Adaptive Feature Selection in Image Segmentation

Volker Roth and Tilman Lange
ETH Zurich, Institut of Computational Science
Hirschengraben 84, CH-8092 Zurich
{vroth, langet}@inf.ethz.ch

Abstract. Most image segmentation algorithms optimize some mathematical similarity criterion derived from several low-level image features. One possible way of combining different types of features, e.g. color- and texture features on different scales and/or different orientations, is to simply stack all the individual measurements into one high-dimensional feature vector. Due to the nature of such stacked vectors, however, only very few components (e.g. those which are defined on a suitable scale) will carry information that is *relevant* for the actual segmentation task. We present an approach to combining *segmentation* and adaptive *feature selection* that overcomes this relevance determination problem. All free model parameters of this method are selected by a resampling-based stability analysis. Experiments demonstrate that the built-in feature selection mechanism leads to stable and meaningful partitions of the images.

1 Introduction

The goal of image segmentation is to divide an image into connected regions that are meant to be semantic equivalence classes. In most practical approaches, however, the semantic interpretation of segments is not modeled explicitly. It is, rather, modeled indirectly by assuming that semantic similarity corresponds with some mathematical similarity criterion derived from several low-level image features. Following this line of building segmentation algorithms, the question of how to combine different types of features naturally arises. One popular solution is to simply stack all different features into a high-dimensional vector, see e.g [1]. The individual components of such a feature vector may e.g. consist of color frequencies on different scales and also on texture features both on different scales and different orientations. The task of grouping such high-dimensional vectors, however, typically poses two different types of problems: on the technical side, most grouping algorithms become increasingly instable with growing input space dimension. Since for most relevant grouping criteria no efficient globally optimal optimization algorithms are known, this "curse of dimensionality" problem is usually related to the steep increase of local minima of the objective functions. Apart from this technical viewpoint, the special structure of feature vectors that arise from stacking several *types* of features poses another problem which is related to the *relevance* of features for solving the actual segmentation task. For instance, texture features on one particular scale and orientation might be highly relevant for segmenting a textile pattern from an unstructured background, while most other feature dimensions will basically contain useless "noise" with respect to this particular task. Treating all features equally, we cannot expect to find a reliable decomposition of the image into meaningful classes. Whereas the "'curse of dimensionality"-problem might be overcome by using a general regularization procedure which restricts the intrinsic complexity of the learning algorithm used for partitioning the image, the special nature of stacked feature vectors particularly emphasizes the need for an adaptive *feature selection* or *relevance determination* mechanism.

In *supervised* learning scenarios, feature selection has been studied widely in the literature. Selecting features in *unsupervised* partitioning scenarios, however, is a much harder problem, due to the absence of class labels that would guide the search for relevant information. Problems of this kind have been rarely studied in the literature, for exceptions see e.g. [2, 9, 15]. The common strategy of most approaches is the use of an iterated stepwise procedure: in the first step a set of hypothetical partitions is extracted (the *clustering* step), and in the second step features are scored for relevance (the *relevance determination* step). A possible shortcoming is the way of combining these two steps in an “ad hoc” manner: firstly, standard relevance determination mechanism do not take into account the properties of the clustering method used. Secondly, most scoring methods make an implicit independence assumption, ignoring feature correlations. It is thus of particular interest to combine feature selection and partitioning in a more principled way. We propose to achieve this goal by combining a Gaussian mixture model with a Bayesian relevance determination principle. Concerning computational problems involved with selecting “relevant” features, a Bayesian inference mechanism makes it possible to overcome the combinatorial explosion of the search space which consists of all subsets of features. As a consequence, we are able to derive an efficient optimization algorithm. The method presented here extends our previous work on combining clustering and feature selection by making it applicable to multi-segment problems, whereas the algorithms described in [13, 12] were limited to the two-segment case.

Our segmentation approach involves two free parameters: the number of mixture components and a certain constraint value which determines the average number of selected features. In order to find reasonable settings for both parameters, we devise a resampling-based stability model selection strategy. Our method follows largely the ideas proposed in [8] where a general framework for estimating the number of clusters in unsupervised grouping scenarios is described. It extends this concept, however, in one important aspects: not only the model order (i.e. the number of segments) but also the model complexity for a fixed model order (measured in terms of the number of selected features) is selected by observing the stability of segmentations under resampling.

2 Image Segmentation by Mixture Models

As depicted in figure 1 we start with extracting a set of N image-sites, each of which is described by a stacked feature vector $\boldsymbol{x}_i \in \mathbb{R}^d$ with d components. The stacked vector usually contains features from different cues, like color histograms and texture responses from Gabor filters, [10]. For assigning the sites to classes, we use a Gaussian mixture model with K mixture components sharing an identical covariance matrix Σ. Under this model, the data log-likelihood reads

$$l^{mix} = \sum_{i=1}^{N} \log \left(\sum_{\nu=1}^{K} \pi_\nu \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_\nu, \Sigma) \right), \tag{1}$$

where the mixing proportions π_ν sum to one, and ϕ denotes a Gaussian density. It is well-known that the classical *expectation-maximization* (EM) algorithm, [3], provides a convenient method for finding both the component–membership probabilities and the model parameters (i.e. means and covariance) which maximize l^{mix}. Once we have trained the mixture model (which represents a parametric density on $\mathbb{R}^d$) we can easily predict the component–membership probabilities of sites different from those contained in the training set by computing Mahalonobis distances to the mean vectors.

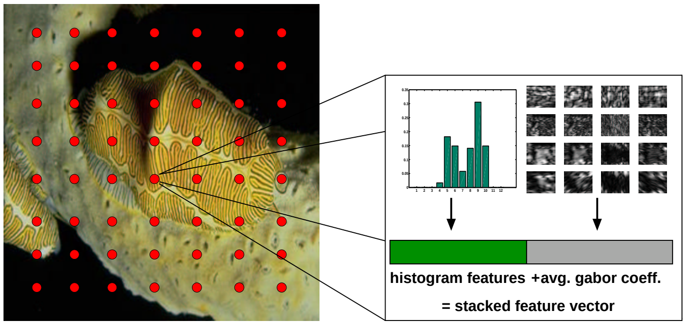

Fig. 1. Image-sites and stacked feature vectors (schematically).

2.1 Gaussian Mixtures and Bayesian Relevance Determination

In order to incorporate the feature selection mechanism into the Gaussian mixture model, the M-step of the EM-algorithm undergoes several reformulations. Following [5], the M-step can be carried out by linear discriminant analysis (LDA) which uses the "fuzzy labels" estimated in the preceding E-step. LDA is equivalent to an *optimal scoring* problem (cf. [6]), the basic ingredient of which is a linear regression procedure against the class-indicator variables. Since space here precludes a more detailed discussion of the equivalence of the classical M-step and indicator regression, we refer the interested reader to the above references and we will concentrate in the following on the aspect of incorporating the feature selection method into the regression formalism.

A central ingredient of optimal scoring is the "blurred" response matrix $\tilde{Z}$, whose rows consist of the current membership probabilities. Given an initial $(K \times K-1)$ *scoring* matrix Θ, a sequence of $K-1$ linear regression problems of the form

$$\text{find } \boldsymbol{\theta}_j, \boldsymbol{\beta}_j \text{ which minimize } \|\tilde{Z}\boldsymbol{\theta}_j - X\boldsymbol{\beta}_j\|_2^2, \quad j = 1, \dots, K-1 \tag{2}$$

is solved. X is the data matrix which contains the stacked feature vectors as rows. We incorporate the feature selection mechanism into the regression problems by specifying a prior distribution over the regression coefficients $\boldsymbol{\beta}$. This distribution has the form of an *Automatic Relevance Determination* (ARD) prior: $p(\boldsymbol{\beta}|\boldsymbol{\vartheta}) \propto \exp[-\sum_{i=1}^{d} \vartheta_i \beta_i^2]$. For each regression coefficient, the ARD prior contains a free hyperparameter ϑ_i, which encodes the "relevance" of the i-th variable in the linear regression. Instead of explicitly selecting these relevance parameters, which would necessarily involve a search over of all possible subsets of features, we follow the Bayesian view of [4] which consists of "averaging" over all possible parameter settings: given exponential hyperpriors, $p(\vartheta_i) = \frac{\gamma}{2}\exp\{-\frac{\gamma\vartheta_i}{2}\}$, one can *analytically integrate out* the relevance-parameters from the prior distribution over the coefficients. Switching to the maximum a posteriori (MAP) solution in log-space, this Bayesian marginalization directly leads to the following ℓ_1–constrained regression problems:

$$\text{minimize} \;\; \|\tilde{Z}\boldsymbol{\theta}_j - X\boldsymbol{\beta}_j\|_2^2 \;\; \text{subject to} \;\; \|\boldsymbol{\beta}_j\|_1 < \kappa\,, \quad j = 1, \ldots, K-1, \qquad (3)$$

where $\|\boldsymbol{\beta}_j\|_1$ denotes the ℓ_1 norm of the vector of regression coefficients in the j-th regression problem. This model is known as the LASSO, see [14]. A highly efficient algorithm for optimizing the LASSO model can be found in [11].

According to [5], in the optimal scoring problem the regression fits are followed by finding a sequence of optimal orthogonal scores $\widehat{\Theta}$ which maximize $trace\{\Theta^\top \tilde{Z}^\top XB\}$, where the matrix B contains the optimal vectors $\widehat{\boldsymbol{\beta}}_1, \ldots, \widehat{\boldsymbol{\beta}}_{K-1}$ as columns. In the unconstrained case described in [5], this maximization amounts to finding the $K-1$ largest eigenvectors $\boldsymbol{v}_i$ of the symmetric matrix $M \equiv \Theta^\top \tilde{Z}^\top XB$. The matrix B is then updated as $B \leftarrow BV$. In our case with active ℓ_1 constraint, the matrix M is no longer guaranteed to be symmetric. Maximization of the symmetrized problem $M_{\text{sym}} \equiv 1/2 \cdot M^\top M$, however, may be viewed as a natural generalization. We thus propose to find the optimal scores by an eigen-decomposition of M_{sym}.

Summing up. For feature selection, we ideally would like to estimate the value of a *binary* selection variable: $\mathcal{S}_i$ equals one, if the i-th feature is considered relevant for the given task, and zero otherwise. Taking into account feature correlations, however, estimation of $\mathcal{S}$ involves searching the space of all possible subsets of features. In the Bayesian ARD formalism, this combinatorial explosion of the search space is overcome by relaxing the binary selection variable to a real-valued relevance parameter. Following a Bayesian inference principle, we introduce hyper-priors and integrate out these relevance parameters, and we finally arrive at a sequence of ℓ_1–constrained LASSO problems, followed by an eigen-decomposition to find the optimal scoring vectors. It is of particular importance that this method combines the issues of grouping and feature selection in a principled way: both goals are achieved simultaneously by optimizing the *same objective function*, which is simply the constrained data log-likelihood.

3 Model Selection and Experimental Evaluation

Our model has two free parameters, namely the number of mixture components and the value of the ℓ_1–constraint κ. Selecting the number of mixture components is referred to as the model order selection problem, whereas selecting the number of features can be viewed as the problem of choosing the complexity of the model. We now describe a method for selecting both parameters by observing the *stability* of segmentations.

Selecting the model complexity. We will usually find many potential splits of the data into clusters, depending on how many features are selected: if we select only one feature, it is likely to find many competing hypotheses for splits, since most of the feature vectors vote for a different partition. Taking into account the problem of noisy measurements, the finally chosen partition will probably tell us more about the exact noise realization than about meaningful splits. If, on the other hand, we select too many features, many of them will be irrelevant for the actual task, and with high probability, the EM-algorithm will find suboptimal solutions. Between these two extremes, we can hope to find relatively stable splits, which are robust against noise and also against inherent instabilities of the optimization method. For a fixed model order, we use the following algorithm for assessing the value of κ:

1. **Sampling:** draw randomly 100 datasets (i.e. sets of sites), each of which contains N sites. For each site extract the stacked feature vector.
2. **Stability analysis:** for different constraint values κ repeat:
   (a) **Clustering:** For each set of sites, train a mixture model with K modes. Assign each of the the sites in the i-th set to one of K groups, based on the estimated membership probabilities. Store the labels $\boldsymbol{l}_i$ and the model parameters $\boldsymbol{p}_i$.
   (b) For each pair $(i, j),\ i \neq j$ of site sets do
   Prediction: use the i-th mixture model (we have stored all parameters in $\boldsymbol{p}_i$) to predict the labels of the j-th sample. Denote these labels by $\boldsymbol{l}_i^j$;
   Distance calculation: calculate the permutation–corrected Hamming distance between original and predicted labels by minimizing over all permutations π:

$$d_{i,j}^{\text{Hamming}} = \min_\pi \textstyle\sum_{k=1}^N 1 - \delta\{\boldsymbol{l}_j(k), \pi(\boldsymbol{l}_i^j(k))\}, \qquad (4)$$

   (δ denotes the Kronecker symbol), and store it in the (100×100) matrix D. The minimization over all permutations can be done efficiently by using the Hungarian method for bipartite matching with time complexity $O(K^3)$, [7].
   (c) **Partition clustering & prototype extraction:** use Wards agglomerative method to cluster the matrix D. Stop merging partition-clusters if the average within-cluster Hamming distance exceeds a threshold $\epsilon = \gamma \cdot (1 - 1/K)$ proportional to the expected distance in a random setting (for random labellings we expect an average distance of $(1 - 1/K)$). In the experiments we have chosen $\gamma = 0.05 = 5\%$. In each partition-cluster, select the partition which is nearest to the cluster centroid as the prototypical partition.

Selecting the model order. In order to select a suitable number K of mixture components, we repeat the whole complexity selection process for different values of K. We consider that K-value as the most plausible one, for which the percentage of partitions in the individual partition clusters attains a maximum. Since in most unsupervised grouping problems there is more than one "interesting" interpretation of the data, we might, however, gain further insights by also studying other K-values with high but not maximal stability, see figure 4 for an example.

Figures 2 and 3 show the results of the model selection process for an artificial image with five segments. Two of the segments are solely defined in terms of different grey value distributions without any texture information. Two other segments, on the other hand, contain the same texture pattern in different orientations which makes them indistinguishable in the terms of grey values. In order to capture both types of information, at each site we stacked 12 grey value histogram bins and 16 Gabor coefficients on different scales and orientations into a 28-dimensional feature vector. The features are normalized to zero mean and unit variance across the randomly chosen set of image-sites. The right panel of figure 2 depicts the outcome of the model-order selection process. The stability curve shows a distinct maximum for 5 mixture components. 83% of all partitions found in 100 resampling experiments are extremely similar: their average divergence is less than 5% of the expected divergence in a random setting.

Figure 3 gives more insight into the model-complexity selection process for this most stable number of mixture components. For small values of the ℓ_1 constraint κ only very few features are selected which leads to highly fluctuating segmentations. This observation is in accordance with our expectation that the selection of only a few single features would be highly sensitive to the sampling noise. The full model containing all features also turns out to be rather instable, probably due to the irrelevance of most feature dimensions. For the task of separating e.g. the two segments which contain the same texture in different orientations, all color features are basically uninformative noise dimensions. Between these two extremes, however, we find a highly stable segmentation result. On average, 13 features are automatically selected. More important than this average number, however, is the fact that in each of the 4 regression fits (we have $K = 5$ mixture components and thus $K - 1 = 4$ fits) the features are selected in an *adaptive* fashion: in one of the regression problems almost exclusively grey-value features are selected, whereas two other regression fits mainly extract texture information. By combining the 4 regression fits the model is able to extract both types of information while successfully suppressing the irrelevant noise content.

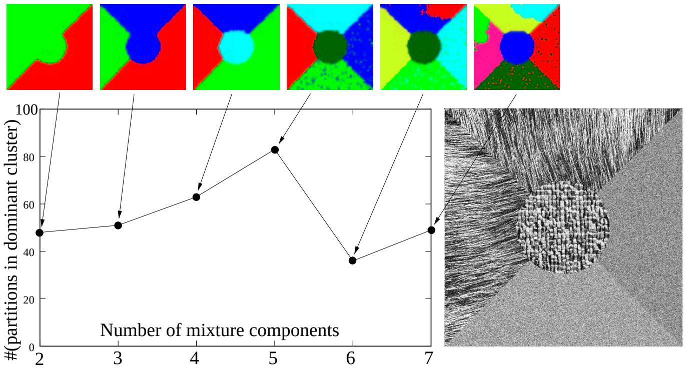

Fig. 2. Model-order selection by resampling: stability of segmentations (measured in terms of percentage of highly similar partitions) vs. number of mixture components. Right: input image.

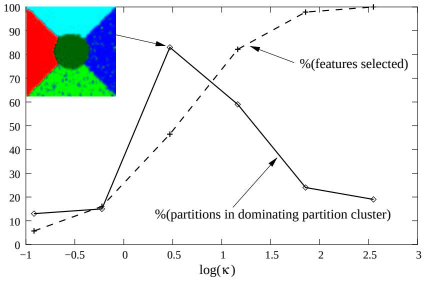

Fig. 3. Selecting the model-complexity for fixed number of mixture components $K = 5$. Solid curve: stability vs. ℓ_1 constraint κ. Dashed curve: number of selected features

Real word examples. We applied our method to several images from the corel database. Figure 4 shows the outcome of the whole model selection process for an image taken from the corel "shell-textures" category, see figure 5. The stability curve for assessing the correct model order favors the use of two mixture components. In this case, the most stable partitions are obtained for a highly constrained model which employs on average only 2 features (left panel). A closer look on the partition clusters show that there is a bimodal distribution of cluster populations: 44 partitions found in 100 resampling experiments form a cluster that segments out the textured shell from the unstructured environment (only texture features are selected in this case), whereas in 37 partitions only color features are extracted, leading to a bipartition of the image into shadow and foreground.

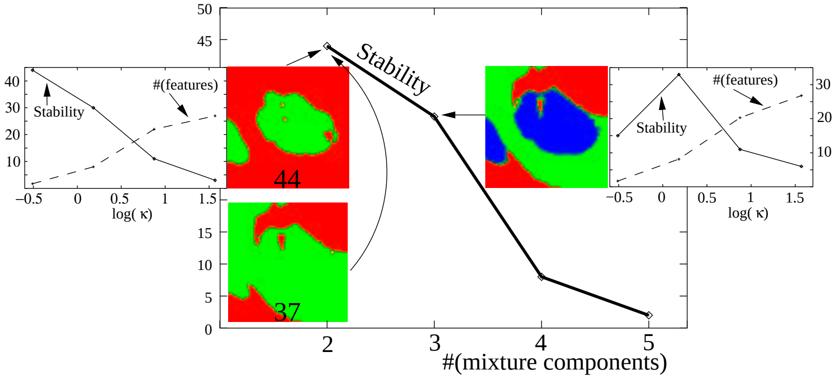

Fig. 4. A shell image from the corel database: model selection by resampling.

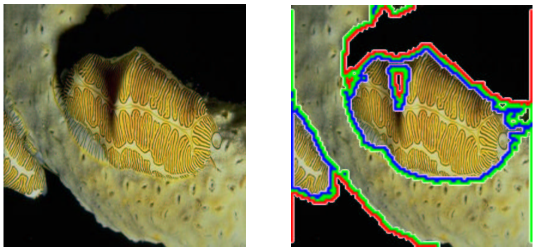

Fig. 5. The shell image and the three-component segmentation solution

Both possible interpretations of the image are combined in the three-component model depicted in the right panel. The image is segmented into three classes that correspond to “shell”, “coral” and “shadow”. The most stable three-component model uses a combination of five texture and three color features. This example demonstrates that due to the unsupervised nature of the segmentation problem, sometimes there are more than one “plausible” solutions. Our feature selection process is capable of exploring such ambiguities, since it provides the user not only with a single optimal model but with a ranked list of possible segmentations. The reader should notice that also in this example the restriction of the model complexity enforced by the ℓ_1 constraint is crucial for obtaining stable segmentations. We applied our method to several other images from the corel database, but due to space limitations we refer the interested reader to our web-page www.inf.ethz.ch/˜vroth/segments_dagm.html.

4 Discussion

In image segmentation, one often faces the problem that relevant information is spread over different cues like color and texture. And even within one cue, different scales might be relevant for segmenting out certain segments. The question of how to combine such different types of features in an optimal fashion is still an open problem. We present a method which overcomes many shortcomings of "naively" stacking all features into a combined high-dimensional vector which then enters a clustering procedure. The main ingredient of the approach is an automatic feature selection mechanism for distinguishing between "relevant" and "irrelevant" features. Both the process of grouping sites to segments and the process of selecting relevant information are subsumed under a common likelihood framework which allows the algorithm to select features in an adaptive task-specific way. This adaptiveness property makes it possible to combine the relevant information from different cues while successfully suppressing the irrelevant noise content. Examples for both synthetic and natural images effectively demonstrate the strength of this approach.

Acknowledgments. The authors would like to thank Joachim M. Buhmann for helpful discussions and suggestions.

References

1. S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color- and texture-based image segmentation using the expectation-maximization algorithm and its application to content-based image retrieval. In *Int. Conf. Computer Vision*, 1998.
2. A. Ben-Dor, N. Friedman, and Z. Yakhini. Class discovery in gene expression data. In *Procs. RECOMB*, pages 31–38, 2001.
3. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B*, 39:1–38, 1977.
4. M. Figueiredo and A. K. Jain. Bayesian learning of sparse classifiers. In *CVPR2001*, 2001.
5. T. Hastie and R. Tibshirani. Discriminant analysis by gaussian mixtures. *J. R. Stat. Soc. B*, 58:158–176, 1996.
6. T. Hastie, R. Tibshirani, and A. Buja. Flexible discriminant analysis by optimal scoring. *J. Am. Stat. Assoc.*, 89:1255–1270, 1994.
7. H.W. Kuhn. The hungarian method for the assignment problem. *Naval Res. Logist. Quart.*, 2:83–97, 1955.
8. T. Lange, M. Braun, V. Roth, and J.M. Buhmann. Stability-based model selection. In *Advances in Neural Information Processing Systems*, volume 15, 2003.
9. M.H. Law, A.K. Jain, and M.A.T. Figueiredo. Feature selection in mixture-based clustering. In *Advances in Neural Information Processing Systems*, volume 15, 2003.
10. B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(8):837–842, 1996.
11. M. Osborne, B. Presnell, and B. Turlach. On the lasso and its dual. *J. Comput. Graph. Stat.*, 9:319–337, 2000.
12. V. Roth and T. Lange. Feature selection in clustering problems. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.
13. Volker Roth and Tilman Lange. Bayesian class discovery in microarray datasets. *IEEE Trans. on Biomedical Engineering*, 51(5), 2004.
14. R.J. Tibshirani. Regression shrinkage and selection via the lasso. *JRSS B*, 58:267–288, 1996.
15. A. v.Heydebreck, W. Huber, A. Poustka, and M. Vingron. Identifying splits with clear separation: a new class discovery method for gene expression data. *Bioinformatics*, 17, 2001.