Adaptive Sequential Feature Selection for Pattern Classification

Liliya Avdiyenko[1], Nils Bertschinger[1] and Juergen Jost[1,2]
[1]*Max Planck Institute for Mathematics in the Sciences, Inselstr. 22, 04103 Leipzig, Germany*
[2]*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, New Mexico 87501, U.S.A.*

Keywords: Adaptivity, Feature Selection, Mutual Information, Multivariate Density Estimation, Pattern Recognition.

Abstract: Feature selection helps to focus resources on relevant dimensions of input data. Usually, reducing the input dimensionality to the most informative features also simplifies subsequent tasks, such as classification. This is, for instance, important for systems operating in online mode under time constraints. However, when the training data is of limited size, it becomes difficult to define a single small subset of features sufficient for classification of all data samples. In such situations, one should select features in an adaptive manner, i.e. use different feature subsets for every testing sample. Here, we propose a sequential adaptive algorithm that for a given testing sample selects features maximizing the expected information about its class. We provide experimental evidence that especially for small data sets our algorithm outperforms two the most similar information-based static and adaptive feature selectors.

1 INTRODUCTION

Machine learning is often confronted with high-dimensional data. A common problem is the so-called "curse of dimensionality", meaning that the amount of data required to find good model parameters grows exponentially with the dimension of the input space. For this reason, as well as computational issues, feature selection is often used to reduce the data dimensionality to the features relevant to solve a given problem, such as classification. Moreover, in a situation when the training set is of a limited size, a classifier built on a smaller number of features usually has better generalization ability.

Basically, one can distinguish between two types of feature selection algorithms: filters and wrappers (Webb, 1999). The former try to reduce the dimensionality of the data while keeping potential clusters in the data well separated. In this case, the relevance of each feature is evaluated using different measures of distances between classes, e.g. probabilistic distance measures. However, the involved probabilities are difficult to estimate and often approximate methods are used. Wrappers also preprocess the data but directly take into account that the resulting features should be useful for a certain classifier. Therefore, features are selected based on the prediction accuracy of the classifier employing these features. This might lead to better results but is usually computationally demanding and prone to overfitting.

In each case, one can look for the best feature subset of a certain cardinality using an optimal search strategy. Since the number of possible subsets is exponentially large, testing all of them is infeasible. A good example is the branch and bound method (Narendra and Fukunaga, 1977) that assumes monotonicity of the selection criterion to avoid an exhaustive search. If such an assumption is not valid and the number of features is large, suboptimal methods have to be used. This class of algorithms includes forward and backward sequential feature selection, e.g. (Ding and Peng, 2005; Abe, 2005). In both cases, the relevance of each feature is evaluated together with the current feature subset.

Among probabilistic criteria used by filters, selection criteria based on Shannon entropy are widely used (Duch et al., 2004). Such criteria select the features to reduce uncertainty about the output class. Battiti was one of the first to use mutual information, a concept closely related to the Shannon entropy, for sequential feature selection (Battiti, 1994). However, this involves estimation of the conditional mutual information (CMI), i.e. the amount of information between the feature and the class given the already selected features, which requires multivariate density estimation. To circumvent this problem, Battiti approximated CMI by pairwise mutual information. Kernel density estimation (discussed below

Avdiyenko L., Bertschinger N. and Jost J..
Adaptive Sequential Feature Selection for Pattern Classification.
DOI: 10.5220/0004146804740482
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (NCTA-2012), pages 474-482
ISBN: 978-989-8565-33-4

in subsection 2.2.1) is a non-parametric technique widely used for multivariate density estimation. It was successfully applied to estimate CMI for the exhaustive search procedure (Bonnlander and Weigend, 1994) and forward feature selection (Kwak and Choi, 2002; Bonnlander, 1996).

Ideally, it should be possible to describe all observations by the same small subset of features. However, when the amount of available training data is limited and the number of features exceeds the number of training samples, it is very likely that no single feature subset is good enough for classification of all observations. For example, one may need different features to discriminate between classes, or even different objects belonging to one class may have different discriminative features. One can partially overcome this problem by having a collection of all relevant feature subsets. This, however, will lead to an increase in the classifier complexity, which in turn will lead to its poor performance, since there is not enough data for training the classifier in high-dimensional space, e.g. see (Raudys and Jain, 1991). Thus, conventional feature selection schemes, which select a fixed subset of features before they are handed to a classifier, can be inefficient.

Thereby, in cases of small data sets, we propose to use different subsets of features for every testing sample, i.e. select the relevant features in an "adaptive" manner. Here, by adaptivity we mean that for a certain testing sample every selected feature should yield the maximum additional information about the class given the already selected features with values observed on this testing sample.

The idea of adaptivity was used by Geman and Jedynak in their active testing model (1996) where they sequentially select tests in order to reduce uncertainty about the true hypothesis. For their problem domain, they assumed that features are conditionally independent given the class. Jiang also used an adaptive scheme (Jiang, 2008), however, without conditioning on the already selected features, which are employed only to update a set of currently active classes. In contrast to these schemes, we adaptively select features taking into account high-order dependencies between them.

Here, we propose an adaptive feature selection algorithm based on CMI that sequentially adds features one by one to a subset of features relevant for a certain testing sample. Even though the multivariate probability densities are hard to estimate in general and from small data sets especially, the algorithm is still able to select informative features in high dimensions.

Our model is also inspired by sequential visual processing, i.e. a principle of eye movements when performing a task. Since a human can foveate only on a small part of an image at time, the scene is perceived sequentially. Moreover, only a few eye fixations are usually enough to analyze the whole scene. This might suggest that optimal saccades for a certain task follow the sequence of the most informative scene-specific locations. For experimental support see (Renninger et al., 2007; Najemnik and Geisler, 2005).

Sec. 2.1 explains the mathematical basis and general idea of our method, whereas Sec. 2.2 gives implementation details. Then, in Sec. 3, we provide results for two image classification tasks using artificially constructed bitmap images of digits and real-world data from the MNIST database of hand-written digits. We show that our method outperforms the Parzen window feature selector (Kwak and Choi, 2002) and the active testing model (Geman and Jedynak, 1996), which are static and adaptive CMI-based feature selectors. Finally, in Sec. 4 we discuss benefits of our approach and future extensions.

2 MODEL

For our model, we start with a standard classification setup. Suppose we have a space of possible inputs $\mathcal{F} = \times_{i=1}^{n} \mathcal{F}_i$, i.e each input is an n-dimensional feature vector $\mathbf{f} = (f_1, \ldots, f_n)$, where the i^{th} feature takes values $f_i \in \mathcal{F}_i$. Our notion of feature is rather general, ranging from simple ones, such as the gray-value of a certain pixel, to sophisticated ones, such as counting faces in an image. Feature combinations are considered as a random variable F with a joint distribution on $\mathcal{F}_1 \times \cdots \times \mathcal{F}_n$ and the observation $\mathbf{f}$ is drawn from that distribution.

Furthermore, each observation has an associated class label $c \in \mathcal{C} = \{c_1, \ldots, c_m\}$. The task of the classifier is to assign a class label to each observation $\mathbf{f}$. Thus, formally it is considered as a map $\phi : \mathcal{F} \to \mathcal{C}$ or, more generally, assigning to each $\mathbf{f}$ the conditional probabilities $p(c|\mathbf{f})$ of the classes c. To learn such a classification, we are given a training set $\mathcal{X} = \{(\mathbf{x}_i, c_i)\}_{i=1}^{T}$ of labeled observations, which are assumed to be drawn independently from the distribution relating feature vectors and class labels. Then the goal is to find a classification rule ϕ that correctly predicts the class of future samples with unknown class label, called testing samples. That is, confronted with a feature vector ξ we would classify it as $c = \phi(\xi)$. Feature selection then means that for this particular task only a subset of features rather than the full feature vector is used.

2.1 Adaptive Feature Selection

Adaptivity: For classification problems with small training sets, we suggest to select features adaptively. Thus, we do not predefine a single subset of the relevant features but rather select a specific one for every new testing sample. The proposed feature selection scheme is a sequential feedforward algorithm. Every next feature added to the subset should be discriminative together with the already selected features, which take particular values observed on the current testing sample.

Sequential feedforward feature selection algorithms use a greedy search strategy, which does not assume the full search and evaluating the relevance of every possible feature subset. Such feedforward algorithms start from the empty set and add features one by one so that every next feature maximizes some selection criterion S considering features selected on the previous steps. Thus, conventionally the feature $F_{\alpha_{i+1}}$ selected on the $(i+1)^{th}$ step should satisfy the following:

$$\begin{aligned} \alpha_{i+1} &= \arg\max_k S(F_{\alpha_1},\ldots,F_{\alpha_i},F_k), \\ F_k &\in \{F_1,\ldots,F_n\} \setminus \{F_{\alpha_1},...,F_{\alpha_i}\}, \end{aligned} \tag{1}$$

where $F_{\alpha_1},\ldots,F_{\alpha_i}$ is a subset of the features selected before the $(i+1)^{th}$ iteration.

Let us consider an adaptive case. Suppose that we have a testing sample ξ. Suppose also that after i steps we have selected the features $F_{\alpha_1},\ldots,F_{\alpha_i}$ and observed their values $\xi_{\alpha_1},\ldots,\xi_{\alpha_i}$ on this testing sample. Then, for this testing sample the next feature $F_{\alpha_{i+1}}$ is selected according to the adaptive criterion:

$$\alpha_{i+1} = \arg\max_k S(F_{\alpha_1}=\xi_{\alpha_1},\ldots,F_{\alpha_i}=\xi_{\alpha_i},F_k). \tag{2}$$

In contrast to the static criterion (1), the adaptive criterion also takes into account the *values* of the already selected features, *which are observed on the current testing sample*.

Probabilistic Selection Criterion: The feature selection scheme proposed here uses a probabilistic selection criterion and is based on the mutual information between the features and class variables (Cover and Thomas, 1991).

The mutual information between two continuous random variables A and B measures the amount of information between them and is defined as follows:

$$I(A;B) = \int_A \int_B p(a,b)\log\frac{p(a,b)}{p_A(a)p_B(b)}\mathrm{d}b\mathrm{d}a, \tag{3}$$

where $p(a,b)$ is the joint probability density function (pdf) of A and B, and $p_A(a) = \int_B p(a,b)\mathrm{d}b$ and $p_B(b) = \int_A p(a,b)\mathrm{d}a$ are their marginal densities. In case of discrete variables, the integration is substituted by summation over the values of the variables.

Our goal is a sequential selection of features that bring the maximum additional information about classes, i.e. those that are both discriminative and non-redundant with respect to the already selected features. Thus, we propose the adaptive mutual information feature selector (AMIFS), which is based on the expected mutual information between the classes and a feature candidate k conditioned on the outcome of the selected features which is observed on the testing sample $I(C;F_k|\xi^i)$. Then, according to AMIFS every next selected feature should satisfy the following:

$$\alpha_{i+1} = \arg\max_k S(F_{\alpha_1}=\xi_1,\ldots,F_{\alpha_i}=\xi_{\alpha_i},F_k) = \arg\max_k \left\{ \int_{\mathcal{F}_k} \sum_{c\in C} p(f_k,c|\xi^i)\log\frac{p(f_k,c|\xi^i)}{p(f_k|\xi^i)p(c|\xi^i)}\mathrm{d}f_k \right\}, \tag{4}$$

where the variable C represents the classes, $C = \{c_1,\ldots,c_m\}$, and $\xi^i = \{F_{\alpha_1}=\xi_{\alpha_1},\ldots,F_{\alpha_i}=\xi_{\alpha_i}\}$ is a shorthand for the set of values which are observed on the selected features of the sample ξ.

Note that the expression (4) is not a conventional CMI since we do not average over all possible outcomes of the features $F_{\alpha_1},\ldots,F_{\alpha_i}$, but rather condition on the specific values that we observe on the particular testing sample. This implies that we look for the feature $F_{\alpha_{i+1}}$ that is informative for the certain region of the input space, which is specified by the observed values of the already selected features. Thefore, we adaptively select a different subset of the relevant features for every sample we want to classify.

Using the definition of the Kullback-Leibler divergence,

$$D(p||q) = \int p(x)\log\frac{p(x)}{q(x)}\mathrm{d}x \tag{5}$$

for two distributions p and q, (4) can be rewritten as follows:

$$\alpha_{i+1} = \arg\max_k \left\{ \sum_{c\in C} p(c|\xi^i) D(p(f_k|c,\xi^i)||p(f_k|\xi^i)) \right\} \tag{6}$$

This is the average distance between the pdf of the feature F_k given a certain class and its marginal pdf, where both pdfs are updated after observing the current feature subset on the sample ξ. Thus, the selection criterion favors features with distinctive posterior distributions for data drawn from the different classes, that is, features that on the $(i+1)^{th}$ step are expected to best discriminate between the classes.

In our algorithm, the first feature is selected independently of the testing sample ξ and should maximize the mutual information with classes:

$$\alpha_1 = \arg\max_k I(C;F_k), \quad F_k \in \{F_1,\ldots,F_n\}. \tag{7}$$

The scheme becomes adaptive only after the first feature is selected and the value it takes on the testing sample is known.

Stopping Rule: Ideally, the algorithm can be stopped when one of the classes has been unambiguously identified. In practice, this is not possible and other stopping criteria have to be used, e.g. minimum additional information that the next feature brings or simply a maximum number of iterations. However, in this paper, we shall not address the issue of stopping rules.

2.2 Estimation of the Selection Criterion

The selection criterion (4) can be rewritten as

$$\alpha_{i+1} = \arg\max_k \left\{ \sum_{j=1}^{m} p(c_j|\xi^i) \times \int p(f_k|\xi^i,c_j) \log \frac{p(f_k,\xi^i|c_j)p(c_j)p(\xi^i)}{p(c_j,\xi^i)p(f_k,\xi^i)} \mathrm{d}f_k \right\}. \tag{8}$$

The pdfs under the logarithm, that do not depend on f_k and therefore do not contribute to $\arg\max_k$, can be dropped. Thus, we obtain

$$\alpha_{i+1} = \arg\max_k \left\{ \sum_{j=1}^{m} p(c_j|\xi^i) E_{p(f_k|\xi^i,c_j)}[\log \frac{p(f_k,\xi^i|c_j)}{p(f_k,\xi^i)}] \right\}. \tag{9}$$

The expression (9) requires estimation of multivariate pdfs as well as the conditional expectation over multivariate pdf.

2.2.1 Kernel Density Method

In our case, we solve both problems with the kernel method, a nonparametric smoothing technique developed by Rosenblatt (Rosenblatt, 1956) and Parzen (Parzen, 1962).

Density Estimation: For a training set consisting of T independently and identically distributed (iid) n-dimensional samples $\mathcal{X} = \{\mathbf{x}_1,...,\mathbf{x}_T\}, \mathbf{x}_i \in \mathbb{R}^n$, the kernel density estimate (KDE) of the pdf $\hat{p}(\mathbf{y})$ is

$$\hat{p}(\mathbf{y}) = (T\prod_{j=1}^{n} h_j)^{-1} \sum_{x_i \in \mathcal{X}} \prod_{j=1}^{n} K(\frac{y_j - x_{i,j}}{h_j}), \tag{10}$$

where $K(\cdot)$ is a univariate kernel function, h_j is a kernel bandwidth parameter and $x_{i,j}$ is the value of the j^{th} feature of the sample $\mathbf{x_i}$. Here, we use a so-called product kernel, which is a commonly used simplification of the general multivariate kernel. Since quality of the density estimation does not particularly depend on the choice of the kernel, for convenience we restrict ourselves to Gaussian kernels $K(w) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{w^2}{2})$.

Bandwidth selection: The bandwidth parameters h_j control the smoothness of the estimated density. Setting them too large, all details of the density structure are lost, whereas setting them too small will lead to a highly variable estimate with many false peaks around every sample point. Therefore, a choice of the proper bandwidth parameters is important. We only briefly mention the bandwidth selection method that we used, for details and an overview of other methods see (Turlach, 1993).

The normal reference rule (Silverman, 1986) is one of the simplest methods based on the asymptotic mean integrated squared error between the true and estimated densities and assumes that the data is Gaussian. The method produces good estimates for univariate densities but tends to oversmooth for multivariate cases. Among more sophisticated methods that can be easily extended to the multivariate densities are Markov chain Monte Carlo (MCMC) methods. They estimate a bandwidth matrix through the data likelihood using cross-validation and are reported to have a good performance, e.g. see (Zhang et al., 2004).

In higher dimensions data become sparser and tend to move away from the modes of the distribution (Scott, 1992). Therefore, the bandwidth parameter of kernel functions should be adjusted to the data dimensionality so that estimates are based on a sufficient number of data points. In our case, the dimension of estimated densities grows iteratively. Moreover, we estimate joint densities of different feature subsets. Ideally, one has to select a unique optimal bandwidth vector for every feature combination of different cardinality. Since it is computationally infeasible, we pick the normal reference rule, which does not require any optimization and automatically gives a bandwidth depending on the dimension of the estimated density. So the bandwidth for feature F_i is defined as

$$h_i = (\frac{4}{d+2})^{\frac{1}{d+4}} \sigma_i T^{-\frac{1}{d+4}}, \tag{11}$$

where d is the dimension of the estimated multivariate density, σ_i is the standard deviation of the data points and T is the number of training samples.

2.2.2 Conditional Expectation

We estimate the conditional expectation over the multivariate pdf $p(f_k|\boldsymbol{\xi}^{\mathbf{i}},c_j)$ using a kernel-based estimator as well. Let us consider a training set $\mathcal{X} = \{(\mathbf{x}_1,\mathbf{y}_1),...,(\mathbf{x}_T,\mathbf{y}_T)\}$, where $\mathbf{x}_i$ and $\mathbf{y}_i$ are realizations of n_x- and n_y-dimensional continuous random variables $\mathbf{x}$ and $\mathbf{y}$, respectively. Suppose, one needs to estimate the expectation of some function $g(\mathbf{x})$ over the conditional distribution $p(\mathbf{x}|\mathbf{y}=\mathbf{a})$, where $\mathbf{a}$ is a particular observation of the variable $\mathbf{y}$. Then, using the nonparametric kernel regression estimator proposed by Nadaraya (1964) and Watson (1964), the conditional expectation of $g(\mathbf{x})$ is:

$$E_{p(\mathbf{x}|\mathbf{y}=\mathbf{a})}[g(\mathbf{x})] = \frac{(T\prod_{j=1}^{n_y} h_j)^{-1}\sum_{x_i\in\mathcal{X}}\prod_{j=1}^{n_y} K_j(a,y_i)g(\mathbf{x})}{(T\prod_{j=1}^{n_y} h_j)^{-1}\sum_{x_i\in\mathcal{X}}\prod_{j=1}^{n_y} K_j(a,y_i)}, \tag{12}$$

where $(h_1,...,h_{n_y})$ is a bandwidth vector of the kernel for the variable $\mathbf{y}$ and $K_j(a,y_i)$ denotes $K(\frac{a_j-y_{i,j}}{h_j})$. Note that the denominator is KDE of $\hat{p}(\mathbf{y}=\mathbf{a})$.

Plugging (12) into the selection criterion (9), we have:

$$\alpha_{i+1} = \arg\max_k \left\{ \sum_{j=1}^{m} \frac{p(c_j|\xi^i)}{p(\xi^i|c_j)} (T_j \prod_{q=1}^{i} h_{\alpha_q})^{-1} \times \sum_{x_r\in\mathcal{X}_j} \prod_{q=1}^{i} K_{\alpha_q}(\xi,x_r) \log \frac{p(f_k=x_{r,k},\xi^i|c_j)}{p(f_k=x_{r,k},\xi^i)} \right\}, \tag{13}$$

where $\mathcal{X}_j$ is a subset of the training samples belonging to the class c_j and $T_j = |\mathcal{X}_j|$.

Note that the expression in the first fraction simplifies just to $p(c_j)$, because $\frac{p(c_j|\xi^i)}{p(\xi^i|c_j)} = \frac{p(c_j)}{p(\xi^i)}$ and $p(\xi^i)$ can be dropped as it does not influence $\arg\max_k$. Finally, using the kernel method to estimate densities and after some simple algebraic transformations, the expression (13) is of the form:

$$\alpha_{i+1} = \arg\max_k \left\{ \sum_{j=1}^{m} p(c_j) T_j^{-1} \times \sum_{x_r\in\mathcal{X}_j} \prod_{q=1}^{i} K_{\alpha_q}(\xi,x_r) \log \frac{\sum_{x_s\in\mathcal{X}_j} K_k(x_r,x_s) \prod_{q=1}^{i} K_{\alpha_q}(\xi,x_s)}{\sum_{x_u\in\mathcal{X}} K_k(x_r,x_u) \prod_{q=1}^{i} K_{\alpha_q}(\xi,x_u)} \right\}. \tag{14}$$

The expression under the logarithm measures a ratio between values of two pdfs in the point x_r. When the pdfs are estimated from small training sets, unreliabilities can lead to large ratios even though there is no real evidence for that. To cope with this, we add a small value to both pdfs which can be interpreted as smoothing them with an improper base distribution. The smoothing should be adjusted to the current dimension of the pdf, thus, we take it to be proportional to the maximum response of the product kernel of the selected features over all training points. Such a simple smoothing works fine for our problem, since we do not need precise values of the criterion in (14), but rather want to find a feature that maximizes it.

3 EXPERIMENTS

Here, we provide an experimental comparison of our method with two feature selection algorithms based on CMI: Parzen window feature selector (PWFS) (Kwak and Choi, 2002) and active testing model (ATM) (Geman and Jedynak, 1996). In our terminology PWFS is a static selection scheme (1). It is based on the conventional CMI estimated with the kernel method. ATM is a feature selector based on the adaptive CMI which uses a simplifying assumption that features are conditionally independent given a class. Since the estimation of the selection criterion, proposed by Geman and Jedynak, was problem-specific, here we use just the general idea of their method. That is, in our experiments ATM selects features as follows:

$$\alpha_{i+1} = \arg\max_k \left\{ \sum_{j=1}^{m} \frac{p(c_j)\prod_{q=1}^{i} p(f_{\alpha_q}=\xi_{\alpha_q}|c_j)}{T_j} \times \sum_{x_r\in\mathcal{X}_j} \log \frac{p(f_k=x_{r,k}|c_j)}{\sum_{v=1}^{m} p(c_v)p(f_k=x_{r,k}|c_v)\prod_{q=1}^{i} p(f_{\alpha_q}=\xi_{\alpha_q}|c_v)} \right\}. \tag{15}$$

To make a fair comparison, all criteria are estimated using KDE with the same bandwidth vector as chosen by the normal reference rule (11).

3.1 Artificial Data Set

For the first experiment, we artificially constructed a data set for image classification. It contains pixel-based black-and-white images of digits belonging to 10 different classes. First, we constructed four distinct examples of every class. From this data set

we generated a new one with 1000 samples by randomly adding 5 pixels of noise to the original images (Fig. 1). Further, we formed 20 training sets with 30 and 300 samples in each and one testing set containing 100 samples by randomly selecting an equal number of samples from each class.

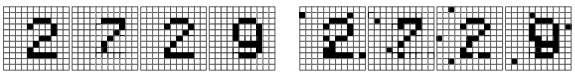

Figure 1: Examples of original and noisy digits.

In our setup, each image is described by a vector of complex features. These, in turn, are functions of simple features of the image. Our simple features are inspired by the complex cells in the primary visual cortex discovered by D. Hubel and T. Wiesel in the 1960s (Hubel and Wiesel, 2005). Both are responsive to primitive stimuli that are independent of their spatial location. Here, each simple feature corresponds to a 3×3 image patch and is activated proportional to the frequency with which the corresponding patch occurs in the image. For normalization and smoothing purposes, patch frequencies are squashed in the interval $[-1, 1]$ via a sigmoidal function.

The complex features correspond to 3×3 image patches as well. Their activation value is computed as a weighted sum of the activations of the simple features. The weight from the simple feature responding to the same patch is 1. For the others, it drops in the number of pixels that differ between the corresponding image patches according to a Gaussian. Thus, the complex features react more robust against pixel noise than the simple features. Since there are 9 binary pixels in each 3×3 patch, an image is described by a vector of $2^9 = 512$ complex feature values.

As a classifier we used the weighted k-nearest neighbor algorithm (wk-NN). It assigns a class to a testing sample based on a distance-weighted vote of the k nearest training samples. The wk-NN is one of the simplest classifiers, but the fact that it does not need learning is useful because the adaptive scheme requires multiple running of the classifier with different features. Here, we used k=20 hand-tuned using validation sets.

To investigate the usefulness of the proposed AMIFS we ran experiments on training sets with T= 30 and 300 samples. Note, that all sets have fewer training samples than features which easily leads to overfitting. The classification errors were evaluated on separate testing samples and compared with the cases when feature selection was done using PWFS, ATM and when the classifier was run on the full feature vector, i.e. without feature selection (Fig. 2). All results are averaged over 20 runs with the different training sets.

One clearly sees the advantage of using an adaptive scheme for feature selection. Not only does the error rate drop very quickly with an increasing number of features, it goes even below the error that the classifier achieves when using all available features. In all our simulations, this effect never occurred for the static scheme PWFS and was particularly pronounced when using an extremely small number of training samples ($T = 30$), i.e. when the classifier is prone to overfitting. Furthermore, our algorithm outperforms the ATM scheme which assumes conditional independence of the features. Thus, especially at the beginning, i.e. when selecting the first few features, it is beneficial to take dependencies between features into account.

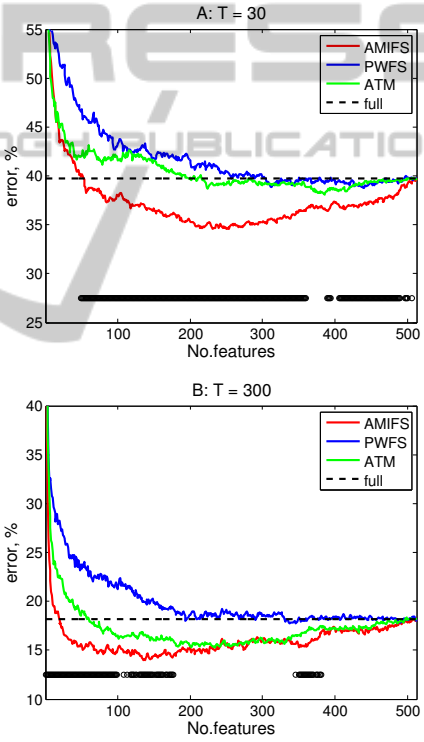

Figure 2: Error against the number of features for digits classification, the black markers indicate regions where AMIFS is significantly better than ATM according to the Wilcoxon signed-rank test at the p-level= 0.05.

Further, we test the ability of the considered schemes to select informative features in high dimensions for the case $T = 300$. For this, we start with initial feature subsets of size 50 and 100, which are preselected by PWFS, and then select further features according to the different algorithms. The results (Fig. 3) show that both adaptive schemes find additional features that are markedly better than the

statically selected ones. Also one can see that at some point ATM, the adaptive scheme assuming conditional independence of the features given a class, starts outperforming AMIFS. This fact suggests that after certain dimension AMIFS is not able anymore to estimate correctly high-order dependencies between the features. Interestingly, when AMIFS selects the features from the beginning (see Fig. 2), it performs better than ATM almost up to 200 features, meaning that the first good features can compensate for unreliable pdf estimates further in higher dimensions. Based on this observation, one could think of a combined scheme that starts with AMIFS and after selecting some features switches to ATM.

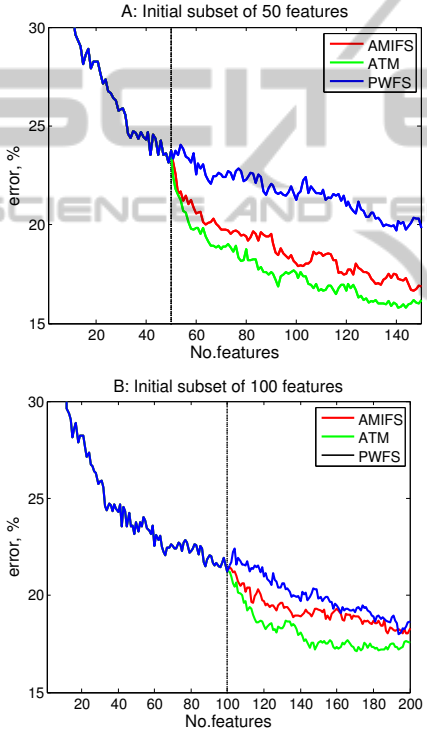

Figure 3: Comparison of ability to add informative features to subsets of 50 and 100 features preselected by PWFS.

3.2 MNIST Data Set

We compared performance of PWFS, ATM and our AMIFS on a real-world data set, the MNIST database of handwritten digits (LeCun and Cortes, nd). The images are 28×28 pixel, black and white, size-normalized and centered. The original training and testing sets consist of $60,000$ and $10,000$ samples, respectively.

The features were learned by LeNetConvPool (Bergstra et al., nd), a convolutional neural network based on the LeNet5 architecture, which was originally proposed by LeCun (LeCun et al., 1998). The convolutional networks are biologically inspired multilayered neural networks. In order to achieve some degree of location, scale and distortion invariance, they imitate arrangement and properties of simple and complex cells in primary visual cortex by implementing local filters of increasing size, shared weights and spatial subsampling.

LeNetConvPool consists of 6 layers: 4 successive convolutional and down-sampling layers (C- and S-layers), a hidden fully-connected layer and a logistic regressor as a classifier. C-layers consist of several feature maps with overlapping 5×5 linear filters. So every filter receives an input from the 5×5 region of the previous layer, computes its weighted sum and passes it through a sigmoidal function. The S-layers perform max-pooling with 2×2 non-overlapping filters. That is, an output of such filter is the maximum activation of units from 2×2 region of the corresponding feature map in the previous C-layer. For both types of the layers, all filters share the same weight parameters within one feature map. First C- and S-layers have 20 feature maps, the next ones - 50. The succeeding hidden layer, which is fully-connected to all units of all feature maps in the previous S-layer, has 500 units with the sigmoidal activation function. The last classification layer consists of 10 units, according to the number of classes, and performs a logistic regression. The weight parameters of all layers are learned using the gradient descent. For all implementation details see (Bergstra et al., nd).

We trained LeNetConvPool on 15 training sets with $5,000$ samples each. After that, the last classification layer was removed and the resulting networks with 500 output units were used as feature extractors. These units are initial features for the feature selectors. Then, from every training set we formed 2 sets of different size, with $T = 100$ and $T = 300$ samples, which were used for feature selection and for classification. We use different amount of training data for feature extraction and for further feature selection and classification to model a situation, when one has good features but there is not enough training data to build an efficient classifier. As a classifier, we used an unweighted k-NN with $k = 5$ (again, hand-tuned on validation sets), which in contrast to wk-NN uses a simple majority vote. For computational reasons, the testing set was reduced to 500 samples, which were randomly selected from the original MNIST testing set, with an equal number of samples per class.

Overall, all algorithms show a similar behavior as on the artificial data set (see Fig. 4). The smaller differences can be attributed to the better available features, as reflected in the much lower error rates, which

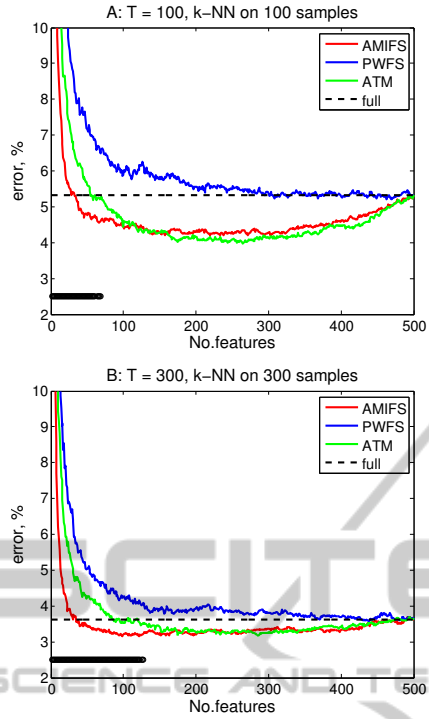

Figure 4: Error against the number of features for MNIST classification. k-NN was run on the same set as feature selection, the black markers indicate regions where AMIFS is significantly better than ATM according to the Wilcoxon signed-rank test at the p-level$= 0.05$.

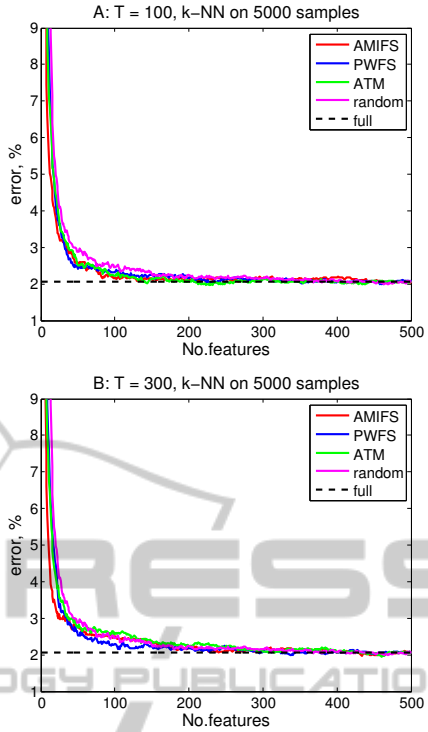

Figure 5: Error against the number of features for MNIST classification. k-NN was run on $5,000$ training samples, feature selection on A: 100 and B: 300 training samples.

have been tuned by the LeNetConvPool. Again, AMIFS outperforms ATM on the first selected features and both adaptive schemes provide some robustness against overfitting.

To see whether feature selection is as beneficial when the classifier is well-trained, we repeated the experiments with a training set of $5,000$ samples. However, as in the previous experiment, the feature selection was done on the small sets of 100 and 300 samples for computational reasons.

Fig. 5 shows that for this particular example one needs approximately 200 features to achieve the minimum error. However, there is no advantage of using any sophisticated feature selection algorithm, and one can see that a size of the training set used for selecting features does not have much influence as well. Moreover, even the random selection works about as good as other methods. We do not want to generalize results of this test by saying that for large data sets one can always select features randomly. We rather emphasize that for small data sets one can achieve better performance with features selected adaptively with our AMIFS .

4 DISCUSSION

Feature selection is a standard technique to reduce data dimensionality. In high-dimensional spaces this can be an efficient way to cope with limited amounts of training data. Usually, features are selected in a preprocessing step. However, we propose an adaptive scheme for feature selection, where each feature is selected as maximizing the expected mutual information with the class given the data point, as well as values of the features already considered.

Despite the fact that estimating the mutual information in high-dimensional spaces is a difficult problem on its own, we find that adaptive feature selection robustly improves the classification performance. In the considered examples, a small number of features is sufficient to achieve a good classification. Since the first few features can be reliably detected, our method does not overfit and can even compensate for shortcomings of the classifier. Our results on both artificial and real-world data show that in case of limited training data, when a classifier is usually prone to overfitting, AMIFS can even improve the error rate compared to using all available features.

Even though the algorithm is less advantageous

on large datasets, we believe that this is not a shortcoming, but merely shows that the need to select features is less pressing if enough data are available. From the point of view of computational expenses, in order to make AMIFS more applicable to large amount of data, one has to think about an approximate implementation which can cut down the computational complexity or, for example, consider using a hybrid scheme, i.e. starting with AMIFS and then after some iterations switching to ATM, which does not require estimating multivariate densities and therefore is computationally cheaper.

In the future, we want to develop a neural implementation of our feature selection scheme. The brain certainly faces a similar problem when it has to decide which features are really relevant to classify a new observation. A neural model could thus provide insights into how this ability can be achieved. Furthermore, we would like to investigate to what extent information theory provides guiding principles for information processing in the brain. In addition, adaptive feature selection could be accomplished via recurrent processing interleaving bottom-up and top-down processes.

REFERENCES

Abe, S. (2005). Modified backward feature selection by cross validation. In *Proc. of the Thirteenth European Symposium on Artificial Neural Networks*, pages 163–168, Bruges, Belgium.

Battiti, R. (1994). Using mutual information for selecting feature in supervised neural net learning. *IEEE Trans. Neural Networks*, 5(4):537–550.

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J., Warde-Farley, D., and Bengio, Y. (n.d.). Deep learning tutorials. Retrieved from: http://deeplearning.net/tutorial/lenet.html.

Bonnlander, B. V. (1996). *Nonparametric selection of input variables for connectionist learning*. PhD thesis, University of Colorado at Boulder.

Bonnlander, B. V. and Weigend, A. S. (1994). Selecting input variables using mutual information and nonparametric density estimation. In *International Symposium on Artificial Neural Networks*, pages 42–50, Taiwan.

Cover, T. M. and Thomas, J. A. (1991). *Elements of information theory*. Wiley-Interscience, New York, NY, USA. pp. 12-49.

Ding, C. H. Q. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *J. Bioinformatics and Computational Biology*, 3(2):185–206.

Duch, W., Wieczorek, T., Biesiada, J., and Blachnik, M. (2004). Comparison of feature ranking methods based on information entropy. In *Proc. of the IEEE International Joint Conference on Neural Networks*, pages 1415–1419, Budapest, Hungary.

Geman, D. and Jedynak, B. (1996). An active testing model for tracking roads in satellite images. *IEEE Trans. Pattern Analysis and Machine Intel*, 18(1):1–14.

Hubel, D. and Wiesel, T. (2005). *Brain and visual perception: the story of a 25-year collaboration*. Oxford University Press US. p. 106.

Jiang, H. (2008). *Adaptive feature selection in pattern recognition and ultra-wideband radar signal analysis*. PhD thesis, California Institute of Technology.

Kwak, N. and Choi, C. (2002). Input feature selection by mutual information based on parzen window. *IEEE Trans. Pattern Analysis and Machine Intel*, 24:1667–1671.

LeCun, J. and Cortes, C. (n.d.). The mnist dataset of handwritten digits. Retrieved from: http://yann.lecun.com/exdb/mnist/.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324.

Najemnik, J. and Geisler, W. S. (2005). Optimal eye movement strategies in visual search. *Nature*, 434:387–391.

Narendra, P. and Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Trans. Comput.*, 28(2):917–922.

Parzen, E. (1962). On estimation of a probability density and mode. *Annals of Mathematical Statistics*, 35:1065–1076.

Raudys, S. J. and Jain, A. K. (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13:252–264.

Renninger, L. W., Verghese, P., and Coughlan, J. (2007). Where to look next? Eye movements reduce local uncertainty. *Journal of Vision*, 7(3):117.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837.

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley. pp. 125-206.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. Chapman and Hall.

Turlach, B. A. (1993). Bandwidth selection in kernel density estimation: a review. In *CORE and Institut de Statistique*, pages 23–493.

Webb, A. (1999). *Statisctical Pattern Recognition*. Arnold, London. pp. 213-226.

Zhang, X., King, M. L., and Hyndman, R. J. (2004). Bandwidth selection for multivariate kernel density estimation using MCMC. Technical report, Monash University.