Generation of task-specific segmentation procedures as a model selection task*

Ralf Herbrich[†] and Tobias Scheffer[‡]

Technische Universität Berlin,

[†]Statistics Research Group, Sekr. FR 6-9,

[‡]Artificial Intelligence Research Group, Sekr. FR 5-8,

Franklinstr. 28/29. D-10587 Berlin, Germany

email: ralfh|scheffer@cs.tu-berlin.de

April 17, 1998

Abstract

In image segmentation problems, there is usually a vast amount of filter operations available, a subset of which has to be selected and instantiated in order to obtain a satisfactory segmentation procedure for a particular domain. In supervised segmentation, a mapping from features, such as filter outputs for individual pixels, to classes is induced automatically. However, since the sample size required for supervised learning grows exponentially in the number of features it is not feasible to learn a segmentation procedure from a large amount of possible filters. But we argue that automatic model selection methods are able to select a region model in terms of some filters.

*This paper is printed in Proceedings of Workshop on Visual Information Processing, Sydney 1997, pp. 11-21

We propose a wrapper algorithm that performs this task. We present results on artificial textured images (Brodatz) and report on our experiences with x–ray images.

Keywords: model based image segmentation, automatic model selection, learning pixel classifier, texture segmentation

1 Introduction

The partitioning of an image into regions with similar properties based on sample segmentations is called supervised segmentation. Given a set of available features, such as grey values or outputs of *a priori* given filter operations and a sample of classified images, the task is to induce a segmentation procedure which approximates the segmentation observed on the sample images well. In general, the potential number of features is very large. Since the sample size required in order to learn a good hypothesis (independent of the algorithm which is used) grows exponentially in the number of features (Blumer *et al.*, 1987), it is not feasible to automatically learn a segmentation procedure from a large library of filters. Known supervised segmentation algorithms therefore require a restricted region model in terms of few, relevant features, *e.g.*, based on the frequency spectrum (Bovik *et al.*, 1990), the second order statistics of the image (Houzelle & Giraudon, 1992), or the assumption that the image is a Markov random field (Cross & Jain, 1983).

Automatic model selection (for an overview, see, *e.g.*, Kearns *et al.*, 1997) deals with the question how complex an optimal model should be, depending on the problem and the available sample size. Too small a model is unlikely to contain any "good" hypothesis, while too rich a model will inevitably incur over-fitting – this problem is often referred to as the bias-variance tradeoff. Wrapper algorithms (Kohavi & John, 1997) find optimal models by repeatedly testing the best hypotheses of different models on hold-out sets. In Section 2, we describe the model selection problem briefly. Then, we discuss wrappers for feature subset se-

lection and show how they can be applied for image segmentation. In Section 5, we present some empirical results on the Brodatz texture data set and report on experiences with the segmentation of x-ray images.

2 Classification and model selection

A *classification problem* can be defined in terms of a distribution P_D, $D = X \times Y$, over pairs of instances $x \in X$ and class labels $y \in Y$. Here, instances are feature vectors $\{x_i\} \in \Re^n$. Unfortunately, the underlying distribution P_D is not known. There is, however, a finite sequence of labelled training instances $S = \langle (x_i, y_i) \rangle$, drawn according to P_D. A *hypothesis* $h : X \to Y$ is a mapping from instances to class labels. The *true* error[1] E_D of a hypothesis h is then defined by

$$E_D(h) \;=\; \int_{X \times Y} L(h(x), y)\, dP_D(x, y) \tag{1}$$

where $L(\hat{y}, y)$ is the loss and returns 0 iff $\hat{y} = y$ 1; otherwise.
The true error of a hypothesis is therefore the expected error of the hypothesis over the space of labelled instances $X \times Y$ with respect to the distribution P_D. Unfortunately, since the distribution P_D is unknown, only the *empirical error* on the training sample S, $E_S(h) = 1/|S| \sum_{(x,y) \in S} L(h(x), y)$, can be observed.
Given a particular hypothesis language, or *model*, $\mathcal{H}_i$, a *learning algorithm* finds the hypothesis $h_i^{min} \in \mathcal{H}_i$, that minimizes the empirical error E_S. Unfortunately, given some large hypothesis space $\mathcal{H}$, selecting the hypothesis which minimizes the empirical E_S error among all hypotheses is usually not a good strategy to find a hypothesis with low true error E_D. In general, the sample size required to assure that a hypothesis which has a low empirical error E_S also has a low true error E_D has to grow exponentially in the size of the model[2] (*e.g.*, the number of attributes)

[1] sometimes also called *generalization* error

[2] Learning theory explains and quantifies this over-fitting effect. The more hypotheses one assesses on a sample, the more optimistically assessed the apparently best hypothesis will be –

(Blumer *et al.*, 1987).

Model selection algorithms stratify the hypothesis space into increasingly large models $\mathcal{H}_1, \mathcal{H}_2, \ldots \subset \mathcal{H}$. The curve of true errors over the increasingly complex models is normally U-shaped. There are two classes of approaches to decide, which model incurs an optimally low error: hold-out testing – *e.g.*, (Kohavi & John, 1997) – and complexity penalization based approaches, such as Minimum Description Length (Rissanen, 1985), or Guaranteed Risk Minimization (Vapnik, 1982).

3 Wrappers for model selection

In order to find the best model we need to estimate the true error E_D of the best hypothesis h_i^{min} of each model $\mathcal{H}_i$. A simple attempt is to split the training sample $S = T \cup H$ into a (smaller) training set T and a hold-out set H. After learning a hypothesis h_T^{min} that minimizes E_T, the estimate $E_H(h_T^{min})$ on the hold-out set H is used to compare the best hypotheses of each model. This estimate is not biased by the size of the model because only one hypothesis from each model is assessed on the hold-out set. Unfortunately, this estimate is subject to high variance. A variant of hold-out testing is n-fold cross validation. Here, the mean of n estimates of the error on disjoint subsets of the training sample incurs a lower variance.

All subsets of the given set of attributes form the *feature space*. In the case of n features the feature space has exactly 2^n elements. Note, however, that there are $\binom{n}{n/2}$ subsets containing $n/2$ out of n features while there is only 1 subset that contains none, or all n features respectively. Since evaluating these models via cross validation incurs some variance, the apparently best model with $n/2$ features will be assessed more optimistically than the only model with all features – which would incur an inappropriate bias towards models with close to $n/2$ features. In order to minimize this bias, wrappers usually perform a greedy search.

which increases the chance of selecting an apparently good hypothesis with a high true error.

On each step i, forward feature subset selection adds one new feature to a list of $i-1$ already selected features. In order to select a new feature, all the different models that contain the $i-1$ features selected earlier and one of the remaining $n-(i-1)$ features are enumerated and assessed via cross validation. The process stops as soon as adding another feature does not improve the cross validation error. If the total number of features is large compared to the number of features actually selected, this procedure incurs only a negligibly small bias towards smaller models (while n different models including one feature are assessed, only $n-1$ models with two features are tested).

4 Automatic model selection in supervised segmentation

In this section, we show how the introduced methods can be used to automatically perform the model selection step in supervised segmentation. A supervised segmentation problem is constituted by a distribution of images and corresponding segmentations, and a set of *a priori* known features corresponding to the pixels. This distribution of images induces a distribution on pairs of pixel descriptions and classes (corresponding to the segments). The incorporation of automatic model selection in this task is outlined in the next three steps.

Feature extraction At the first stage, for each pixel of the sample images a fixed set of features has to be extracted (*e.g.*, by convolving the image with – task specific – filters). This yields an n-dimensional feature vector $x^{(i)}$ associated with each pixel i.

Feature subset selection Each pixel i in the sample images is associated with a class label $y^{(i)}$ corresponding to the region it lies on. Thus, a training sample $S = \langle (x^{(i)}, y^{(i)}) \rangle$ is formed. A model selection algorithm (*e.g.*, a wrapper for feature subset selection, see Section 3) needs to select a subset of features that forms a model that minimizes the expected true error E_D.

Induction Within the selected model, a hypothesis h^{min} has to be induced. This hypothesis is the resulting segmentation procedure mapping pixel features to a class label which corresponds to the region the pixel is associated with.

5 Experiments and Results

In this section we will present some results of our algorithm. In each experiment we fixed the size of the images to 256x256 and the size of the training set to 1000 randomly drawn pixels, *i.e.,* 1.5% of all pixels. When performing feature subset selection we use 10-fold cross validation. In our implementation, we use a learning algorithm that induces ripple down rules (Scheffer, 1996) which are rules with hierarchically nested exceptions. Ripple down rules incur a language bias similar to decision trees, the only difference being that every node in a ripple down rules contains a conjunction of interval constraints rather than a single test.

Texture segmentation In these experiments, the task is to learn the segmentation of a patchwork image consisting of Brodatz textures taken from (Brodatz, 1966). We use the two images in Figure 1 (above tables (a) and (b)).
In the feature extraction phase, we convolved the image with a bank of Gabor filters to compute the amplitudes. Gabor filters are well suited if the regions have distinct peaks in their frequency spectrum (Bovik *et al.*, 1990; Weldon & Higgins, 1997). The resulting segmentations are shown in Figure 1. In simple models (first columns of Figure 1) the algorithm finds the best segmentation procedures with and without feature subset selections. As one can see in the leftmost column, feature subset selection yields barely any gain if the available set of features is so small that almost any subset misses relevany information. As the number of filters is increased to 32, the results improve and feature subset selection gains significant enhancement of the segmentation, compared to pure learning in the complete hypothesis language. Feature subset selection reduces the number of features taken into account from 32 to 8, and from 64 to 14, respectively. As

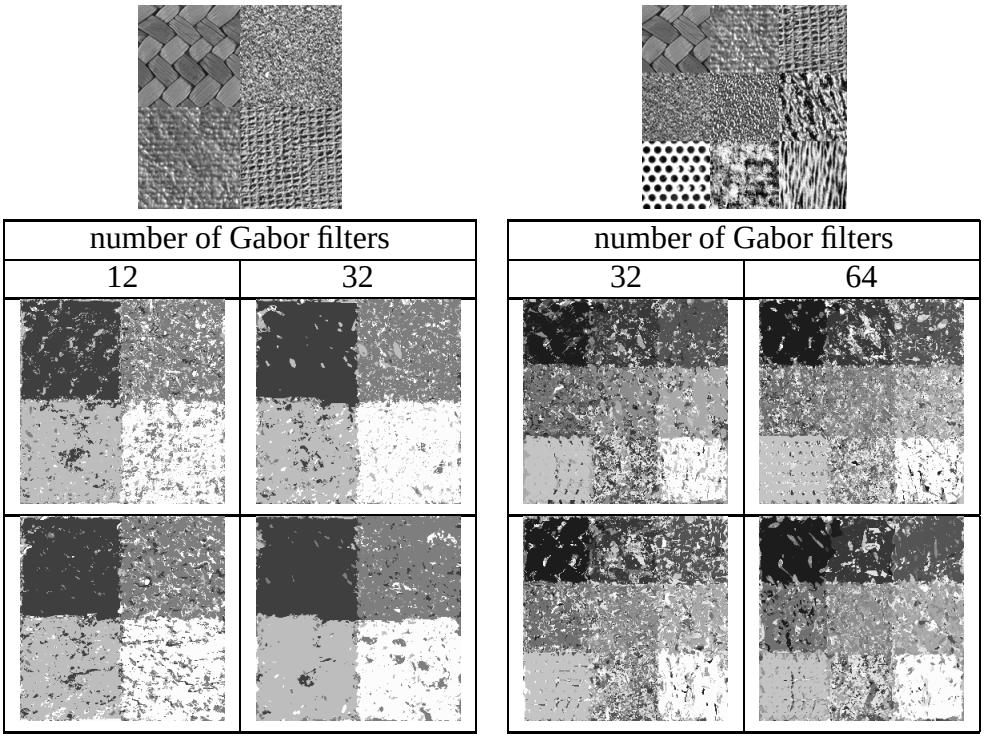
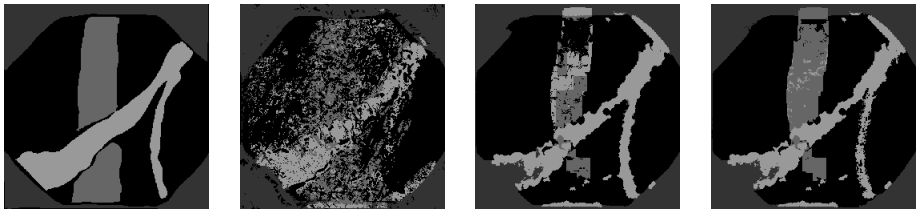


Figure 1: Segmentation results on **(a)** 4 and **(b)** 9 textures. First row: results w/o feature subset selection; second row: forward feature subset selection.

the number of available training examples is increased further, the set of features selected by forward feature subset selection remains stable.

Segmentation of x-ray images Here, the task is to learn the segmentation of x-ray images of the chest. The sample is drawn from nine images. The classes that need to be distinguished are “spine”, “colon”, “tissue”’ and “collimator”. The data set contains the images with their optimal segmentations and filter images of six highly non-linear filters, which were hand-crafted and strongly adapted to

the domain. Additional features are grey values and a bank of 32 Gabor filter convolutions. Thus, we obtained a 39-dimensional feature vector $x^{(i)}$ for each pixel i.
Exemplary, Figure 2 shows the results on one image, without feature subset selection (b), with forward feature subset selection (c) and, additionally, the enhancement obtained by combining the learning algorithm with a majority voting technique (d).



(a) (b) (c) (d)

Figure 2: **(a)** optimal segmentation; **(b)** without feature subset selection **(c)**; with forward feature subset selection; **(d)** with committee of 10 hypotheses

If we induce a segmentation procedure without model selection, the results are unacceptably poor. (Figure 2 (b)). eature subset selection drops all the Gabor features and finds a good hypothesis consisting only of a combination of the domain-specific features and the grey value (Figure 2 (c)). Increasing the sample size S further does not increase the number of actually selected features and keeps the quality of the output stable.

6 Discussion

We discussed the problem of supervised segmentation in the presence of a large number of features (such as filter outputs, gery values, and outputs of task-specific operators). We argued that, when the number of available features is reasonably large, it is impossible to use a learning algorithm without prior selection of a good

model. We further argued that the selection of a model can in fact be performed by a model selection algorithm, such as wrappers for feature subset selection. Using model selection algorithms is unlikely to improve the results if the available feature set is very small and already a good approximation of the optimal model. But it will yield a high gain if the available feature set consists of a large library of filters, as is often the case in pattern recognition tasks. Unfortunately, model selection incurs a higher-level over-fitting of the data, similar to the over-fitting caused by too large hypothesis spaces (Ng, 1997). If the number of models that are compared by cross validation is large, chances are that one model matches the hold-out sets particularly well without actually being a good model. Therefore, an engineer who guesses the best model in the first place will gain a lower expected error than an algorithm that assesses a large number of models – even if the optimal model is in the set of assessed models (because the variance in model assessment incurs the risk of selecting a sub-optimal one). One way to overcome this problem is to *learn* an appropriate model selection bias for a class of similar problems (Baxter, 1997).

While the Gabor filters which we use as attributes are fixed and the task is to select a subset of them to form a well discriminating hypothesis, Weldon and Higgins (1997) propose a method for optimizing Gabor filters for a specific classification task. Their algorithm selects filters that maximize the ratio of amplitudes of different segments. Greenspan (1996) uses a Naive Bayes classifier in order to learn a hypothesis in terms of the output of a fixed set of Gabor filters. This paper extends Greenspan's work by introducing a model selection step that makes learning from a large set of features possible.

ACKNOWLEDGMENT

We wish to thank Lucas Parra who provided the x-ray software and supported us on using it. Additionally, we thank Hauke Bartsch and Christian Piepenbrock for the essential insights they gave us into the field of Gabor functions. The second author is supported by an Ernst-von-Siemens fellowship.

References

Baxter, J. (1997). A model for bias learning. *Journal of the ACM*. submitted.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. (1987). Occam's razor. *Information processing Letters, 24*, 377–380.

Bovik, A. C., Clark, M., & Geisler, W. S. (1990). Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(1), 55–73.

Brodatz, P. (1966). *Textures: A Photographic Album for Artists and Designers*. Dover.

Cross, G. R., & Jain, A. K. (1983). Markov random field texture models. *IEEE Pattern Analysis and Machine Intelligence, 5*(1), 25–39.

Greenspan, H. (1996). *Non-Parametric Texture Learning*, pp. 1–31. Oxford University Press.

Houzelle, S., & Giraudon, G. (1992). Model based region segmentation using cooccurrence matrices. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 636–639.

Kearns, M., Mansour, Y., Ng, A. Y., & Ron, D. (1997). An experimental and theoretical comparsion of model selection methods. *Machine Learning, 27*, 7–50.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Journal of AI, Special Issue on Relevance*. to appear.

Ng, A. (1997). Preventing overfitting of cross validation data. In *Proceedings of the International Conference on Machine Learning*, pp. 245–253. Morgan Kaufmann.

Rissanen, J. (1985). Minimum–description–length principle. *Annuals of the Statistics, 6*, 461–464.

Scheffer, T. (1996). Algebraic foundation and improved methods of induction of ripple down rules. In *Proccedings of the Pacific Rim Workshop on Knowledge Acquisition*, pp. 279–292.

Vapnik, V. (1982). *Estimation of Dependences Based on Empirical Data*. Springer–Verlag, New York.

Weldon, T. P., & Higgins, W. E. (1997). Algorithm for designing multiple gabor filters for segmenting multi-textured images. unpublished.