Methods for Updating of Model Parameters Applied within the Area of Symbol Recognition

M. Holden L. Eikvil G. Storvik

Norwegian Computing Center
P.O.Box 114, Blindern
N-0314 Oslo
Norway

Abstract

Use of statistical classification methods in symbol recognition requires the specification of a number of model parameters. Such parameters are usually obtained from a training set. This paper describes methods for improving parameter estimates obtained from a training set. The improvement is made by using new observations for updating the parameter estimates. The class membership of these new observations may either be known or unknown.

The potential of parameter updating techniques within the area of symbol recognition is discussed and tested on some examples. The results of using parameter updating for automatic training are promising and may lead to a considerable simplification of the training process. The problem of adapting the parameter estimates to a slightly changed situation, like for instance new fonts, did not lead to that good results. Nevertheless the results showed that also here there is a potential if the methods are further developed.

1 Introduction

In statistical classification we need a **class description base** (cdb) to describe the densities of the different classes. Assuming parametric distributions, the densities are described by a set of parameters. The parameters involved are usually estimated from a training set consisting of a number of elements from each class.

The parameters from the training process are only estimates. Sometimes they are not good enough for classifying new data correctly. There may be several reasons for that. A too small training set, for instance, may lead to high variability in the estimates. Another reason may be that the training set is chosen such that the training set for a certain class is too homogeneous to be representative. A third reason may be that the situation in which we want to do the classification has changed slightly from the training situation. By using information from new unclassified data we may update the parameter estimates in such a way that the new parameter estimates are more reliable. This updating of parameter estimates may also be used to simplify the time-consuming training process. This is possible by first training on a very small training set and then using parameter updating techniques on unclassified data to get more reliable estimates.

In Storvik et.al. [6] parameter updating techniques have been applied to MR-images of sections from the brain with good results. In Huseby et.al. [3] automatic training was used on bottle crate images from Tomra A/S with promising results. In this paper the possibility of using automatic training within the area of symbol recognition will be examined.

2 Parameter updating: Theory

Let us start by specifying the problem and introducing some notation.

Let x be a p-dimensional feature vector of interest. Assume x may be a member of one of a predefined set of classes, $1, \dots, K$ and the aim is to make a classification of x into one of the possible classes. The distribution of x is assumed to be $f_k(x;\theta)$ when x is a member of class k. In order to construct a classification rule, estimates of θ are needed. Typically, θ consists of a set of parameters $\{\theta_1, \dots, \theta_K\}$ where $f_k(x;\theta) = f_k(x;\theta_k)$. In this paper we will assume that f_k is the multinormal distribution for all k. In that case the parameters describing the distributions are the class-expectations and covariance matrices, that is $\theta_k = (\mu_k, \Sigma_k)$ where μ_k is the expectation vector and Σ_k is the covariance matrix.

In order to estimate θ, a training set $\{y_j^k, j = 1, \ldots, m_k, k = 1, \ldots, K\}$ is to our disposal. This training set consists of observations with known class-memberships where y_j^k is the jth observations from class k. The usual assumption is that this training set consists of samples from the same distributions as x, the observation to be classified. In that case, the training set may be used to find the maximum likelihood estimates for θ.

In some situations, the training set may consist of samples from distributions that are different from the ones for x. This may for instance be the case when only a training set from a different font-type is available. Similarities between the types of data may however give some *a priori* information about θ from the training set.

In addition to the training set, we will assume a new set of observations is available from the same population as x. These observations may either have known or unknown class-memberships. In the case of known class-memberships, this set will be denoted by $\{x_i^k, i = 1, \ldots, n_k, k = 1, \ldots, K\}$ while when the class-memberships are unknown, the set will be denoted by $\{x_i, i = 1, \ldots, n\}$.

In this paper we will study the estimation of θ under the two different situations for the training set. In the situation when both the x's and the y's are samples from the same population, **maximum likelihood** estimates may be derived using all the available data. In the other situation, when the y's are samples from a different population, a **Bayesian approach** will be used where the training set will be used to construct *a priori* distributions for the parameters θ. For simplicity, we will assume all the parameters to be independent in the prior distribution.

Complete data: Consider first the situation with complete data, that is the classes of the new data are known. This combined with maximum likelihood updating is nothing more than the traditional situation with an increasing training set. This situation will therefore not be considered in this paper. Consider next the Bayesian approach. Assuming a prior distribution on μ_k as the multinormal distribution with expectation η_k and covariance matrix A_k ($\mu_k \sim N(\eta_k, A_k)$), the following equation for the optimal estimate of μ_k may be derived[1]:

$$\mu_k = \{A_k^{-1} + n_k\Sigma_k^{-1}\}^{-1}\{A_k^{-1}\eta_k + n_k\Sigma_k^{-1}\hat{\mu}_k\}. \quad (1)$$

where $\hat{\mu}_k = \frac{1}{n_k}\sum_{i=1}^{n_k} x_i^k$.

For the covariance matrices, we will use the Wishart distribution (see Mardia et.al. [5]) as *a priori* distribution for the inverse of the covariance matrices, $\Sigma_k^{-1} \sim W(C_k, d)$. Here C_k is the the expectation of Σ_k^{-1}, and d is the degrees of freedom (for simplicity assumed equal for all classes). It can then be shown that the optimal estimate for Σ_k has to satisfy

$$\Sigma_k = \frac{(d-p-1)C_k^{-1} + n_k\widehat{\Sigma}_k}{(d-p-1)+n_k}. \quad (2)$$

where $\widehat{\Sigma}_k = \frac{\sum_{i=1}^{n_k}(x_i^k - \widehat{\mu}_k)(x_i^k-\widehat{\mu}_k)'}{n_k} + (\widehat{\mu}_k - \mu_k)(\widehat{\mu}_k - \mu_k)'$.

Note that the expression for μ_k depend on Σ_k and vice verse, so that the estimates can not be directly solved. Iteration between equations (1) and (2) should however result in the desired solution.

Incomplete data: Consider now the situation of incomplete new data, that is the classes of the new data are unknown. For this situation the estimation problem is more complicated because of the incompleteness of the observations. Typically, no explicit expressions for the estimates are possible to be obtained and numerical solutions are needed. The EM-algorithm (Dempster et.al. [1], Wu [7]) is an algorithm that is specifically constructed for estimation problems with incomplete data. The algorithm consists of two steps, an E-step and an M-step, which are performed in cycles. In the **E-step** the incomplete data is estimated using the current estimate of θ as the true parameter. In our case, the incomplete data will be the class-memberships, or actually the probabilities of the class-memberships. We will denote the probability of observation x_i belonging to class k given all the data by $p_i(k|x)$. These probabilities are found through $p_i(k|x) = \frac{\pi_k f_k(x_i;\theta_k)}{\sum_{m=1}^{K}\pi_m f_m(x_i;\theta_k)}$ under the assumption of all observations being independent, which will be the case in the situations considered in this paper. In the **M-step**, θ is estimated based on the new and old observations treating the estimates of the incomplete data as the true observations.

For the case when the x's and y's are samples from the same population (i.e. the maximum likelihood case), the expressions for the estimates of the expectations and covariance matrices are given by

$$\mu_k = \frac{m_k\overline{\mu}_k + \sum_{i=1}^n p_i(k|x)x_i}{m_k + \sum_{i=1}^n p_i(k|x)}. \quad (3)$$

[1] All formulae and equations in this section are derived in Storvik et.al. [6] or Eikvil et.al. [2].

$$\Sigma_k = \frac{m_k\overline{\Sigma}_k + m_k(\mu_k - \overline{\mu}_k)(\mu_k - \overline{\mu}_k)'}{m_k + \sum_{i=1}^n p_i(k|x)} + \frac{\sum_{i=1}^n p_i(k|x)(x_i - \mu_k)(x_i - \mu_k)'}{m_k + \sum_{i=1}^n p_i(k|x)}. \quad (4)$$

where $\overline{\mu}_k$ and $\overline{\Sigma}_k$ are the maximum likelihood estimates based on the training set alone.

Expressions 3 and 4 describe the M-step in the EM-algorithm for the maximum likelihood approach. As initial values of the parameters, the estimates based only on the training set may be used.

For the case when the y's are samples from another population than the x's (i.e. the Bayesian approach) we will assume the same *a priori* distributions on the parameters as in the case with complete data, that is $\mu_k \sim N(\eta_k, A_k)$ and $\Sigma_k^{-1} \sim W(C_k, d)$. In that case, the estimates are given by

$$\mu_k = \{A_k^{-1} + n\Sigma_k^{-1}\hat{\pi}_k\}^{-1}\{A_k^{-1}\eta_k + n\Sigma_k^{-1}\hat{\pi}_k\hat{\mu}_k\}. \quad (5)$$

$$\Sigma_k = \frac{(d-p-1)C_k^{-1} + n\hat{\pi}_k\widehat{\Sigma}_k}{(d-p-1) + n\hat{\pi}_k}. \quad (6)$$

where in this case $\widehat{\Sigma}_k = \frac{\sum_{i=1}^n (x_i - \widehat{\mu}_k)(x_i - \widehat{\mu}_k)' p_i(k|x)}{\sum_{i=1}^n p_i(k|x)} + (\widehat{\mu}_k - \mu_k)(\widehat{\mu}_k - \mu_k)'$ and $\hat{\mu}_k = \frac{\sum_i p_i(k|x) x_i}{\sum_i p_i(k|x)}$. Equations 5 and 6 describe the M-step in the EM-algorithm for the Bayesian approach. As initial values for the algorithm, the expectations in the *a priori* distributions may be used.

3 Updating in symbol recognition

We have tested the updating methods for some problems from symbol recognition. Here, as always for statistical classification, a good class-description-base is critical to get a good result. Creating such a class-description-base through training sets is a time-consuming process. Each symbol in the training set must be manually marked with the correct class. A simplification of this process would therefore be welcome.

A class-description-base must usually be adapted to the different types and fonts of symbols one wishes to recognize. This may be dependent on the set of features chosen, but font-dependent differences in the symbols can afflict the features considerably, making correct classification difficult without any font-type adaptation. This means that for each new class of documents it is usually necessary to construct a new training set. Even though the new documents do not introduce any new symbols, training can not be omitted if the symbols are of a different font-type. The training-process is often performed by the user. This procedure would be simplified if one could use a standard class-description-base which could be updated automatically when symbols of the new font-type are obtained, that is a training set gets available.

In the rest of this section we will examine the possibility of using parameter updating techniques for solving the problems sketched above.

3.1 Test data and feature vectors

We scanned four images containing symbols of four different fonts; <u>times</u>, <u>avant garde</u>, <u>bookman</u> and <u>zapf chancery</u>. Each image contained 13 different symbol classes with 30 symbols of each class, giving a total of 390 symbols per image. However, 30 symbols from each class would not be enough to make a separate training set and test-set with a reasonable number of symbols from each class. To obtain a larger set of symbols, we therefore generated slightly different images by setting different thresholds. For each font we produced 4 images by binarizing each image using four different thresholds. Inspection of the feature vectors for the symbols obtained by this procedure showed that the variability due to different thresholds was large enough to consider the 4 symbols obtained from one original symbol to be approximately independent samples.

Using the symbols from the lowest and highest thresholds for training and the symbols from the two middle thresholds for testing, gave us a training set and a test-set each containing 60 symbols from each class for each of the four fonts. The size of the symbols was about 35×40 pixels. None of the symbols in the sets were rotated. The classes of symbols represented in the set were : `b, c, d, e, f, h, m, n, r, s, t, x` and `z`. The four different fonts are shown below. The fonts are from top to bottom: times, avant garde, bookman and zapf chancery.

The classification was performed with the same set of features in all the examples presented below. The set used was based on the rotation variant elliptic Fourier descriptors of Kuhl & Giardina [4], which have previously given good results for symbol recognition. The feature-vector was 6-dimensional containing the following descriptors: $a_1^*, b_1^*, c_1^*, d_1^*, a_2^*, b_2^*$.

For the case with unknown classes of the new data the resulting feature vectors were used directly. For the case with known classes the vectors were added noise. Why this was done will be explained later.

3.2 Updating using Maximum likelihood and incomplete new data.

In this test we wanted to compare the results from the classification using a manually trained system to those of an automatically trained system. We started by constructing an initial class description from the training set by using only a few symbols from each class. Then automatic training was performed on a larger set of symbols of the same font. The resulting updated class descriptions were used for classification of another set containing symbols of the same font. The results from this classification were compared to that of performing classification based only on the initial class description and that of having manually trained on the same number of symbols as were used for the updating.

If the results using the automatically produced class description are comparable to those using the manually produced description, the automatic approach is preferable as it is less time-consuming.

The test was performed for all the four different fonts. We had one training set and a different test-set, achieved as described above, giving 60 symbols of each class in both training set and test set. The outlier probability for the classification was set to zero.

Using a training set of 60 symbols for each class, all the symbols in the test-set are correctly classified for all four fonts, showing that the symbols are easily separated inside each font-type. Reducing the training set to 8 symbols for each class, some of the symbols of the training set are classified to outliers. In table 1 the number of correct classified symbols are given for each symbol class along with the total classification rate.

Using the test-set to update the class-descriptions, these outliers disappear, giving 100% correct classification for the bookman and zapf chancery fonts, while a few errors appear for the avant garde and times fonts (see table 1).

3.3 Updating using Bayes and incomplete new data.

In this test we had a class description achieved from a set consisting of only one font, while we wanted to classify symbols of three other fonts. We wanted to see the effect of automatically updating the class description. The results were compared to those of performing the classification using only the initial class description, and using a description produced from all the fonts in the set.

The initial class description used for this test was constructed from a set of symbols from the times-font, 60 symbols of each class. The symbols we wanted to classify were of three other fonts: avant garde, bookman and zapf chancery. Of these three fonts, bookman was the one most similar to times, while zapf chancery was the one most different from this font. This is reflected in the results using only the initial class description for classification, as shown in table 2 below. In table 3 we give the classification results after use of automatic updating.

For all the three fonts the results of the classification improve after the updating. However, if all the symbols of one class are initially wrongly classified (like we see for the symbol `b`) the updating will not be able to correct this. In the cases where some symbols are initially correctly classified the updating has a very good effect (see f.ex. the symbols `t` and `x` of the bookman font). Still, the results cannot be compared to those resulting from using an initial class description based on symbols from the font in question.

3.4 Updating using Bayes and complete new data.

The setup for this test corresponds to the setup for the test based on incomplete new data. First an initial class description based on a basis font was used as before. Then for each of the three fonts the class descriptions were updated with varying numbers of symbols from each class. At last classification was performed on sets of symbols of the three different fonts, using the corresponding updated class description.

For the symbols we had in hand, the separation between the different classes were very good. In such cases use of a small training set typically will give better results than the more complicated updating schemes discussed here. In cases where the symbols are more "noisy", that is the separation between the classes are not too good, much larger training sets are needed. In such cases it may be preferable to utilize

Table 1: *Number of correct classified symbols for each symbol and total classification rate for three different schemes.*

Trained on a set of 60 symbols of each class.

font	b	c	d	e	f	h	m	n	r	s	t	x	z	Corr	Err	Out
avant garde	60	60	60	60	60	60	60	60	60	60	60	60	60	**100.0**	**0.0**	**0.0**
bookman	60	60	60	60	60	60	60	60	60	60	60	60	60	**100.0**	**0.0**	**0.0**
times	60	60	60	60	60	60	60	60	60	60	60	60	60	**100.0**	**0.0**	**0.0**
zapf chancery	60	60	60	60	60	60	60	60	60	60	60	60	60	**100.0**	**0.0**	**0.0**

Trained on a set of 8 symbols of each class.

font	b	c	d	e	f	h	m	n	r	s	t	x	z	Corr	Err	Out
avant garde	60	60	60	58	51	6	58	60	18	60	60	60	60	**86.0**	**0.0**	**14.0**
bookman	60	60	60	60	42	60	60	57	60	60	60	60	59	**97.2**	**0.0**	**2.8**
times	60	60	60	60	60	60	57	59	60	60	60	60	60	**99.5**	**0.0**	**0.5**
zapf chancery	60	59	44	60	60	60	36	60	60	60	60	42	60	**92.4**	**0.0**	**7.6**

Trained on a set of 8 symbols of each class and automatically updated.

font	b	c	d	e	f	h	m	n	r	s	t	x	z	Corr	Err	Out
avant garde	60	60	60	60	60	60	58	60	60	60	60	60	60	**99.7**	**0.3**	**0.0**
bookman	60	60	60	60	60	60	60	60	60	60	60	60	60	**100.0**	**0.0**	**0.0**
times	60	60	60	60	60	60	57	59	60	60	60	60	60	**99.5**	**0.5**	**0.0**
zapf chancery	60	60	60	60	60	60	60	60	60	60	60	60	60	**100.0**	**0.0**	**0.0**

Table 2: *Trained on a set of symbols of the times-font.*

Avant Garde

	b	c	d	e	f	h	m	n	r	s	t	x	z	Out	%Corr	%Err.
b	**0**	0	0	0	0	60	0	0	0	0	0	0	0	0	0.0	100.0
c	0	**60**	0	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
d	0	0	**0**	0	0	0	0	0	0	0	0	0	0	60	0.0	0.0
e	0	0	0	**0**	0	0	0	0	0	0	0	0	0	60	0.0	0.0
f	0	0	0	0	**0**	0	0	0	0	60	0	0	0	0	0.0	100.0
h	0	0	0	0	0	**0**	0	60	0	0	0	0	0	0	0.0	100.0
m	0	0	0	0	0	0	**60**	0	0	0	0	0	0	0	100.0	0.0
n	0	0	0	0	0	0	0	**60**	0	0	0	0	0	0	100.0	0.0
r	0	0	0	0	0	0	0	0	**51**	9	0	0	0	0	85.0	15.0
s	0	0	0	0	0	0	0	0	0	**60**	0	0	0	0	100.0	0.0
t	0	0	0	0	0	0	0	0	0	0	**41**	0	7	12	68.3	11.7
x	0	0	0	0	0	0	0	0	0	0	0	**60**	0	0	100.0	0.0
z	0	0	0	0	0	0	0	0	0	32	28	0	**0**	0	0.0	100.0
Tot															**50.3**	**32.8**

Bookman

	b	c	d	e	f	h	m	n	r	s	t	x	z	Out	%Corr	%Err.
b	**0**	0	0	0	0	60	0	0	0	0	0	0	0	0	0.0	100.0
c	0	**60**	0	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
d	0	0	**60**	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
e	0	0	0	**59**	0	0	0	0	0	0	0	0	0	1	98.3	0.0
f	0	0	0	0	**60**	0	0	0	0	0	0	0	0	0	100.0	0.0
h	0	0	0	0	0	**60**	0	0	0	0	0	0	0	0	100.0	0.0
m	0	0	0	0	0	0	**60**	0	0	0	0	0	0	0	100.0	0.0
n	0	0	0	0	0	0	0	**60**	0	0	0	0	0	0	100.0	0.0
r	0	0	0	0	0	0	0	0	**60**	0	0	0	0	0	100.0	0.0
s	0	0	0	0	0	0	0	0	0	**60**	0	0	0	0	100.0	0.0
t	0	0	0	0	0	0	0	0	0	0	**22**	0	38	0	36.7	63.3
x	4	0	0	0	0	0	0	0	1	4	0	**5**	0	50	8.3	8.3
z	0	0	0	0	0	0	0	0	0	0	0	0	**60**	0	100.0	0.0
Tot															**80.3**	**13.2**

Zapf Chancery

	b	c	d	e	f	h	m	n	r	s	t	x	z	Out	%Corr	%Err.
b	**0**	0	0	0	0	0	0	0	0	59	0	0	0	1	0.0	98.3
c	0	**60**	0	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
d	0	0	**0**	0	0	0	0	0	0	0	0	0	0	60	0.0	0.0
e	0	17	0	**42**	0	0	0	0	0	0	0	0	0	1	70.0	28.3
f	0	0	0	0	**0**	0	0	0	0	0	0	0	0	60	0.0	0.0
h	0	0	0	0	0	**0**	0	0	0	0	0	0	2	58	0.0	3.3
m	0	0	0	0	0	0	**0**	60	0	0	0	0	0	0	0.0	100.0
n	0	0	0	0	0	3	0	**57**	0	0	0	0	0	0	95.0	5.0
r	0	0	0	0	0	0	0	0	**0**	0	0	0	0	60	0.0	0.0
s	0	0	0	0	0	0	0	0	0	**29**	0	0	0	31	48.3	0.0
t	0	0	0	0	0	0	0	0	60	0	**0**	0	0	0	0.0	100.0
x	0	0	0	0	0	0	0	0	0	0	0	**0**	0	60	0.0	0.0
z	0	0	0	0	0	0	0	0	0	0	0	0	**0**	60	0.0	0.0
Tot															**24.1**	**25. 8**

Table 3: *Trained on times, automatically updated for each font.*

Avant Garde

	b	c	d	e	f	h	m	n	r	s	t	x	z	Out	%Corr	%Err.
b	**0**	0	0	0	0	60	0	0	0	0	0	0	0	0	0.0	100.0
c	0	**60**	0	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
d	0	0	**60**	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
e	0	0	0	**60**	0	0	0	0	0	0	0	0	0	0	100.0	0.0
f	0	0	0	0	**0**	0	0	0	0	60	0	0	0	0	0.0	100.0
h	0	0	0	0	0	**0**	0	60	0	0	0	0	0	0	0.0	100.0
m	0	0	0	0	0	0	**60**	0	0	0	0	0	0	0	100.0	0.0
n	0	0	0	0	0	0	0	**60**	0	0	0	0	0	0	100.0	0.0
r	0	0	0	0	0	0	0	0	**59**	1	0	0	0	0	98.3	1.7
s	0	0	0	0	0	0	0	0	0	**60**	0	0	0	0	100.0	0.0
t	0	0	0	0	0	0	0	0	0	0	**59**	0	1	0	98.3	1.7
x	0	0	0	0	0	0	0	0	0	0	0	**60**	0	0	100.0	0.0
z	0	0	0	0	0	0	0	0	0	31	29	0	**0**	0	0.0	100.0
Tot															**69.0**	**31.0**

Bookman

	b	c	d	e	f	h	m	n	r	s	t	x	z	Out	%Corr	%Err.
b	**0**	0	0	0	0	60	0	0	0	0	0	0	0	0	0.0	100.0
c	0	**60**	0	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
d	0	0	**60**	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
e	0	0	0	**60**	0	0	0	0	0	0	0	0	0	0	100.0	0.0
f	0	0	0	0	**60**	0	0	0	0	0	0	0	0	0	100.0	0.0
h	0	0	0	0	0	**60**	0	0	0	0	0	0	0	0	100.0	0.0
m	0	0	0	0	0	0	**60**	0	0	0	0	0	0	0	100.0	0.0
n	0	0	0	0	0	0	0	**60**	0	0	0	0	0	0	100.0	0.0
r	0	0	0	0	0	0	0	0	**60**	0	0	0	0	0	100.0	0.0
s	0	0	0	0	0	0	0	0	0	**60**	0	0	0	0	100.0	0.0
t	0	0	0	0	0	0	0	0	0	0	**60**	0	0	0	100.0	0.0
x	4	0	0	0	0	0	0	0	0	0	0	**56**	0	0	93.3	6.7
z	0	0	0	0	0	0	0	0	0	0	0	0	**60**	0	100.0	0.0
Tot															**91.8**	**8.2**

Zapf Chancery

	b	c	d	e	f	h	m	n	r	s	t	x	z	Out	%Corr	%Err.
b	**0**	0	0	0	0	0	0	0	0	60	0	0	0	0	0.0	100.0
c	0	**60**	0	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
d	0	0	**60**	0	0	0	0	0	0	0	0	0	0	0	100.0	0.0
e	0	0	0	**60**	0	0	0	0	0	0	0	0	0	0	100.0	0.0
f	0	0	0	0	**60**	0	0	0	0	0	0	0	0	0	100.0	0.0
h	0	0	0	0	0	**0**	0	0	60	0	0	0	0	0	0.0	100.0
m	0	0	0	0	0	0	**0**	60	0	0	0	0	0	0	0.0	100.0
n	0	0	0	0	0	0	0	**60**	0	0	0	0	0	0	100.0	0.0
r	0	0	0	0	0	0	0	0	**60**	0	0	0	0	0	100.0	0.0
s	0	0	0	0	0	0	0	0	0	**60**	0	0	0	0	100.0	0.0
t	0	0	0	0	0	0	0	0	60	0	**0**	0	0	0	0.0	100.0
x	60	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0.0	100.0
z	0	6	0	54	0	0	0	0	0	0	0	0	**0**	0	0.0	100.0
Tot															53.8	46.2

class-descriptions from other fonts for saving the work with constructing training sets.

In order to test the method on data that are not so well separated, the test was performed on simulated data constructed from the original class descriptions and added noise. The amount of noise was chosen such that the existence of large training sets would give error rates around 20-40%. By adding noise we wanted to show that this method has a large potential for symbols with large within class variation, such as f.ex. handwritten symbols.

When using updating schemes with classes known for the new symbols, we would like the method to behave such that for a small number of symbols from the new font, the method would use the information in the class-description base of the other font(s) to remove some of the variability due to a small training set. When the number of new symbols increase, the method should however put more weight on these and less on the class-description base for the other font(s). For this test we have therefore tried out different sizes of the training sets. The results are shown in figures 1-3.

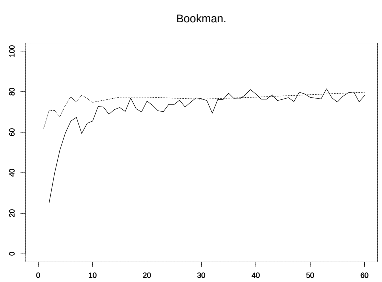

Figure 1: *Classification rate as a function of training set size for the Bookman font.*

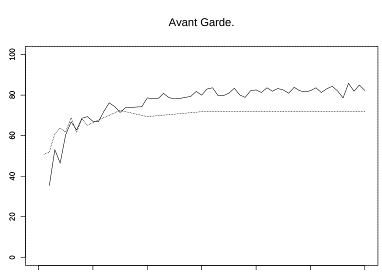

Figure 2: *Classification rate as a function of training set size for the Avant Garde font.*

The smoothest line in the figures describes the correct classification rate when the initial times-based class description was updated with an increasing number of symbols from the current font. The x-axis denotes the number of symbols from each font in the set used for updating and the y-axis the percentage of correctly classified symbols. The other line illustrates the correct classification rate when training and classification was performed on the symbols from the same font. The x-axis here denotes the number of symbols of each class in the training set.

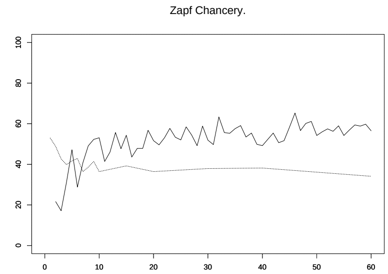

Figure 3: *Classification rate as a function of training set size for the Zapf Chancery font.*

For the Bookman font, the method behaves as wanted, in that for small training sets, the updating scheme is doing better than when only using the training set. When the size of the training set increases, the methods seems to behave equally well.

Also for the Avant Garde and the Zapf Chancery fonts, the updating scheme works better for small training sets. In this case, however, the updating scheme is doing worse for larger training sets. This may be due to giving too large weight to the original class-description base compared to the new training set.

4 Discussion and conclusion.

In this paper we have described methods for improving parameter estimates obtained from a training set by using a new set of data. This is done both for the case where we know the class membership of the new data and for the case in which these are unknown.

We have studied the use of procedures of updating schemes on class-description-bases in symbol-recognition. First we combined existing training sets with known correct classes for the symbols with a new training set with unknown classes for the symbols. These experiments showed promising results concerning simplification of the interactive training process traditionally used in symbol recognition.

The second situation considered was the use of class-description-bases originated from other fonts than the one of interest (the Bayesian approach) combined with a new training set with unknown classes for the symbols. For this situation the experiments showed that for classes with similar descriptions for different font-types the method was able to update the class-descriptions quite reasonably, while for classes not having this similarity feature, large classification error occurred.

Finally, the Bayesian approach was combined with a new training set with known classes for the symbols. In this case the methods worked quite well when the new training sets were small. For larger training sets, the methods had some unpleasant characteristics expected to be improved with more tuning of the parameters involved.

The experiments of updating-procedures so far have been applied only using the general technology described in section 2 without any special treatment to symbol recognition. Similar to the approach in Storvik et.al. [6] on MR-images, structures of how the feature-vectors change from one font to another could be built into the model. If plausible structures can be found, larger deviations between the class-descriptions for symbols of different fonts can be allowed when using the Bayesian approach.

Another improvement may be achieved by choosing the type of feature-vectors more carefully. The choice made in the experiments in the previous sections were only made for making the separation between classes of the same font-type as good as possible. When updating class-descriptions from other font-types is the concern, choosing feature-vectors that are more similar for different font-types may improve the updating performance.

References

[1] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm (with discussion). *Journal of Royal Statistical society, series B*, 39:1–38, 1977.

[2] L. Eikvil, M. Holden, and G. Storvik. Methods for updating of model parameters applied within the area of symbol recognition. Technical Report 859, Norwegian Computing Center, 1992.

[3] R. B. Huseby, J. Ólafsdóttir, and K. Aas. Classification of objects in multispectral images. Technical Report BILD/07/, Norwegian Computing Center, 1992.

[4] F. P. Kuhl and C. R. Giardina. Eliptic Fourier Features of a Closed Contour. *Computer Graphics and Image Processing*, 18:236–258, 1981.

[5] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis.* Academic Press, inc, 1979.

[6] G. Storvik, M. Holden, and V. Bosnes. Improving statistical image classification by updating model parameters using unclassified pixels. Technical Report 857, Norwegian Computing Center, 1992.

[7] C. F. J. Wu. On the convergence of the em-algorithm. *Ann. Statist.*, 11:95–103, 1983.