

THE STRUCTURAL FORM IN IMAGE CATEGORIZATION

Juha Hanni, Esa Rahtu and Janne Heikkilä

Machine Vision Group, Department of Electrical and Information Engineering, University of Oulu, Finland

Keywords: Image categorization, Clustering, Generative model.

Abstract: In this paper we show an unsupervised approach how to find the most natural organization of images. Previous methods which have been proposed to discover the underlying categories or topics of visual objects create no structure or at least the structure, usually tree-shaped, is defined in advance. This causes a problem since the most relevant structure of the data is not always known. It is worthwhile to consider a generic way to find the most suitable structure of images. For this, we apply the model of finding the structural form (among eight natural forms) to automatically discover the best organization of objects in visual domain. The model simultaneously finds the structural form and an instance of that form that best explains the data. In addition, we present a generic structural form, so called meta structure, which can result in even more natural connections between clusters of images. We show that the categorization results are competitive with the state-of-the-art methods while giving more generic insight to the connections between different categories.

1 INTRODUCTION

As more and more images and image categories become available, organizing them is crucial. By learned organization we can enable a quicker identification of an unknown object, explore the relations between the clusters of images and easily find categories similar to each other. This can help us to obtain a better classification result.

Recent applications considering previous issues are presented at least in (Sivic et al., 2008; Bart et al., 2008; Marszalek and Schmid, 2008). Relations between categories can also be useful in object recognition and detection, as shown in (Ahuja and Todorovic, 2007; Parikh and Chen, 2007). A drawback here is that all the previous methods deal with tree-shaped structures only.

Unsupervised probabilistic latent topic discovery models like probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA), earlier used in text categorization (Blei et al., 2003; Hoffmann, 2001), are straightforward to employ in case of visual data by using visual vocabulary (Sivic et al., 2005; Bosch et al., 2006). These models create a flat topic structure where each document has a probability of belonging to each topic. To extract the relations between topics, a hierarchical LDA has recently been applied to image data in (Sivic et al., 2008). They

showed it to improve the classification accuracy but this method exploits again only a tree-shaped structure to describe the relations between topics. A better way still might be to look more inside the data and find the structure that best describes it. This way we can exploit the gained organization most.

In this paper, we simultaneously categorize images and find the best structure to describe the connections between categories, all in an unsupervised manner. In other words, we propose a method which gives us a chance to learn automatically the most natural structure of images instead of using a fixed structure. The method we use is based on the algorithm introduced in (Kemp and Tenenbaum, 2008).

We improve the algorithm to consider also a so called meta structure. In theory, a meta structure can adapt to any structure that exists. In addition, we propose how to add samples to a given structure and how this algorithm can be applied to large datasets. The experiments reveal a competitive classification accuracy, and furthermore, the generated structures fit the data seemingly well.

This paper is organized as follows. Section 2 reviews shortly the algorithm of discovering the structural form and shows the improvements we made. In section 3, we show how to employ this method in visual domain. Section 4 then describes the experiments made on two set of images: MSRC-B1 dataset and a

set of faces. Finally, section 5 presents our conclusions.

2 THE STRUCTURAL FORM

In this section, we first shortly review the basics of the algorithm of discovering the structural form. This algorithm tries to find a structure (among eight natural forms) which describes the data most likely. The structures are presented by graphs and each graph is characterised by a specific graph grammar. The nodes of the graph represent categories and the edges represent similarities between the categories. A grammar defines now a generative process to create a structure. Note that each form has its own specific grammar which can be described as a node splitting rule for the generative process.

To go further on this subject, we prefer to have a more flexible structure to describe the relations between categories than any of the used forms. Thus the end of this section concentrates on how to make it possible to learn an instance of a meta structure by using a meta-grammar which consists of several node splitting rules.

2.1 Discovering the Structural Form

The approach described here is adapted from (Kemp and Tenenbaum, 2008) (matlab implementation available online) and we call it "Kemp's algorithm" or just "the algorithm".

We define form F to be any of the following forms: partition, chain, order, ring, hierarchy, tree, grid and cylinder. Structure S , generated from form F , is presented by a graph with nodes corresponding to clusters of entities. An entity graph, S_{ent} , is a graph where entities are included to cluster nodes by adding an extra node for each entity and connecting it by an edge to the cluster node which entity is assigned to. An example is presented in figure 1.

Let D be an $n \times m$ entity-feature matrix and S a structure of form F . We are now searching for a structure and form which together maximize the posterior probability

$$P(S, F|D) \propto P(D|S)P(S|F)P(F), \quad (1)$$

where $P(F)$ is a uniform distribution over all the possible forms considered.

Probability $P(S|F)$ in equation (1) is the probability of that the structure is generated from a given form. We define

$$P(S|F) \propto \begin{cases} \theta^{|S|}, & \text{if } S \text{ is compatible with } F \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

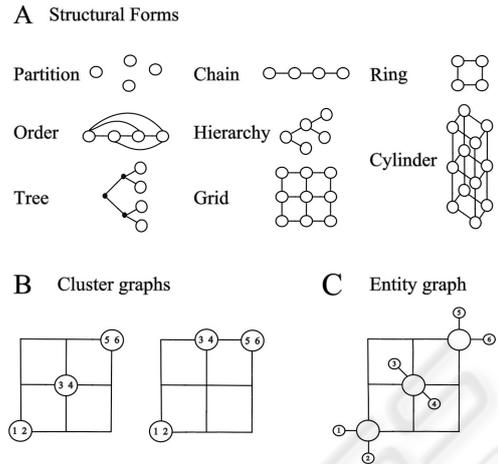


Figure 1: (A) Eight structural forms. (B) A structure on the left is compatible with grid structure while the other one is not. (C) The entity graph obtained from the left one in spot(B).

where $|S|$ is the number of nodes in graph S and $\theta \in (0, 1)$. Structure S is compatible with form F if it can be generated using the generative process (graph grammar) defined for F and if graph does not contain empty nodes when projected along its component dimensions. The latter notion is relevant in the case of grids and cylinders to prevent them from getting too complex, meaning many empty nodes in the graph.

Probability $P(S|F)$ is defined so that if the number of nodes in graph is large, it gives smaller values (bigger penalty for the model). Let $|S|$ be the number of nodes in graph S . When we write $\theta = \exp(-x)$, $x > 0$, log likelihood $\log P(S|F) = |S| \log(\theta) = -|S|x$ decreases by a constant x whenever an additional node is introduced. This way we tend to get small and simple graphs when using bigger values of x .

Secondly, we want to find a structure of a given form that fits best the data. This is achieved by maximizing the probability $P(D|S) = P(D|S_{ent})$ by assuming that feature values in data matrix D are independently generated from a multivariate Gaussian distribution with dimension for each node in the graph S_{ent} . This means that $P(D|S)$ is high if the features in matrix D vary smoothly over the graph S , that is, if entities nearby in S have similar feature values.

Let $W = [w_{ij}]$ be a weight matrix, i.e. a matrix which is comparable with edge lengths in entity graph S_{ent} . We define $w_{ij} = \frac{1}{e_{ij}}$ whenever nodes i and j are connected by an edge with length of e_{ij} , otherwise $w_{ij} = 0$. A generative model for a single feature vector f that favours now the feature values f_i to be similar in nearby nodes in S_{ent} is given by

$$P(f|W) \propto \exp\left(-\frac{1}{4} \sum_{i,j} w_{ij} (f_i - f_j)^2\right) = \exp\left(-\frac{1}{2} f^t \Delta f\right),$$

where $\Delta = E - W$, is the graph Laplacian and E is a diagonal matrix $e_{ii} = \sum_j w_{ij}$.

Finally, by assuming that a feature value f_i at any entity node has an a priori variance of σ^2 , we obtain a proper priori $f|W \sim N(0, \hat{\Delta}^{-1})$, where $\hat{\Delta}$ is Δ with $1/\sigma^2$ added on the diagonal of the first n positions. Note that the entity graph S_{ent} and weight matrix W are defined so that the entities are in the first n positions and the rest are the latent cluster nodes.

The priors for edge lengths e_{ij} and for σ are drawn from exponential distribution with parameter $\beta = 0.4$, as in (Kemp and Tenenbaum, 2008). Now we can compute the likelihood $P(D|S_{ent}, W, \sigma)$ and

$$\begin{aligned} \log P(D|S_{ent}, W, \sigma) &= \log \prod_{i=1}^m P(f^i|W) = \\ &= \frac{mn}{2} \log(2\pi) - \frac{m}{2} \log |\hat{\Delta}^{-1}| - \frac{1}{2} tr(\hat{\Delta} D D^t), \end{aligned} \quad (3)$$

where m is the number of feature vectors and f^i is the i th feature vector. By integrating out σ and edge weights we obtain the likelihood $P(D|S_{ent})$.

For further information, we refer on (Kemp and Tenenbaum, 2008).

2.2 Assigning New Data to a Learned Structure

It is interesting to notice that the algorithm does not necessarily need the feature data D itself but can use a covariance matrix $\frac{1}{m} D D^t$. As long as we know this covariance matrix, this approach can be used even though we actually do not have the actual features. This means that we can learn structures from some similarity matrix by assuming that this similarity matrix represents a covariance matrix of the data. We prefer to use similarity matrix due to its ability of being flexible to choose. If the metric of a feature space cannot reveal the relations between observations, it is worth using a suitable similarity measure. Later on, we run into previous matter in case of histogram data.

For classification purposes it would be convenient to be able to add new samples to a given structure. To assign a new sample, first we compute the similarities of the sample and the training samples used to build the structure. One by one, we go through all the cluster nodes and join the new sample by an edge to the node we are visiting to. In each case at a time, we can compute the likelihood in equation (3). The edge weight between the new sample and a cluster node is set to be the mean value of all edge weights in the graph. Although edge weights can be optimized, we found it to be quite ineffectual and slow when dealing with hundreds of samples. Finally, the sample is assigned to the cluster node which gives the highest

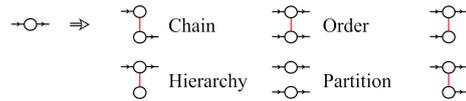


Figure 2: Node splitting rules for a meta-grammar. On the right side the correspondences with the primitive forms having the same grammar. Note that we cannot do the very first split by the 4th rule, because we want graph to be connected.

likelihood score. The probability $P(S|F)$ can obviously be forgotten since the cluster graph S is fixed.

2.3 A Meta-grammar

Using the eight presented forms can still lead to the circumstances where the structures simply cannot reveal the true, possibly complex, nature of the data. This leaves a room for a more generic form. As mentioned earlier, each form is characterised by the grammar it uses to split the nodes. When a grammar is an arbitrary mixture of several grammars, we call it a meta-grammar and a structure that uses this meta-grammar is called a meta structure. The idea of a meta-grammar was introduced in (Kemp and Tenenbaum, 2008) but was never used. The template of that meta-grammar was a combination of grammars of the six forms (all but grid and cylinder), so called primitive forms, illustrated in figure 1.

In this paper, we propose a slightly different meta-grammar and also put it into practice. The choices for node splits that our meta-grammar uses are shown in figure 2. First, we do not allow any nodes in the graph to be empty. For example the node splitting rule i.e. generative process used to create trees is not valid since the branch nodes will be empty. Secondly, we do not want to split the graph into two disjoint graphs so the very first split cannot be done by the generative process designed for partition structure. These notions allow us to make simple, connected graphs.

The two rightmost rules in figure 2 do not generate any natural structure themselves but give a necessary (and sufficient) complement to our meta-grammar. Using this meta-grammar provides the graph with more opportunities to organize itself. In practise, to split a node, we try each splitting rule present in a meta-grammar and choose the best one with respect to the likelihood (3).

One problem we face now is the difficulty of computing the probability $P(S|F)$. The normalization constant for the distribution in (2) is the sum

$$\sum_S P(S|F) = \sum_{k=1}^n S(n,k) C(F,k) \theta^k, \quad (4)$$

where $S(n,k)$ is the number of ways to partition n elements into k nonempty sets and $C(F,k)$ is the number

of structures of form F with k occupied cluster nodes.

When considering the form of meta structure and the number of possibilities how the meta-grammar can generate a structure with k nodes, we can clearly see that computing the exact number $C(F, k)$ is too hard. Anyway, we can easily find a rough upper limit for the number $C(F, k)$, since we can easily verify that the number of ways to draw edges in the case of k cluster nodes is $2^{k(k-1)/2}$. Thus, we have a lower boundary for likelihood $P(S|F)$.

3 THE ORGANIZATION OF VISUAL OBJECTS

We represent images as histograms of quantized descriptors. This bag-of-words (BOW) method has been successfully used in many papers such as (Sivic et al., 2008; Marszalek and Schmid, 2008; Bosch et al., 2006). Moreover, we extract descriptors from a grayscale image by computing the SIFT-features (Lowe, 2004) on a dense grid using an implementation available online (van de Sande et al., 2010).

As stated in section 2.2, we can use a similarity matrix of the histograms as an input to the algorithm. Besides, due to computational efficiency of the algorithm, we found the similarity matrix behave better than pure feature data. To compute the similarity matrix we transform χ^2 -distances between histograms to similarity values within range $[0, 1]$.

For reasonable execution time of the algorithm, we can use only a subset of samples to find the best structure and assign the rest of the samples to a learned structure, as described in section 2.2. More specifically, the samples which have a small variance in similarities are excluded from the training process.

Although, the algorithm decides itself which is the best structure, we can also examine different structures manually by comparing extracted log likelihoods, $\log P(S, F|D)$, of the model.

3.1 Assessing Structures using Classification

In evaluation, we use the same "classification overlap score", as described in (Sivic et al., 2008). Classification overlap score indicates how well the entities of a particular, manually labeled, object class are assigned to a single node in a tree. Obviously, we want high recall and high precision, so we want most of the class attributes to have a common node, which hopefully does not contain attributes from another class. The scale of this score is from 0 to 1. If score is 1 then

all object classes are fully separated at some node in the structure. Disadvantage is that this score is not directly usable in case of the other structures than tree or partition. One possibility is to modify the structures to be tree-shaped in a way we next describe.

As we know, the results of the algorithm are basically weighted graphs. Cluster graphs are graphs with no separate edges between entities and cluster nodes, as we have declared earlier. For each cluster graph, multiple clustering is obtained by running the Normalized Cuts (Shi and Malik, 2002) with varying the number of clusters. After this, we create a co-occurrence matrix of how many times each pair of the nodes in the graph appears in the same cluster. This matrix can be used as a similarity matrix for the hierarchical agglomerative clustering (Hastie et al., 2009), which creates a tree structure. After this operation, we are able to assess the classification accuracy based on the classification overlap score, regardless of the structure type.

It is apparent that the nature of structures suffers from this transformation and this measure suits better for tree-shaped structures. However, we have no other measure on hand at the moment so we trust that this measure gives at least a good estimate of the classification ability of each structure.

4 EXPERIMENTS

4.1 MSRC Dataset

We consider now a dataset MSRC-B1 (Winn et al., 2005) consisting of 240 images which are manually segmented to 12 different object classes. We use 543 segments of 9 different object classes: faces, cows, grass, trees, buildings, cars, airplanes, bicycles and sky. Other three classes: sheep, horses and ground are represented by only so few samples that we ignore them. This is exactly similar to (Sivic et al., 2008).

The SIFT-descriptors are computed at every 5th pixel in an image. Each image segment is then described by all visual words with centroids within the segment. We use 150 segments as a training data for finding the structure and assign the remainder to the



Figure 3: Log-likelihoods of each structure in case of data MSRC-B1. A constant has been added along y-axis so that the worst performing structure receives score close to zero. The best performing structure is marked by an asterisk.

Table 1: Image classification accuracy on MSRC-B1 data. Accuracy is measured by classification overlap score. The results of our method are the average of ten repeats.

method	topics	score
LDA	5/10/15/20	0.50/0.46/0.57/0.61 ^a
hLDA	-	0.72 ^a
partition	8	0.53
tree, ring	15	0.65
meta	16	0.65
other structures	11-15	0.44-0.63

^a (Sivic et al., 2008)

learned structure. We set $\theta = \exp(-200)$ for all forms considered and use a vocabulary of size one thousand words. The vocabulary is obtained by using all the samples.

4.1.1 Comparison of the Structures

In the case of the tree and partition structure we can compute the classification overlap score directly. For other structures we use the method described in section 3.1. The results are shown in table 1.

When compared to the results of LDA, the partition structure gives a better score with respect to the number of clusters. It gives eight clusters which is much closer to the number (9) of manually labeled classes than in the case of the best score (20 clusters) achieved by LDA. Most of the other structures give better results when compared to the best gained by LDA. We also see that hLDA gives better results in this case but our results are still comparable, in spite of forcing a tree-shape to the structures.

Figure 3 indicates how the meta structure is the model's choice of the best structure now. However, when comparing the classification accuracy, the ring, tree and meta structures all get the same score. The meta, ring and tree structures are presented in figures 4, 5. The clusters formed by each structure are quite similar to each other but the relations between clusters differ. It seems quite fair that meta structure wins when looking at the graphs. Although, in this case some of the clusters (for example buildings) are very different from the rest and meaningful connections are difficult to draw even for a human.

When comparing the results with the hierarchy in (Sivic et al., 2008), it appears that our approach creates more natural clustering than hLDA does. Unlike in our results, the number of small, meaningless clusters is large in their approach. We may say that by making more compact categorization, we lose some units in accuracy.

To return to the aspect of this paper, hLDA has one disadvantage when it comes to the possible relationships which a tree structure or any other structure

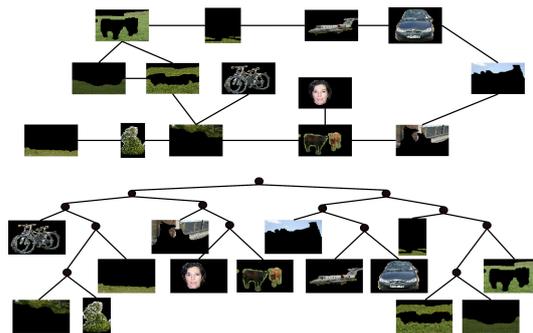


Figure 4: Uppermost the meta structure learned on the MSRC-B1 dataset of 543 image segments of 9 object classes. The images presenting each cluster are chosen to be the ones which are the most similar with the clusters' majority class. The edge lengths correspond to the edge weights. Down below a tree-shaped structure obtained from the meta structure by combining normalized cuts and hierarchical clustering.

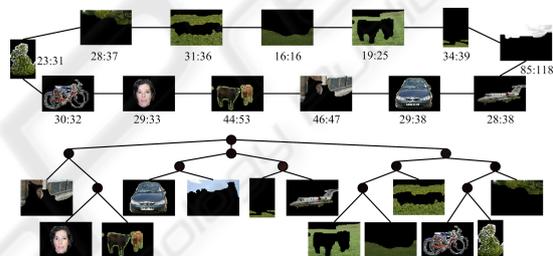


Figure 5: The ring and tree structures learned on the MSRC-B1 dataset. The images presenting each cluster are chosen to be the ones which are the most similar with the clusters' majority class. In the case of the ring structure, nodes are labeled by the number of images coming from the same class as the representative image of the node versus the number of all images in the node.

defined in advance cannot reveal. In the previous example, a tree-shaped structure worked as well as any since the image categories hardly shared anything in common. What about when the categories really have some underlying structure. How can we be sure that a certain chosen structure really match the data then? That is why it is good to consider a more generic view on creating the structure of images to gain deeper insight for any use of the structure.

4.2 Face Dataset

Let us then consider a situation where we have exactly one feature that is assumed to describe a set of images. If the values of this feature varies smoothly between images, we can imagine that it is not easy or even possible to get this information stored in a tree structure.

Example of the effects of this one feature can be found in case of faces. The feature is now the ori-

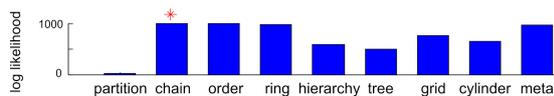


Figure 6: Log-likelihoods in case of an individual from the face data. A constant has been added along y-axis so that the worst performing structure receives a score close to zero. The best performing structure is marked by an asterisk.



Figure 7: Solid line represents the chain structure learned on an individual of a dataset of faces. Each node is represented by one face in the node. The dashed line correspond to the extra edges which the meta structure creates.

entation of faces. We use the Sheffield (previously UMIST) Face Database (Graham and Allinson, 1998) which consists of 564 images of 20 individuals. The range of poses vary from from profile to frontal views. We discover that the chain structure is the most probable, as indicated (for an individual) in figure 6.

The chain structure gives now a perfect solution in organizing faces according to their orientation (figure 7). It is also remarkable that the meta structure creates exactly the same clustering as the chain does and quite similar likelihood too, only few extra edges have been added to otherwise pure chain structure. However, we can see the capability of the meta structure to adapt to the natural structure of the data.

Another thing this example demonstrates (figure 6) is that tree-shaped structures cannot reveal the natural organization of face orientations. This concerns not only the structures presented in this paper but likely all the hierarchical organizations that exist.

5 CONCLUSIONS

We have presented a generic, unsupervised way to find the structure to describe image data. Previous methods in image categorization are able to create only an instance of a single, predefined form, usually tree form. Kemp's algorithm used in this paper defines a more generic view of finding the underlying structure in data. We have suggested how to apply the algorithm for visual objects and shown how this might help to find the more natural organization of a set of unlabeled images. In addition, we proposed our prototype for the most generic structure, meta structure. This creates graphs which can capture the relations in data even more accurately and can adapt to any underlying structure. The categorization or classifica-

tion results are competitive with topic discovery models (LDA, hLDA). Moreover, the way we can present image categories and the relations between categories seems to be more natural and definitely more flexible than in the state-of-the-art methods.

REFERENCES

- Ahuja, N. and Todorovic, S. (2007). Learning the taxonomy and models of categories present in arbitrary images. In *Proc. ICCV*.
- Bart, E., Porteous, I., Perona, P., and Welling, M. (2008). Unsupervised learning of visual taxonomies. In *Proc. ICPR*.
- Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.
- Bosch, A., Zisserman, A., and Muoz, X. (2006). Scene classification via pls. In *Proc. ECCV*.
- Graham, D. and Allinson, N. (1998). Characterizing virtual eigensignatures for general purpose face recognition. *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences*, 163:446–456.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, New York.
- Hoffmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196.
- Kemp, C. and Tenenbaum, J. (2008). The discovery of structural form. In *Proceedings of the National Academy of Sciences*. <http://www.psy.cmu.edu/~ckemp/>.
- Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- Marszalek, M. and Schmid, C. (2008). Constructing category hierarchies for visual recognition. In *Proc. ECCV*.
- Parikh, D. and Chen, T. (2007). Unsupervised learning of hierarchical semantics of objects (hSOs). In *Proc. CVPR*.
- Shi, J. and Malik, J. (2002). Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905.
- Sivic, J., Russell, B., Efros, A., Zisserman, A., and Freeman, W. (2005). Discovering object categories in image collections. In *Proc. ICCV*.
- Sivic, J., Russell, B., Zisserman, A., Freeman, W., and Efros, A. (2008). Unsupervised discovery of visual object class hierarchies. In *Proc. CVPR*.
- van de Sande, K., Gevers, T., and Snoek, C. (2010). Evaluating color descriptors for object and scene recognition. *IEEE TPAMI*, (in press).
- Winn, J., Criminisi, A., and Minka, T. (2005). Object categorization by learned universal visual dictionary. In *Proc. ICCV*.