National University, Canberra, Australia

1 Introduction

Over the last decade, the category-level object recognition problem has drawn considerable attention within the computer vision research community and as a result, much progress has been realized. International challenges and benchmarking data sets such as the yearly Pascal Visual Object Classes challenge, Caltech101 and later Caltech256 fostered this research by providing annotated data sets for training and testing. Nevertheless, researchers soon identified the dependence on the availability of annotated training data as a serious restriction: the expensive manual annotation process hampers the extension to large numbers of object classes (on the order of thousands or even more, taking into account that humans easily distinguish around 30,000 object categories (Biederman 1987)). More importantly, it forces methods to extrapolate from a relatively small number of training examples. Better performance, and especially better recall, is to be expected if more data were available for training. In response to this, weakly supervised as well as fully unsupervised methods have been investigated. In this paper, we want to experimentally validate the feasibility of fully unsupervised methods for object recognition, also referred to as *object discovery*, where only unlabeled data are provided.

Unsupervised learning can be formulated as a search for patterns in the data above and beyond what would be considered to be pure unstructured noise. Such patterns could be anything, and need not have a semantic interpretation. This makes it hard to evaluate unsupervised methods because multiple solutions (interpretations) can exist and all be equally valid as far as the data itself is concerned. Here, we build on the approach proposed in Sivic et al. (2005): a data set is constructed composed of images from a fixed

Published online: 25 July 2009

number of predefined categories. The goal is then to separate the different categories. This approach provides ground truth information against which the results obtained by different methods can be evaluated quantitatively.

In contrast to the supervised case, where systematic comparison on benchmarking data sets and challenges is standard practice, to date no in-depth evaluation and comparison of different unsupervised methods has been performed. Instead, each paper reports results on its own data set, generated ad hoc by throwing together a selection of object category images from one of the many annotated data sets available. Moreover, more often than not, evaluation is limited to one or two such data sets. This may result in over-optimistic results due to parameter fine-tuning.

Various evaluation metrics have been proposed, with *classification accuracy* probably the most popular one (Sivic et al. 2005). In this case, each discovered category is linked (often manually) to a ground truth category. Images are then assigned to the most probable category and evaluation is performed as for a standard classification problem (with accuracy defined as the sum of the diagonal elements of the confusion matrix). However, as the number of object categories and/or the imbalance in the data set increases, categories can be merged or split. We argue *conditional entropy* provides a more intuitive and scalable measure for a proper evaluation. We also propose a new evaluation scheme for the more realistic setting with multiple object categories per image, without imposing pixel-wise classification.

In this paper, a range of different methods are compared, including baseline methods such as random assignment, k-means clustering and principal component analysis, and more advanced methods such as various latent variable models and spectral clustering schemes. All of these start from the same underlying image representation: a simple bag-of-visual-words model that describes the image in terms of a set of quantized local image patch descriptors. We experiment with several local feature detectors, various vocabulary sizes, as well as different normalization schemes. More complex image representations that include spatial configuration information could yield further improvements, but fall outside the scope of this paper. Also global image representations are not considered, as these would probably be more sensitive to changing backgrounds, cannot deal with multiple objects per image, and would make the study too extensive.

We tried to avoid any kind of manual parameter tuning or model selection for a specific data set, as this could be considered a violation of the unsupervised character of our methods. Instead, we selected reasonable parameters in advance and held them fixed for all of the experiments. We report results on more than ten different test sets, each containing 20 or more different object categories, always using the same parameters.

The only parameter that we assume to be known in advance is the number of object categories in the data set. Ultimately, the machine should be able to decide on this too without any human intervention, but here we have preferred to avoid the complex issue of model selection and focus on comparing the different models given a known number of classes.

1.1 Related Work

Probably the first work to tackle the problem of unsupervised object category discovery has been Weber et al. (2000), building on a simplified constellation model-like scheme.

Especially probabilistic models seem well suited for tackling the unsupervised object discovery task and have been studied by several authors. Sivic et al. (2005) have proposed a method that builds on Probabilistic Latent Semantic Analysis (PLSA), as proposed by Hofmann (1999), to separate images of four distinct object categories (faces, airplanes, rear cars, and motorbikes). They later extended their work, experimenting with both PLSA as well as Latent Dirichlet Allocation (LDA) (Blei et al. 2003) and using multiple image segments as the equivalent of documents, so as to better localize the objects in the images (Russell et al. 2006). Very similar is the work of Tang and Lewis (2008), except that they use non-negative matrix factorization (Lee and Seung 1999). Liu and Chen (2007) extend the PLSA model with the integration of a temporal model so as to discover objects in video.

Grauman and Darrell (2006) propose an alternative method based on spectral clustering. They pay special attention to separate the objects from the background or other objects present in a single image, and also propose a semi-supervised extension.

Finally, Kim et al. (2008) build on link analysis techniques developed in the context of graph-mining and typically used in web search engines.

Recently, unsupervised object discovery methods including spatial information (Wang and Grimson 2008; Todorovic and Ahuja 2006) and hierarchical organization of categories (Sivic et al. 2008; Bart et al. 2008) have been proposed as well, but these fall outside the scope of this paper.

The rest of this paper is organized as follows. First, we introduce our evaluation metrics, both for the case of a single object per image as for the case of multiple objects per image (Sect. 2). Then we give a concise description of the image representation we use throughout this paper (see Sect. 3). Next, in Sect. 4, we give an overview of different methods for unsupervised object discovery, starting with the baseline methods (Sect. 4.1), followed by latent variable methods (Sect. 4.2) and spectral clustering (Sect. 4.3). In

Sects. 5 and 6, all these methods are applied both to subsets of Caltech256 and MSRC2 data sets and compared both quantitatively as well as qualitatively. Section 7 concludes the paper.

2 Evaluation Metrics

Recent reviews of the literature on measures for clustering can be found in Meila (2007), Rosenberg and Hirschberg (2007). Standard measures for scoring clustering quality against a known standard include *purity*, defined as the mean of the maximum class probabilities for the ground truth category labels X and obtained cluster labels Y. Given variables (x, y) sampled from the finite discrete joint space $X \times Y$, this is defined as

$$\text{Purity}(X|Y) = \sum_{y \in Y} p(y) \max_{x \in X} p(x|y).$$

In practice the distribution $p(x, y)$ is unknown so it is estimated from the observed frequencies in a test sample, resulting in an *empirical purity* estimate.

A second measure is the *mutual information or entropy gain*

$$I(X|Y) = H(X) - H(X|Y), \tag{1}$$

where

$$H(X|Y) = \sum_{y \in Y} p(y) \sum_{x \in X} p(x|y) \log \frac{1}{p(x|y)}. \tag{2}$$

Likewise, observed frequencies are used instead of probabilities, so one has an *empirical entropy gain* estimate. Our experiments have shown that purity and entropy gain are highly correlated in their ranking of different clusterings Y.

Note we usually estimate these quantities from a test set, say T, so when we do, we denote them as $\text{Purity}^T(X|Y)$ and $I^T(X|Y)$.

2.1 Conditional Entropy

The use of entropy gain was originally introduced to allow the measure to be compared *across different* domains. In a single domain, the base entropy $H(X)$ is constant, so one can also use the conditional entropy, $H(X|Y)$, as a simpler measure to evaluate different algorithms for generating clusters or components. This has a nice and intuitive interpretation, as it gives the average amount of uncertainty that remains about X once the value of Y is known. Here we use it to measure how much uncertainty remains in the true class given the instances estimated topic or cluster label. Conditional entropy has the following properties:

$$0 \leq H(X|Y) \quad \text{with equality iff } Y \text{ determines } X, \tag{3}$$

$$H(X|Y) \leq H(X) \quad \text{with equality iff } Y \text{ is independent of } X. \tag{4}$$

Roughly, assuming base 2 logarithms, one can interpret $H(X|Y)$ as saying that on average there remain about $2^{H(X|Y)}$ choices for X once Y is known. Thus with 20 classes in ground truth, $H(X|Y) = 1$ leaves about $2/20$ choices and $H(X|Y) = 2$ leaves about $4/20$ choices.

2.2 Using More Clusters

In experiments, one can soon build more clusters or components than is known in the ground truth, that is so $|X| < |Y|$. How can these subsequently be evaluated against the ground truth? The problem with the evaluation metrics described so far is that as $|Y|$ increases, purity and conditional entropy get better and better. If $|Y|$ were allowed to go arbitrarily large, purity would go to 1 and conditional entropy would go to 0. But this is due to over-fitting rather than having a good clustering.

A simple approach is to use an *oracle* to assign each discovered component to its best known class, and then evaluate the resultant assignments as before, using purity, conditional entropy or entropy gain. In practice, the oracle is created from a data set that we will call the *tuning set*. Care must be taken to ensure an unbiased evaluation and therefore, the tuning set of the oracle must be separate from the test set used to estimate the probabilities $p(x, y)$. We use tuning set T_1 for the oracle and test set T_2 to estimate probabilities. Probabilities estimated from either set are denoted $\hat{p}_{T_1}()$ and $\hat{p}_{T_2}()$ respectively.

We use a mapping $\sigma : Y \to X$ that represents the assignment of clusters to labels used in the ground truth. Each data point $(x, y) \in X \times Y$ is now mapped to a value $(x, \sigma(y))$, a co-domain we denote as $X \times \sigma(Y)$. Thus any joint distribution $p(x, y)$ on $X \times Y$ infers a joint distribution $p(x, x')$ on $X \times \sigma(Y)$. We then take measures such as purity or entropy on this inferred distribution w.r.t. the test set. Note that $\sigma()$ could be referred to as a prediction function, and $x' = \sigma(y)$ as the prediction.

Given the tuning set T_1, estimate the best mapping, i.e., the oracle, by maximizing the purity/entropy as follows:

$$\sigma_{\text{purity}} = \text{argmax}_\sigma \text{Purity}^{T_1}(X|\sigma(Y)), \tag{5}$$

$$\sigma_{\text{entropy}} = \text{argmax}_\sigma \text{H}^{T_1}(X|\sigma(Y)). \tag{6}$$

Then use the test set T_2 to get an unbiased estimate of the measures:

$$\text{Purity}(X|Y) \approx \text{Purity}^{T_2}(X|\sigma_{\text{purity}}(Y)), \tag{7}$$

$$H(X|Y) \approx H^{T_2}(X|\sigma_{\text{entropy}}(Y)). \tag{8}$$

In the case where $T_1 = T_2$ and $|Y| = |X|$, these measures are equivalent to observed estimates of the previous versions, and the effect of bias minimal. By keeping the tuning and test sets different, one can see that as $|Y|$ gets arbitrarily large, the purity and conditional entropy measures no longer tend towards their ideal values (one and zero respectively). This is the effect of using unbiased estimates.

Note, with the above formulation, one can easily do multiple train-test splits, and even perform N-way cross-validation to get, arguably, better estimates.

Also, note that for purity, the estimates can be re-phrased as sample-based sums:

$$\sigma_{\text{purity}} = \text{argmax}_\sigma \sum_{(x,y)\in T_1} 1_{x=\sigma(y)}, \tag{9}$$

$$\text{Purity}(X|Y) \approx \frac{1}{|T_2|} \sum_{(x,y)\in T_2} 1_{x=\sigma_{\text{purity}}(y)}. \tag{10}$$

This form may be more intuitive to some people, but the formulation as in (5) and (7) brings out the correspondence with entropy.

2.3 Multi-Class Ground Truth

It is more realistic that each test image contains different classes of content. For instance, one image of a city scene might have labels including buildings, cars and trees. The MSRC2 image set, used in Sect. 6, is such a collection.

Now we could ascribe proportions of each image to the ground truth. Thus one image of a city scene might have proportions including 27% buildings, 10% cars, 22% trees and 41% other (or unknown) class. However, this is not generally considered a good measure of content, since background classes like sky or grass often cover larger parts of the image than the actual foreground objects. Because we do not know of any established means of assessing proportions in a meaningful manner to classes in an image, we only consider the existence or non-existence of each class. So the ground truth for an image is the subset of classes existing in the image.

In this case the purity measure is clearly useless since it assumes each image belongs to one class. One can instead replace purity by *accuracy*, or go to the more general measures used in such cases where multiple classes are being assessed, *precision* and *recall*. In our case, precision shall be taken to mean the proportion of predictions we make that are accurate, and recall shall be the proportion of true classes that we predict. In evaluation, we use their harmonic mean called $F1$ given by $\frac{\text{precision}*\text{recall}}{\text{precision}+\text{recall}}$.

For a given image, let S denote the subset of X giving classes that occur in the image (and thus are positive). Likewise, let S' denote the subset of $\sigma(Y)$ giving classes that have been predicted to occur in the image. Then for one image and a particular prediction function $\sigma()$, the precision is $\frac{|S\cap S'|}{|S'|}$ and the recall is $\frac{|S\cap S'|}{|S|}$. We wish to accumulate those over the full test set, so quantities we estimate empirically are instead gathered using a mean. Thus

$$\begin{aligned} \text{precision}_\sigma &= \frac{\text{E}(|S\cap S'|)}{\text{E}(|S'|)}, \\ \text{recall}_\sigma &= \frac{\text{E}(|S\cap S'|)}{\text{E}(|S|)}, \end{aligned} \tag{11}$$

where $\text{E}()$ is the expected value function approximated as $\text{E}^T()$ by taking the mean of the quantity over a data set T.

Samples now take the form of a subset of classes (denoting the true classes for an image) together with either the best component or the probability scores for the components. Thus a sample is (S, y) where $S \subseteq X$ and $y \in Y$, or, in case of methods outputting a vector of component scores, $(S, \mathbf{q})$ where $S \subseteq X$ and $\mathbf{q} \in \Re^Y$. The prediction function σ on Y used previously maps to the space $\sigma(Y) = X \cup \{void\}$ which now includes a null class. We need the null class since some components may not correspond to true classes but some "other". Thus each component either maps to a known class, or it can map to nothing at all. We arbitrarily extend $\sigma()$ to the score case $\mathbf{q} \in \Re^Y$ by defining

$$\sigma(\mathbf{q}) = \left\{ x : \sum_{y:\sigma(y)=x} q_y > 0.01 \right\}. \tag{12}$$

In this case, the predicted classes S' is given by $\sigma(\mathbf{q})$.

The $F1$ score for the prediction function $\sigma()$ estimated on the data set T is the harmonic mean of precision and recall given earlier, which simplifies to

$$F1^T_\sigma(X|Y) = \frac{\text{E}^T(|S\cap S'|)}{\text{E}^T(|S| + |S'|)}. \tag{13}$$

An unbiased estimate for $F1$ using the same formulation as for entropy is then

$$\sigma_{\text{F1}} = \text{argmax}_\sigma F1^{T_1}_\sigma(X|Y), \tag{14}$$

$$F1(X|Y) \approx F1^{T_2}_{\sigma_{\text{F1}}}(X|Y). \tag{15}$$

Notice an unbiased estimate for accuracy also follows the model of purity given earlier:

$$\sigma_{\text{accuracy}} = \text{argmax}_\sigma \sum_{(S,y)\in T_1} \frac{\text{hamming}(S, \sigma(y))}{|X|}, \tag{16}$$

$$\begin{aligned} &\text{Accuracy}(X|Y) \\ &\approx \frac{1}{|T_2|} \sum_{(S,y)\in T_2} \frac{\text{hamming}(S, \sigma_{\text{accuracy}}(y))}{|X|}. \end{aligned} \tag{17}$$

Here, hamming() denotes the Hamming distance.

3 Image Representation

Images are represented using local features. To extract these we experiment with two scale-invariant feature detectors, Hessian-Laplace and Harris-Laplace (Mikolajczyk and Schmid 2004), and dense sampling of image patches using a regular grid over multiple scales. The dense sampling extracts circular patches with diameters of 12, 18, 27, 40, 60 and 90 pixels and centers spaced by respectively 6, 9, 14, 20, 30 and 45 pixels, resulting in about 6000 features for a 640×480 image. The number of features extracted by Hessian-Laplace or Harris-Laplace varies according to the image, but is usually much lower.

In each case, the patch content is described using SIFT (Lowe 2004) and vector-quantized using k-means, resulting in a fixed-size visual vocabulary. We experimented with vocabularies of 1000, 5000, and 20,000 words (using the Approximated K-Means algorithm proposed by Philbin et al. 2007 for the largest vocabularies). A single vocabulary is learned for each feature type, on a subset of the images. When combining multiple feature types, we still compute the vocabularies separately and concatenate the resulting bag-of-visual-words. This brings an automatic balancing between the different feature types. The SIFT features are not made rotation invariant because (for upright images) feature orientation typically yields valuable class information. For the dense patches, homogeneous patches are identified prior to the normalization performed by SIFT. To this end, we test if all elements are below a fixed threshold, which we set empirically. If so, we set the SIFT vector equal to zero. If not, we perform the normalization. The effect is that all homogeneous patches end up in a single visual word. Additionally, for the MSRC2 data set, we also experimented with a color-based descriptor. To this end, a patch is subdivided into 2×2 sub-patches. Each of these is described with a color histogram with 32 bins. This results in a 128 dimensional feature vector, as was the case for SIFT. Instead of using a regular discretization in color space, we compute the bins in a data-driven way, by vector-quantizing the HSV color space with k-means. This results in more realistic colors for the bin centers and a more uniform distribution over the bins.

The resulting features are collected to form a bag-of-words image representation. Spatial information such as the image positions of features, their scale or the relative positions of feature pairs, is discarded.

4 Methods for Unsupervised Object Discovery

4.1 Baseline Methods

In order to provide a yardstick for the more specialized techniques, we have tested several baseline methods:

- *Random assignment (RAN)*. Each image is assigned to a category randomly with uniform probability. Random assignment provides a sanity check in that other methods should always perform better.
- *k-means on bag of words (BOW)*. Starting from the bag-of-words image representation, 20 runs of the k-means algorithm are performed and the labeling of the run with the lowest reconstruction error is reported.
- *k-means on L1 normalized bow (L^1-BOW)*. As in the previous method, 20 runs of k-means are performed on a bag-of-words representation, but in this case the histograms are normalized by their L1 norm before running k-means. This gives them the properties of a probability distribution over visual words and compensates for the variable number of features found in different images.
- *k-means on L2 normalized bow (L^2-BOW)*. As before, 20 runs of k-means are performed on a bag-of-words representation, but in this case the histograms are normalized by their L2 norm before running k-means. This gives all bag-of-word vectors unit length and compensates for the variable number of features found in different images.
- *k-means on binarized bow (bin-BOW)*. Again, 20 runs of k-means are performed on a bag-of-word representation, but in this case the histograms are binarized before running k-means. The threshold is set for each dimension separately as the mean of that feature dimension. Such binarization process is expected to bring additional robustness.
- *k-means on tf-idf weighted bow (tfidf-BOW)*. As above, 20 runs of k-means are performed on a bag-of-words representation, but the histogram entries are weighted by the product of term frequency and inverse document frequency, where these are defined as follows. Term frequency is the number of times a given visual word appears in the image, normalized by the total number of features in the image. Inverse document frequency is a measure of the general importance of the visual word and obtained by dividing the number of all images in the database by the number of images containing the visual word, and then taking the logarithm of that quotient.
- *k-means on PCA of BOW (PCA)*. We repeated the experiments above, but now the dimensionality of the (possibly normalized) bag-of-words representation is reduced to 20 using principal component analysis (PCA), thereby reducing the influence of noise in the data. Afterwards 20 runs of k-means are performed as above.

We have chosen k-means for the vector quantization process, as this is the default clustering algorithm to build visual vocabularies used in the literature. Of course, also other clustering methods could be used. We experimented with a Gaussian mixture model, and found this to give comparable results in most cases, but also to produce degenerate solutions from time to time. Due to lack of space, we only report on the results obtained with k-means.

4.2 Latent Variable Methods

Good introductions to latent variable models from a number of different viewpoints can be found in Blei et al. (2003), Canny (2004), Buntine and Jakulin (2006). Latent Dirichlet Allocation (Blei et al. 2003) is a Bayesian version of the earlier Probabilistic Latent Semantic Analysis of Hofmann (1999), and (except when the priors are being optimized as well) both are versions of the general Gamma-Poisson model of Canny (2004). Non-negative matrix factorization (Lee and Seung 1999) is a simplified minimum entropy or maximum likelihood version as well. The relationships among these models are discussed in Buntine and Jakulin (2006), which characterizes them as methods for *discrete independent component analysis*. The differences are largely at the level of optimization algorithms, the use or lack of priors or weighted likelihoods, and the mathematical language used (likelihood, entropy, posterior, cost function, etc.).

We now present the general model. We will use the vocabulary of visual words, bags and images but all of the models generalize to other kinds of data including notably text. In statistical terminology a visual word is an observed variable and an image (instance) is characterized by its set of observed variables. Image data is supplied in the form of *counts for visual words*. Let w_j denote the number of words of type j that appear in the image, and $L = \sum_j w_j$ denote the total number of words in the image. The image is therefore represented as a bag of (visual) words as a J-dimensional data vector $\mathbf{w}$, where J is the size of the visual vocabulary.

For each image there is also a K-dimensional vector $\mathbf{l}$ of latent (hidden or unobserved) *topic scores*, where K is the number of different topics in the model. In the literature, topics are sometimes also called components, factors, aspects or clusters.

The main parameter matrix of the model is the $J \times K$ topic loading matrix Θ, with entries $\theta_{j,k}$, which give the propensities of each topic for each visual word. The column for each topic k is normalized across features, $\sum_j \theta_{j,k} = 1$, so it represents the frequency with which the various words/features occur in topic k.

When using Bayesian or full probability modeling, a prior is needed for Θ. A Dirichlet prior can be used for each k-th topic of Θ with J prior parameters γ_j, so $\boldsymbol{\theta}_{\cdot,k} \sim \text{Dirichlet}_J(\boldsymbol{\gamma})$. In practice we use a Jeffreys' prior, which corresponds to $\gamma_j = 0.5$. The use of a Dirichlet has no strong justification other than conjugacy (i.e. analytical tractability), but the Jeffreys' prior has some minimax properties (Clarke and Barron 1994) that make it more robust.

4.2.1 The Conditional Gamma-Poisson Model (NMF)

The general Gamma-Poisson model, introduced in Canny (2004), is now considered in more detail. We present a variant of it, which corresponds to a modified form of non-negative matrix factorization (NMF).

The latent topic scores vector $\mathbf{l}$ has entries l_k which are independent and Gamma distributed

$$l_k \sim \text{Gamma}(\alpha_k, \beta_k) \quad \text{for } k = 1, \ldots, K.$$

The parameters (α_k, β_k) can be collected into K-dimensional model parameter vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. The observed data $\mathbf{w}$ is assumed to be Poisson distributed, for each j

$$w_j \sim \text{Poisson}\left((\Theta\mathbf{l})_j\right).$$

The full likelihood for each image, i.e. $p(\mathbf{w}, \mathbf{l} \mid \boldsymbol{\beta}, \boldsymbol{\alpha}, \Theta, K, \text{Gamma-Poisson})$, is composed of two parts. The first comes from the K independent Gamma distributions for the l_k, and the second from the J independent Poisson distributions for the w_j with parameters $\sum_k l_k \theta_{j,k}$:

$$\prod_k \frac{\beta_k^{\alpha_k} l_k^{\alpha_k - 1} \exp\{-\beta_k l_k\}}{\Gamma(\alpha_k)} \times \prod_j \frac{(\sum_k l_k \theta_{j,k})^{w_j} \exp\{-(\sum_k l_k \theta_{j,k})\}}{w_j!}. \tag{18}$$

In practice, when fitting the parameters $\boldsymbol{\alpha}$ in the Gamma-Poisson or Dirichlet-Multinomial models, it is often the case that the α_k become very small, so for example 90% of the topic scores l_k might turn out to be less than 10^{-8} once normalized. Rather than maintaining these negligible values, we can allow the l_k to become zero with some finite probability. Allow the l_k to be independently zero with probability ρ_k and otherwise gamma distributed with probability $1 - \rho_k$. The full likelihood is now $p(\mathbf{w}, \mathbf{l} \mid \boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\rho}, \Theta, K, \text{NMF})$, and modifying the above (18), the full likelihood for each image replaces the term inside $\prod_k$ with:

$$(1 - \rho_k) \frac{\beta_k^{\alpha_k} l_k^{\alpha_k - 1} \exp\{-\beta_k l_k\}}{\Gamma(\alpha_k)} + \rho_k 1_{l_k = 0}. \tag{19}$$

4.2.2 The Dirichlet-Multinomial Model (LDA)

The Dirichlet-multinomial form was introduced as MPCA (Multinomial PCA) in Buntine (2002), but is also equivalent to LDA (Blei et al. 2003). In this case the latent variables $\mathbf{l}$ are kept normalized, so denote them differently as $\mathbf{m}$.

$$\mathbf{m} \sim \text{Dirichlet}_K(\boldsymbol{\alpha}),$$

$$\mathbf{w} \sim \text{Multinomial}(L, \Theta\mathbf{m}).$$

The arguments of the multinomial are the (known) total word count and the vector of probabilities. The full likeli-

hood is now $p(\mathbf{w}, \mathbf{m} \mid L, \boldsymbol{\alpha}, \Theta, K, \mathrm{DM})$, and the likelihood for each image is

$$C^L_{w_1,\ldots,w_J} \Gamma\Big(\sum_k \alpha_k\Big) \prod_k \frac{m_k^{\alpha_k - 1}}{\Gamma(\alpha_k)} \prod_j \Big(\sum_k m_k \theta_{j,k}\Big)^{w_j} \tag{20}$$

where $C^L_{\mathbf{w}}$ is L choose $w_1, \ldots, w_J$. The DM model can also be derived from the above Gamma-Poisson model if $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are held constant (Buntine and Jakulin 2006), but in practice one typically tries to fit $\boldsymbol{\alpha}$ in LDA so the strict equivalence does not hold.

4.2.3 Algorithm

The algorithm used here is the default high-performance algorithm supplied in the DCA code,[1] where:

- The Griffiths-Steyvers algorithm is used for model fitting. This is a specialized Gibbs sampler that is considerably more efficient because it only runs on the word-to-topic assignments (the $\mathbf{k}$ for each image), the other unknowns having been marginalized out.
- Only 300 cycles of Griffiths-Steyvers are used for convergence, with a further 100 cycles of a regular Gibbs sampler being run to record the results (the hidden variables $\mathbf{l}$ or $\mathbf{m}$ for each image, and the topic loading matrix Θ).
- The hyper-parameters of the Dirichlet or Gamma priors ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) are estimated using a bounded gradient-descent algorithm that wraps around the above.
- Thus, in total only one run of $300 + 100$ cycles is done for each experiment.

4.3 Spectral Clustering Methods

Spectral clustering denotes a family of techniques that rely on the eigen-decomposition of a modified similarity matrix to project the data prior to clustering. The variant most commonly referred to as Spectral Clustering first projects the data using the eigenvectors of an appropriately defined Laplacian followed by k-means clustering in the projected space (Ng et al. 2002). The projection of the data based on the Laplacian can be viewed as a variant of a well justified dimensionality-reduction technique called the Laplacian eigenmap (LEM) (Belkin and Niyogi 2003). There are similar methods based on Kernel PCA (KPCA). In fact Laplacian eigenmaps and KPCA solve very closely related learning problems (Bengio et al. 2004). As the two variants have differing behavior depending on the employed feature representation (Sect. 5.4), we have included results for both. The techniques and their relationship are discussed in the following sections.

[1] See http://nicta.com.au/people/buntinen.

4.3.1 Kernel PCA Clustering (KPCA)

KPCA performs PCA on data that are projected and centered in a Hilbert space defined by a kernel function (Schölkopf et al. 1998). In the case of a linear kernel this is equivalent to PCA, but in the case of a RBF kernel—i.e. one that can be written in the form $k(x, x') = f(d(x, x'))$ where d is a metric—the projection enhances locality in d and hence tends to decrease intracluster distances while increasing intercluster ones. The linear case (PCA) is one of our baseline methods, and in order to extend the technique to the nonlinear case we have experimented with two exponential kernels, the Gaussian kernel, which uses the standard L^2 metric,

$$k_{\mathrm{Gauss}}(x, x') = e^{-\frac{1}{2\sigma^2}\sum_{i=1}^d (x_i - x'_i)^2} \tag{21}$$

and the χ^2-kernel, which relies on the χ^2 distance:

$$k_{\chi^2}(x, x') = e^{-\frac{1}{2\sigma^2}\sum_{i=1}^d \frac{(x_i - x'_i)^2}{x_i + x'_i}}. \tag{22}$$

In both cases, the scale parameter σ^2 is set to the mean of the unscaled exponent. The Gaussian kernel with the standard L^2 metric is commonly used in spectral clustering algorithms (Belkin and Niyogi 2003; Ng et al. 2002), while the kernel using the χ^2 distance has been shown to be particularly effective for histogram data (Chapelle et al. 1999).

Our algorithm for clustering using kernel PCA is as follows:

$$K_{i,j} = k(x^{(i)}, x^{(j)}) \tag{23}$$

$$\tilde{K} = K - \frac{1}{l} ee^T K - \frac{1}{l} K ee^T + \frac{1}{l^2}(e^T K e) ee^T \tag{24}$$

$$(U, \Sigma) = \mathrm{eigs}(\tilde{K}, \mathrm{dim}) \tag{25}$$

$$\tilde{X} = K U \Sigma^{-\frac{1}{2}} \tag{26}$$

$$C = \mathrm{kmeans}(\tilde{X}, \mathrm{dim}) \tag{27}$$

where X_i is the ith row of X, e represents a vector of all ones, U is a matrix whose columns correspond to the dim largest eigenvectors of $\tilde{K}$, Σ is a diagonal matrix whose entries correspond to the dim largest eigenvalues of $\tilde{K}$, and C is a vector containing the cluster assignment of each image, $C_i \in 1, \ldots, N$. Equation (24) is a centering step. It ensures that the resulting kernel matrix $\tilde{K}$ corresponds to the dot products of the vectors in a data set that is centered at the origin of the Hilbert space implicitly defined by k (Schölkopf et al. 1998).

4.3.2 Normalized Cuts Spectral Clustering (LEM)

Normalized cuts spectral clustering has the same form as KPCA clustering, but employs an embedding based on a dif-

ferent interpretation of the similarity matrix. Given a similarity matrix K, we define the unnormalized Laplacian $L \equiv D - K$ where D is a diagonal matrix that contains the row sums of K, and the symmetric normalized Laplacian $\mathcal{L} \equiv D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. As described in Ng et al. (2002), the normalized cuts algorithm consists of the following steps

$$K_{i,j} = k(x^{(i)}, x^{(j)}) \tag{28}$$

$$\mathcal{L} = D^{-\frac{1}{2}} K D^{-\frac{1}{2}} \tag{29}$$

$$X = \text{eigs}(\mathcal{L}, \text{dim}) \tag{30}$$

$$\tilde{X}_i = \frac{X_i}{\|X_i\|} \tag{31}$$

$$C = \text{kmeans}(\tilde{X}, \text{dim}). \tag{32}$$

To see the relationship between this algorithm and the KPCA algorithm, we consider also the random walks Laplacian $\mathcal{L}_{rw} \equiv D^{-1} L$. The eigenvectors of $\mathcal{L}$ and $\mathcal{L}_{rw}$ are related in a straightforward way: λ is an eigenvalue of $\mathcal{L}_{rw}$ with eigenvector u if and only if λ is an eigenvalue of $\mathcal{L}$ with eigenvector $w = D^{\frac{1}{2}} u$ (von Luxburg 2007). The Laplacian eigenmap of $\mathcal{L}_{rw}$ is defined as the embedding of the data that solves

$$\min_{\alpha, \alpha^T D \alpha = 1} \alpha^T L \alpha = \min_{\alpha} \frac{\alpha^T L \alpha}{\alpha^T D \alpha} = \max_{\alpha} \frac{\alpha^T K \alpha}{\alpha^T D \alpha}. \tag{33}$$

If $D \approx dI$ where d is some scalar, then the eigenvectors obtained from KPCA using K will be the same as the generalized eigenvectors of $\mathcal{L}_{rw}$ as well as $\mathcal{L}$. The eigenvectors differ, however, in the case that D has a non-uniform spectrum.

4.3.3 Analysis of Spectral Clustering

A useful interpretation of the Laplacian Eigenmap is that if the data lie on a submanifold and are uniformly and densely sampled on it, the matrix employed is a discrete approximation to the Laplace-Beltrami diffusion operator on the submanifold (Belkin and Niyogi 2003). Performing k-means clustering in a linear projection of this matrix then approximates clustering based on distances within the submanifold.

Apart from the number of clusters, the only free parameter in these algorithms is the dimensionality dim of the spectral feature space, i.e. the number of eigenvectors kept in the dimensionality reduction. A good value for this can be estimated from the spectrum of the kernel matrix, which is typically rapidly decreasing. Despite the inverse square root weighting of the eigenvalues in (26), the overall influence of non-informative dimensions is still small (proportional to the square root of their eigenvalue) as K itself contains a power $+1$ weighting of dimensions by eigenvalues. This makes the KPCA clustering insensitive to overestimating the dim parameter. In contrast, normalized cuts spectral clustering is more sensitive to the right choice of dim, as the eigenvectors are not scaled by the square root of the eigenvalues. If dim is chosen too large, this will include directions that consist mainly of noise.

For our main experiments we set the number of dimensions equal to the number of clusters. See Sect. 5.4 for a discussion of the behavior of the spectral clustering algorithms for varying numbers of dimensions.

5 Experimental Evaluation on Caltech 256

5.1 Data Set

We first evaluate our methods on the Caltech 256 data set (Griffin et al. 2007) which contains 256 object categories with over 80 images each, plus one generic category for 'image clutter'. To avoid over-fitting to a particular data set we select several different subsets of 20 categories each for testing. First, we have manually selected a subset of 20 categories that we believe could be discriminated relatively well based on their visual feature distribution. These are listed in Table 1. In addition to this, 12 subsets have been selected consisting of object category numbers 1–20, 21–40, ..., 221–240 in the original numbering that comes with the data set (which is based on an alphabetical ordering of the object categories). We perform a detailed evaluation on the 'selected' subset, but also report results on the other test sets to verify whether the conclusions are sufficiently generic in nature.

5.2 Comparison of the Baseline Methods

We first compare the performance of the baseline methods described in Sect. 4.1: RAN, BOW, L^1-BOW, L^2-BOW, bin-BOW, and tfidf-BOW. We experiment with several image representations, i.e. different feature detectors, as well as combinations thereof. The results for the selected test set of Table 1 are summarized in Table 2.

First, note that randomly assigning images to clusters results in a conditional entropy score of 3.62 for this data set.

Table 1 20 object categories selected manually for easy discrimination

American flag	diamond ring	dice
fern	fire extinguisher	fireworks
French horn	ketch 101	killer whale
leopards 101	mandolin	motorbikes 101
pci card	rotary phone	roulette wheel
tombstone	Pisa tower	zebra
airplanes 101	faces easy 101	

Table 2 Comparison of different baseline methods using different image representations on the Caltech-256 selected test set of Table 1, measured as conditional entropy (lower is better)

Features	Voc. size	BOW	L^1-BOW	L^2-BOW	bin-BOW	tfidf-BOW
Harris Laplace	1000	2.86 ± 0.10	2.93 ± 0.11	2.41 ± 0.04	2.52 ± 0.02	2.88 ± 0.08
Hessian Laplace	1000	2.89 ± 0.08	2.69 ± 0.05	2.22 ± 0.04	2.52 ± 0.04	2.68 ± 0.04
HarLap + HesLap	2000	2.96 ± 0.10	2.66 ± 0.05	2.08 ± 0.03	2.43 ± 0.04	2.71 ± 0.15
Dense patches	1000	2.49 ± 0.10	2.13 ± 0.04	1.96 ± 0.04	1.78 ± 0.03	2.34 ± 0.02
HarLap + dense	2000	2.48 ± 0.10	2.25 ± 0.07	1.75 ± 0.03	1.73 ± 0.03	2.66 ± 0.13
HesLap + dense	2000	2.48 ± 0.09	2.13 ± 0.07	1.74 ± 0.04	1.72 ± 0.03	2.59 ± 0.07
HarLap + HesLap + dense	3000	2.52 ± 0.10	2.24 ± 0.07	$\mathbf{1.67} \pm 0.04$	1.77 ± 0.05	2.61 ± 0.13
Hessian Laplace	5000	3.22 ± 0.12	3.09 ± 0.14	2.18 ± 0.07	3.04 ± 0.13	3.22 ± 0.09
Hessian Laplace	20,000	3.36 ± 0.15	3.31 ± 0.09	2.27 ± 0.06	3.39 ± 0.15	3.38 ± 0.17

This value is already lower than $\log_2(20) = 4.32$, because the selected image categories are not equally sized. The best performing baseline method achieves a conditional entropy score of 1.67. In other words, by applying it, the remaining uncertainty on the true object category has been reduced from $2^{3.62} = 12.3$ out of 20 for the random assignment down to $2^{1.67} = 3.2$ out of 20 for the best combination.

Second, normalization of the bag-of-words representation is crucial when using a simple algorithm like k-means for object discovery. However, which type of normalization gives the best results depends on the type of features used. When using interest points (Hessian-Laplace or Harris-Laplace), normalization using L2 norm gives the best results, followed by a binary bag-of-words. When using dense sampling, on the other hand, the binarized version of the bag-of-words gives best results, followed by L2-normalization. However, on the other test sets, in most cases L2 normalization gives best results, also for dense patches. L1 normalization is in most cases better than no normalization, but not as good as L2 or binary. The good performance of the binary bag-of-words may be somewhat surprising, as a lot of information is thrown away. Yet, the non-linearity of the thresholding process reduces the effect of very strong peaks in the histograms, as is typically found for the bin corresponding to homogeneous patches in case of dense sampling.

Third, the results of tf-idf are somewhat disappointing and below our expectations. It does improve the performance for small vocabulary sizes. For combined features or large vocabularies, on the other hand, it does not seem to have much influence at all. This form of normalization has proven useful in text document analysis, but the underlying reasoning seems not to generalize to the object discovery setting.

Fourth, dense patches give better results than the interest point detectors. Hessian-Laplace features seem slightly better than Harris-Laplace (at least for this test set). However, the more feature types are combined, the better the results get.

Finally, we increased the size of the vocabulary for the Hessian Laplace features from 1000 to 5000 and 20,000. Although this allows one to capture more fine-grained details, none of the baseline methods seems to benefit significantly from the larger vocabularies.

PCA dimensionality reduction We also experimented with a dimensionality reduction based on Principal Component Analysis (PCA) prior to the k-means clustering. However, when combined with dense sampling or with L2 normalization (i.e., the methods performing best without PCA), this did not have any effect. When combined with interest points and L1 normalization or binary bag-of-words, a minor improvement was found, but never more than 0.3 and never beating the best combination without PCA.

Other test sets The results over all 13 Caltech256 test sets for a selected set of baseline methods are shown in Fig. 1. The overall ranking of the methods is relatively stable over the different test sets. This confirms that our conclusions drawn for the selected test set are general in nature. In most cases, the dense patches give better results than the interest point detectors, but not always. However, in all cases, the combination of all feature types and L2 normalization gives the best result.

On two of the test sets ('selected' and '141–160') all scores are systematically lower than on the remaining test sets. This holds for all methods including random selection. These are test sets containing one or more object categories with a significantly larger number of images than the others. For instance, airplanes and motorbikes have around 800 images each, while the average number of images is only 116. With unbalanced data sets, the detected clusters tend to have more peaked distributions of ground truth labels and as a result lower conditional entropy scores. It is important for

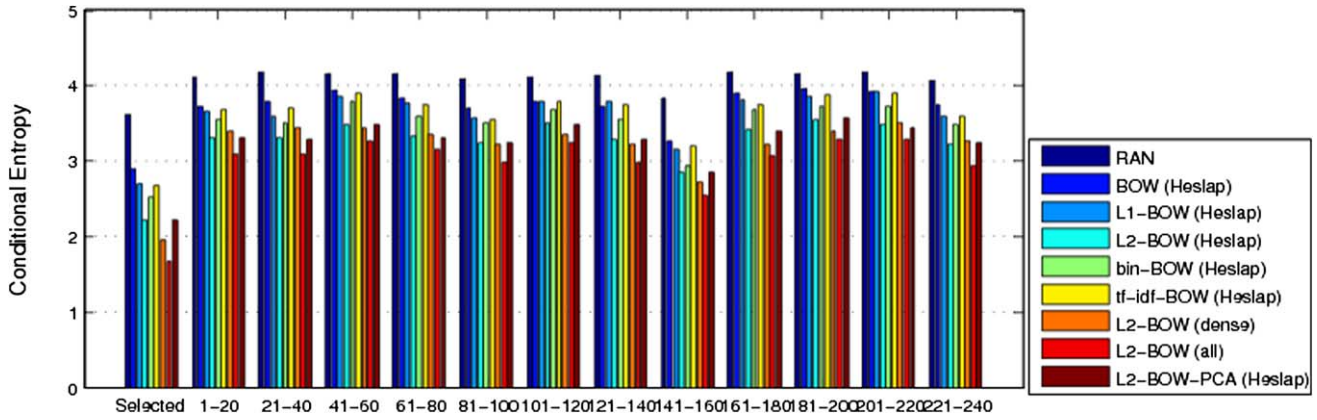


Fig. 1 (Color online) Conditional entropies for the different baseline methods on all 13 test sets of Caltech256

Table 3 Results of the latent variable methods using different image representations, for the selected Caltech256 categories of Table 1, measured in conditional entropy (lower is better)

Features	Voc. size	LDA	NMF
Harris Laplace	1000	2.63 ± 0.03	2.56 ± 0.03
Hessian Laplace	1000	2.40 ± 0.03	2.37 ± 0.05
HarLap + HesLap	2000	2.31 ± 0.05	2.28 ± 0.05
Dense patches	1000	2.17 ± 0.05	2.18 ± 0.05
HarLap + dense	2000	2.12 ± 0.05	2.14 ± 0.03
HesLap + dense	2000	2.06 ± 0.07	2.06 ± 0.07
Hessian Laplace	5000	2.15 ± 0.04	2.13 ± 0.03
Hessian Laplace	20,000	$\mathbf{1.99} \pm 0.02$	$\mathbf{2.00} \pm 0.04$

unsupervised methods to be able to handle such imbalances correctly. Therefore, we intentionally did not balance our test sets. However, for comparison, we also performed some tests on balanced test sets (using the first 80 images of each category). This gave roughly the same results and won't be reported in detail here. Note that the imbalances also prevent the comparison of absolute entropy scores between the different test sets (see also Sect. 2.1).

5.3 Comparison of the Latent Variable Models

Next, we evaluate the latent variable methods: Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF), described in Sect. 4.2. The main results are summarized in Table 3 and Fig. 3. First, we notice that the results are not as good as the results reported earlier, with the lowest conditional entropy being 1.99, which corresponds to a remaining uncertainty of $2^{1.99} = 4$ out of 20 object categories. This is worse than some of the results obtained with the much simpler baseline methods. This can be explained by the fact that the latent variable models are more general, in that they have been developed especially to deal with the case of multiple components per image. However, for the Caltech256 data set, where there is only a single object category per image, this higher level of flexibility is not needed. Hence, clustering-based methods, which exploit this special characteristic, yield better results.

Second, we observe that the latent variable methods do not benefit that much from combining different feature types, in contrast to some of the baseline methods. This can be explained by the fact that they are essentially based on probability distributions (Buntine and Jakulin 2006): when dense data is provided, it dominates the distributions, so incorporating additional sparse features has little effect. Maybe a re-weighting of the different bag-of-words before concatenating them could overcome this problem. For the same vocabulary size, dense patches still outperform the interest point detectors, and Hessian Laplace again gives better results than Harris-Laplace (at least for this test set). Larger vocabularies seem to be especially beneficial in this context. By increasing the size of the vocabulary, the conditional entropy for Hessian Laplace features can be reduced from 2.4 down to 1.99.

Third, the differences between the two latent variable models are minimal. NMF slightly outperforms LDA when interest point detectors are used, while the opposite is true when using dense sampling—but these differences are insignificant.

Finally, in contrast to what we observed earlier for the baseline methods, the latent variable models do seem to ben-

Table 4 Results of the spectral clustering based methods using different image representations, for the selected Caltech256 categories of Table 1, measured in conditional entropy (lower is better)

Features	Voc. size	L^2-KPCA-$\mathcal{G}$	L^2-KPCA-χ^2	L^2-LEM-$\mathcal{G}$	L^2-LEM-χ^2
Harris Laplace	1000	2.42 ± 0.02	2.32 ± 0.01	2.57 ± 0.02	2.54 ± 0.03
Hessian Laplace	1000	2.23 ± 0.02	2.26 ± 0.02	2.31 ± 0.01	2.25 ± 0.01
HarLap + HesLap	2000	2.06 ± 0.02	2.09 ± 0.01	2.21 ± 0.01	2.10 ± 0.02
Dense patches	1000	2.00 ± 0.01	1.81 ± 0.02	2.04 ± 0.02	1.83 ± 0.02
HarLap + dense	2000	1.79 ± 0.02	1.65 ± 0.01	1.85 ± 0.03	1.65 ± 0.05
HesLap + dense	2000	1.77 ± 0.01	1.65 ± 0.02	1.95 ± 0.03	1.62 ± 0.02
HarLap + HesLap + dense	3000	1.73 ± 0.01	1.64 ± 0.02	1.86 ± 0.01	$\mathbf{1.58} \pm 0.02$
Hessian Laplace	5000	2.22 ± 0.02	2.20 ± 0.02	2.33 ± 0.00	2.22 ± 0.02
Hessian Laplace	20,000	2.28 ± 0.03	2.35 ± 0.04	2.37 ± 0.03	2.29 ± 0.02

efit significantly from the use of larger vocabularies. By increasing the size of the vocabulary, the Hessian Laplace features outperform the results of the (small vocabulary) dense features. This can be explained by the fact that larger vocabularies make the bag-of-words descriptions sparser.

5.4 Comparison of Spectral Clustering Based Methods

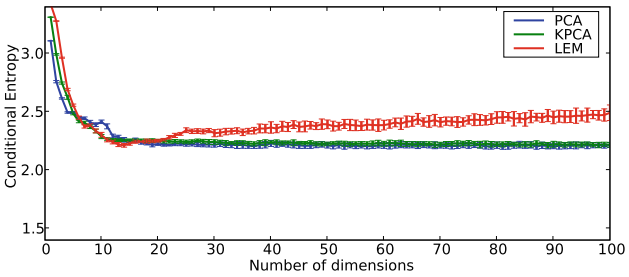
Next, we evaluate the object discovery methods based on spectral clustering: Laplacian Eigenmaps (LEM) and kernel-PCA (KPCA). For the 20 categories of Table 1, the most important results are summarized in Table 4.

For both methods (LEM and KPCA) we have experimented with two types of kernels (Gaussian and χ^2) on top of both L1 and L2 normalized bag-of-words. The χ^2 kernel gives consistently better or similar results compared to the Gaussian kernel. Using the χ^2 kernel, no significant differences in performance between the two types of normalization (L1 and L2) are found. Using the Gaussian kernel, L2 normalization seems to work slightly better than L1. Due to the power of the spectral clustering, the choice of the right normalization scheme seems to be less critical. We only include the L2 normalization results in the table.

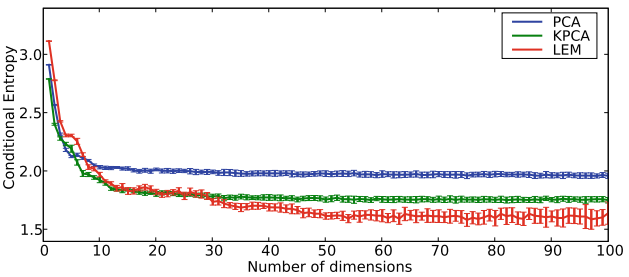
Comparing KPCA and LEM, no significant differences in performance are found. The overall best result using spectral clustering is obtained when using a combination of all feature types, L2 normalization and Laplacian eigenmaps with a χ^2 kernel, giving a conditional entropy score of 1.58. This corresponds to a remaining uncertainty on the true object category of $2^{1.58} = 3.0$ out of 20, and makes the spectral clustering methods the best performing scheme on this test set.

The spectral clustering methods work best when as many features as possible are combined. Especially including the dense patches seems to be beneficial. The larger vocabularies, however, do not bring any further improvement, but rather reduce performance.

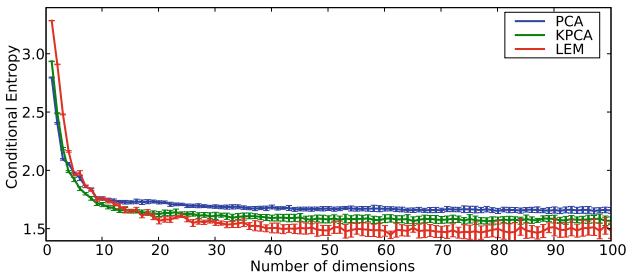
Number of dimensions For the spectral clustering methods, as for principal component analysis, there is a free parameter determining the number of dimensions onto which to project the original space. So far, we have fixed this parameter to the (known) number of object categories in the data set, i.e. 20. In Fig. 2 we explore the effect of changing this parameter. Interestingly, for the image representations based



(a) Hessian-Laplace



(b) Dense



(c) Harris-Laplace + Hessian-Laplace + Dense

Fig. 2 (Color online) Conditional entropy as a function of the dimensionality of the reduced space for PCA, LEM, and KPCA, using Hessian-Laplace (*top*), dense patches (*middle*), and all three feature types combined (*bottom*)

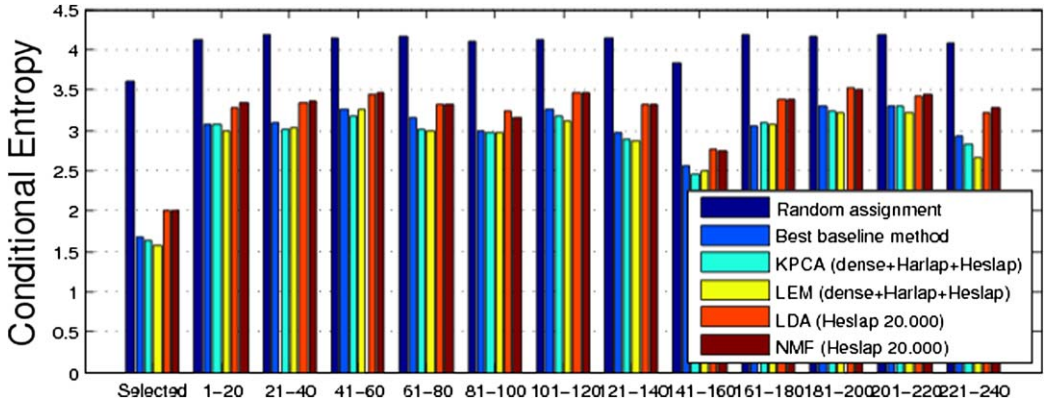


Fig. 3 (Color online) Conditional entropies for the best performing combinations on all 13 test sets of Caltech256

on interest points (Hessian Laplace and Harris Laplace), the performance of the Laplacian Eigenmaps decreases with increasing dimension. In fact, the optimum is even below 20 dimensions. However, when the dense patches are included in the image representation (by themselves, or in combination with the interest points), LEM performance keeps improving with more dimensions, though it also gets noisier. This behavior can be explained by the scaling of the eigenvectors (see Sect. 4.3). KPCA and PCA use a different scaling method and are not affected: they saturate nicely for all (combinations of) feature types (PCA earlier than KPCA).

Other test sets Figure 3 shows the results for the best performing methods on all 13 test sets of Caltech256. Again, the overall ranking of the methods is found to be relatively stable over the different data sets. The spectral clustering methods always give the best results. The latent topic models cannot compete with the spectral clustering results nor with the best baseline method (i.e., using all feature types and L2 normalization). The difference between the two different latent topic models is negligible, and the same holds for the difference between the two versions of spectral clustering.

However note that even the best models are far from perfect in these tests, with the lowest conditional entropy scores typically around 3, reducing the residual label ambiguity from around 18 out of 20 for the random assignment to around 8 out of 20.

5.5 Qualitative Evaluation

Apart from the quantitative analysis given above, it is interesting to visually evaluate the components found by the different methods. Indeed, even if they do not correspond to the 20 ground truth object classes, the components may still capture relevant patterns in the background or focus on common characteristics shared by multiple object categories.

Figures 4 and 5 illustrate this for latent variable model NMF and spectral clustering LEM, again using the 20 categories of Table 1. Each method uses the feature set that gives the best performance for it in the above experiments, that is the combination of all three feature types for spectral clustering and the 20,000 words vocabulary based on Hessian Laplace for the latent variable model. Each figure has the 20 detected components as rows, sorted in increasing order of their conditional entropy (i.e., the cleanest components are shown on top). In each row 12 images are shown, randomly sampled from those assigned to the category. Note that this is in contrast to most of the existing presentations in the literature where only the top N "most typical" images are shown. However, we believe that showing random images gives a more realistic image of the methods capabilities. To the right of each row we give the conditional entropy of the component and the number of images assigned to it.

A detailed examination of the images gives rise to several observations:

- The unsupervised methods sometimes discover a finer granularity of object classes than expected, e.g. splitting airplanes in the sky from airplanes on the ground.
- Both methods are affected by the unbalanced input data. For example motorbikes, airplanes and faces have respectively 798, 800 and 453 images. As a result, these categories have relatively more weight in the clustering. In practice the clustering almost always splits them into more than one component, which in turn forces some of the other less frequent object categories to be merged. The latent variable methods seem to be more robust in this respect than the spectral clustering.
- The images are more or less equally spread over the discovered categories, but less so for the latent variable methods than for the spectral clustering.
- For spectral clustering, the top three components have a conditional entropy of 0.00, which indicates that for each of these components, all of the images assigned to it belong to a single ground truth object category.

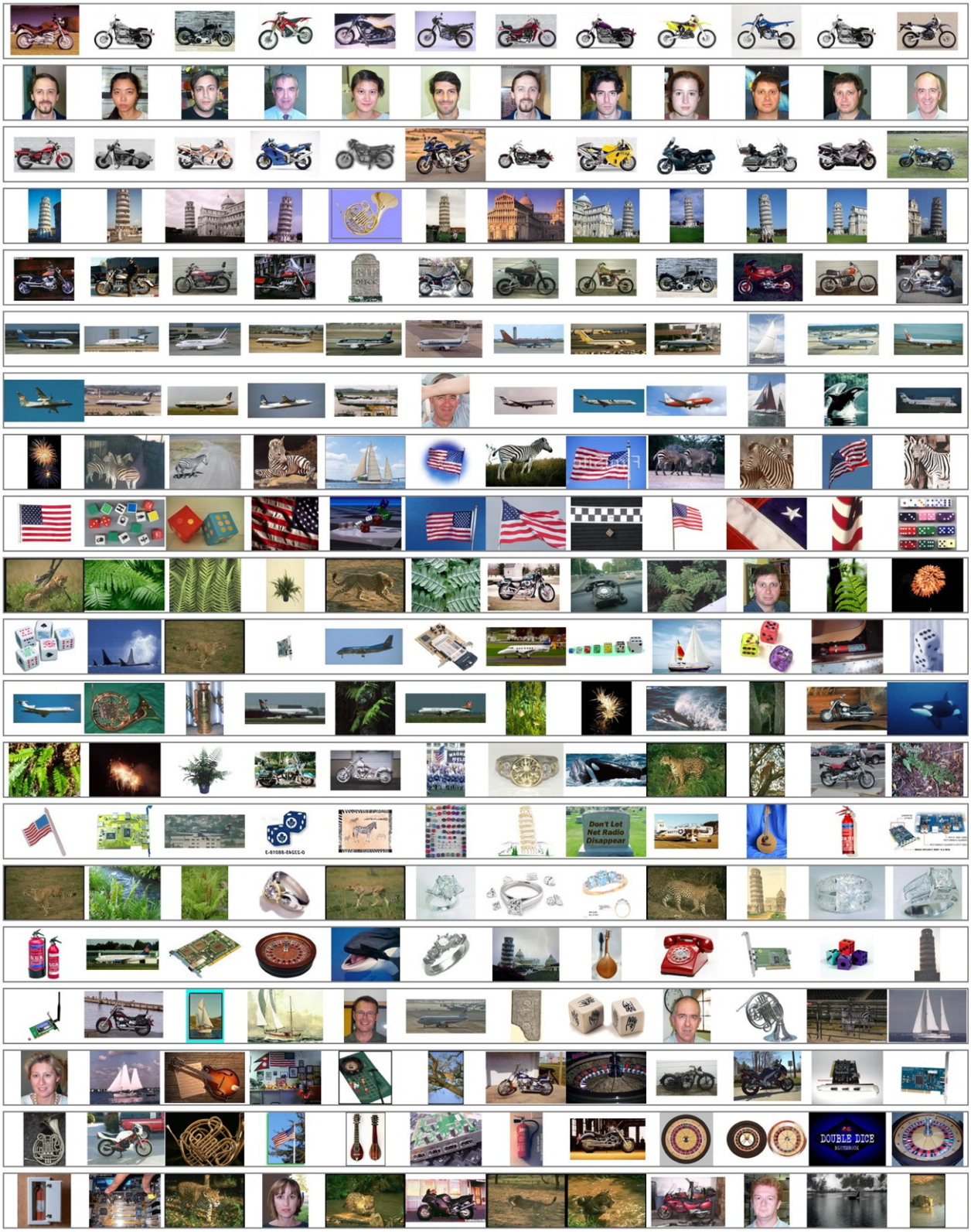


Fig. 4 (Color online) 12 prototypical images for each of the 20 components, as detected with the latent variable method NMF, using the optimal settings (a 20,000 words vocabulary based on Hessian-Laplace), for the selected test set of Table 1. For each component, we also indicate the conditional entropy (*black*) and the number of images assigned to this topic (*blue*)

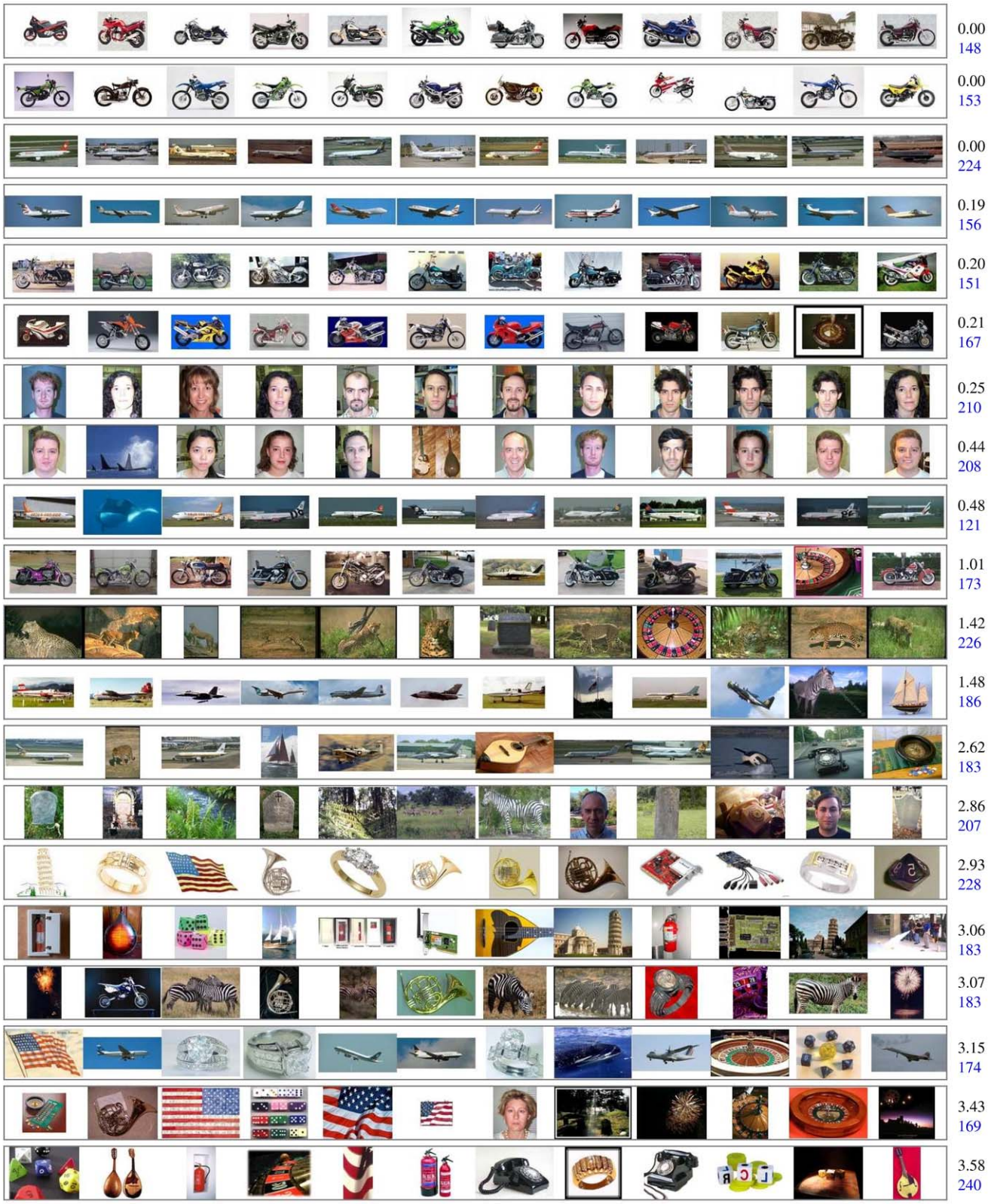


Fig. 5 (Color online) 12 prototypical images for each of the 20 components detected by spectral clustering, using the optimal settings (all feature types combined, χ^2 kernel, L2 normalization), for the selected test set of Table 1. For each component, we also indicate the conditional entropy (*black*) and the number of images assigned to this topic (*blue*)

– Conditional entropy scores above 2 indicate that there are still more than 4 choices for the true object category after being informed about the discovered category (see also Sect. 2.1). In a sense, we could say that discovered categories with a conditional entropy above this threshold are not really matching to any of the ground truth categories. With this criterion, our methods have only discovered the airplanes, motorbikes, and faces. On top of that, spectral clustering was able to discover one more class, being leopards, while NMF discovered the Pisa tower.
– For the latent variable methods, the discovered object categories sometimes focus on part of the image only. For instance, two different components are used for the American flags, where one focuses on the stripes (and as a result, also contains images of zebras) while the other focuses on the stars (including also many dice).
– The image representation is not robust to intensity inversions. As a result the dice with dark spots are in a different category as the dice with bright spots (for the latent variable method, at least).
– Some of the object categories were not isolated by any method, including rotary dial telephones, mandolins and fire extinguishers. Presumably the feature extraction fails to find a sufficiently large set of distinctive features that appear consistently on these classes and not on any others.

6 Experimental Evaluation on MSRC2 Data Set

6.1 Data Set

The Caltech 256 data set used so far is limited in that it only contains a single object category per image and the object is always centered in the image, often with a homogeneous background. This simplifies the object discovery task significantly and makes the experiment somewhat artificial. That is why we have also included experiments on the MSRC2 data set (Shotton et al. 2006).

This data sets contains 591 images of 23 different object categories, some of which are truly objects (e.g., car, book, cow) while some are more scene types (e.g., grass, road, sky). All the images have been manually segmented and often contain multiple object categories simultaneously.

6.2 Quantitative Evaluation

Instead of conditional entropy, we evaluate the performance of the different methods using the scheme proposed in Sect. 2.3: based on a tuning set which acts as an oracle, the discovered components are first assigned to the ground truth object categories (or none, if such assignment harms the quality measure). This does not introduce a bias, since the evaluation is then performed on a held-out test set. All results are computed using 4-way cross-validation. Here, we report results using both the F1-measure as well as purity.

The most important results are summarized in Table 5. Interestingly, on this data set and using the multi-class evaluation scheme, the latent variable method outperforms the clustering-based method. This can be explained by the fact that the clustering scheme cannot separate the different components in the images.

Splitting up the F1-measure in precision and recall shows a significant difference between the two methods: the spectral clustering method has a relatively high precision (≈70) but a low recall (≈30), while the latent variable method has more average precision and recall (≈45 and ≈55 respectively). Indeed, the spectral clustering groups all images with the same scene type together (e.g. cows on a field with sky above), and does so relatively accurately, but cannot generalize the concept to include also the same objects on a different background or in combination with other objects. The latent variable model does better in this respect. However, there is still a lot of room for improvement. Especially

Table 5 Comparison of the different methods based on the MSRC2 data set, measured in F1 score as well as purity (higher is better)

Using 23 components			F1-measure			Purity		
Features	Descriptor	Voc. size	L^2-BOW	NMF	L^2-LEM-χ^2	L^2-BOW	NMF	L^2-LEM-χ^2
Dense patches	SIFT	1000	0.196	0.239	0.201	0.853	**0.869**	0.853
Dense patches	color	1000	0.170	**0.253**	0.154	0.853	0.866	0.853
Hessian Laplace	color	1000	0.143	0.215	0.151	0.853	0.860	0.853
Hessian Laplace	SIFT + color	1000	0.162	0.224	0.149	0.853	0.863	0.853
Using 50 components								
Dense patches	SIFT	1000	0.193	0.243	0.204	0.853	**0.872**	0.852
Dense patches	color	1000	0.176	**0.253**	0.176	0.853	0.867	0.853
Hessian Laplace	color	1000	0.147	0.201	0.161	0.853	0.860	0.853
Hessian Laplace	SIFT + color	1000	0.162	0.225	0.152	0.854	0.864	0.853

the more complex objects are still problematic. Maybe this could be overcome by switching to a larger data set, with more examples of each object category and more variability in scene composition.

As before, using dense patches seems to give better results than using interest point detectors. Since color seems to be important in this data set (e.g., to distinguish between road, sky, and grass), we have also experimented with a color-based descriptor (see Sect. 3). The color-based results are competitive, but not significantly better than the results based on SIFT. Also the combination of the shape and color descriptor (where both feature vectors are simply concatenated prior to the vector quantization) does not bring any improvement.

More than 23 components As an additional experiment, we have also tested the effect of increasing the number of discovered components from 23 to 50. This gives the methods more flexibility and allows the discovery of more fine-grained patterns in the data. Again using an oracle to assign the components to the ground truth object categories, we obtain the results shown in the lower half of Table 5.

Increasing the number of components does not affect the purity. Also the F1-measure for the baseline and latent variable methods is mostly unaffected. However, for the spectral clustering method the F1-measure increases significantly. This is due to both an increase in precision as well as an increase in recall.

6.3 Qualitative Evaluation

Apart from a quantitative analysis, it is also interesting to look at the results from a more qualitative point of view. To this end, we again show a few random images for each component, for the two best performing combinations (dense patches and SIFT for LEM and Hessian Laplace with a vocabulary of 20,000 visual words for NMF)—see Figs. 6 and 7. Again, we add the number of images for which this component got the highest score. However, these numbers need to be interpreted with care, as now we are working in a multi-class setting and images are actually assigned to multiple components. In red, we also indicate the ground truth object category each component was assigned to by the oracle.

At first sight, the spectral clustering results look cleaner. However, as mentioned earlier, this method has actually discovered the different scene types rather than the different object categories out of which the scenes are composed.

By optimizing the F1-measure, the oracle searches for a compromise between precision and recall. Thus, components are mostly labeled with more frequent categories like sky or grass rather than airplane or cow.

Note the need for LEM to find a single component assignment per image means it is clustering the images, whereas NMF tends to have a lot more "void" assignments which correspond to "other" aspects of the images, a significant factor by proportion of features.

Finally, the random selection of images for each component used for visualization are, unfortunately, not comparable across methods. For LEM, there is only one topic for each image. For NMF, there are potentially 3–4 topics for each image, so the chance of selecting a poorer one is much greater.

7 Conclusion and Future Work

In this work, we evaluated the performance of different unsupervised methods for object discovery, based on a simple bag-of-visual-words image representation. Setting up such an evaluation framework is an important contribution of this paper.[2] In this framework, we developed unbiased evaluation methods for unsupervised learning on images with multiple ground-truth classes assigned.

In case there is only one object category per image, clustering-based methods give the best results, significantly outperforming the latent variable models. Even the baseline methods using k-means clustering already yield state-of-the-art results, only outperformed by spectral clustering. To maximize the performance, it is important to select the right image representation (interest points, dense patches, or both) as well as to normalize the bag-of-words histograms correctly. Moreover, these design choices are different for latent variable models than for the (spectral) clustering based methods.

In case there are multiple object categories per image, both the object discovery task as well as an unbiased evaluation protocol become more challenging. The latent variable methods hold good promise in this setting, as they model images as mixtures of components, however, so too would multi-class versions of spectral clustering if they could be developed. Based on our preliminary experiments, latent variable methods do indeed outperform the spectral clustering based approaches. Though, none of the investigated methods really succeeds in separating the different object categories (although they do discover some structure in the image data set). This may in part be alleviated by increasing the size of the data set. Nevertheless, our main conclusion is that fully unsupervised methods for object discovery, under realistic circumstances (i.e. with multiple objects per image and more than just a couple of object categories) is still a largely unsolved problem.

Bringing in some weak form of spatial information is one interesting avenue of future research that might bring

[2]The image representations, some of the implemented methods, and the evaluation scripts are available at http://homes.esat.kuleuven.be/~tuytelaa/unsupervised.html.

Fig. 6 (Color online) 12 prototypical images for each of the 20 topics, as detected with the latent variable method NMF, using the optimal settings (dense patches) for the MSRC2 data set. For each topic, we also indicate the number of images assigned to this topic (*blue*) as well as the class label assigned to the topic by the oracle (*red*)

Fig. 7 (Color online) 12 prototypical images for each of the 20 topics detected by spectral clustering, using the optimal settings (all feature types combined, using both sift and color descriptors), for the MSRC2 data set. For each topic, we also indicate the number of images assigned to this topic (*blue*) as well as the class label assigned to the topic by the oracle (*red*)

us closer to our goal. There are several ways this can be done: providing approximate image segmentations, providing seed classes using segmented images (for instance, pairing some image patches), building components or clusters on segmented images, and so forth.

Acknowledgements The authors acknowledge support from the EU projects CLASS (IST project 027978), PerAct (IST project 504321) and the EU Network of Excellence PASCAL2. Tinne Tuytelaars was supported by the Fund for Scientific Research Flanders.

Wray Buntine's involvement in CLASS is funded through the University of Helsinki. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

Bart, E., Porteous, I., & Perona, P. (2008). Unsupervised learning of visual taxonomies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*(6), 1373–1396.

Bengio, Y., Delalleau, O., Le Roux, N., Paiement, J.-F., Vincent, P., & Ouimet, M. (2004). Learning eigenfunctions links spectral embedding and kernel pca. *Neural Computation*, *16*(10), 2197–2219.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, *94*(2), 115–147.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Buntine, W. L. (2002). Variational extensions to EM and multinomial PCA. In *13th European conference on machine learning (ECML'02)*, Helsinki, Finland.

Buntine, W. L., & Jakulin, A. (2006). Discrete components analysis. In C. Saunders, M. Grobelnik, S. Gunn & J. Shawe-Taylor (Eds.), *Subspace, latent structure and feature selection techniques*. Berlin: Springer.

Canny, J. (2004). GaP: a factor model for discrete data. In *SIGIR 2004* (pp. 122–129).

Chapelle, O., Haffner, P., & Vapnik, V. (1999). Svms for histogram-based image classification. In *IEEE transactions on neural networks, special issue on support vectors*.

Clarke, B. S., & Barron, A. R. (1994). Jeffrey's prior is asymptotically least favorable under entropy risk. *Journal of Statistical Planning and Inference*, *41*, 37–60.

Grauman, K., & Darrell, T. (2006). Unsupervised learning of categories from sets of partially matching image features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset*. Technical Report 7694, California Institute of Technology.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Research and development in information retrieval* (pp. 50–57).

Kim, G., Faloutsos, C., & Hebert, M. (2008). Unsupervised modeling of object categories using link analysis techniques. In *IEEE conference on computer vision and pattern recognition*.

Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

Liu, D., & Chen, T. (2007). A topic-motion model for unsupervised video object discovery. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoint. *International Journal of Computer Vision*, *2*(60), 91–110.

Meila, M. (2007). Comparing clusterings: an information based distance. *Journal of Multivariate Analysis*, *98*, 873–895.

Mikolajczyk, K., & Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, *1*(60), 63–86.

Ng, A., Jordan, M., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems, Vol. 14*. Cambridge: MIT Press.

Philbin, J., Chum, O., Isard, M., Sivic, J., & Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, EMNLP-CoNLL* (pp. 410–420).

Russell, B. C., Efros, A. A., Sivic, J., Freeman, W. T., & Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*(5), 1299–1319.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *Proceedings European conference on computer vision*.

Sivic, J., Russell, B. C., Efros, A., Zisserman, A., & Freeman, W. T. (2005). Discovering object categories in image collections. In *Proceedings of the international conference on computer vision*.

Sivic, J., Russell, B. C., Zisserman, A., Freeman, W. T., & Efros, A. A. (2008). Unsupervised discovery of visual object class hierarchies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Tang, J., & Lewis, P. (2008). Non-negative matrix factorisation for object class discovery and image auto-annotation. In *ACM international conference on image and video retrieval*.

Todorovic, S., & Ahuja, N. (2006). Extracting subimages of an unknown category from a set of images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

Wang, X., & Grimson, E. (2008). Spatial latent Dirichlet allocation. In *Proceedings of neural information processing systems conference*.

Weber, M., Welling, M., & Perona, P., (2000). Towards automatic discovery of object categories. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.