The choice of vantage objects for image retrieval

Christian Hennig [a] Longin Jan Latecki [b]

[a] *Universität Hamburg, Fachbereich Mathematik, 20146 Hamburg, Germany*

[b] *Universität Hamburg, Fachbereich Mathematik, 20146 Hamburg, Germany*

Abstract

Suppose that we have a matrix of dissimilarities between n images of a database. For a new image, we would like to select the most similar image of our database. Because it may be too expensive to compute the dissimilarities for the new object to all images of our database, we want to find $p < n$ "vantage objects" (Vleugels and Veltkamp, 2002) from our database in order to select a matching image according to the least Euclidean distance between the vector of dissimilarities between the new image and the vantage objects and the corresponding vector for the images of the database. In this paper we treat the choice of suitable vantage objects. We suggest a measure for the quality of a set of vantage objects, which averages the resulting dissimilarities when a different image is selected for each image of the same database. We compare two classes of choice strategies: The first one is based on a stepwise forward selection of vantage objects to optimize the quality function. The second is to choose objects as representative as possible for the whole range of the database.

Key words: Cross-validation, leave-one-out, stepwise forward selection, shape similarity

1 Introduction

In this paper we deal with the following problem: Suppose that we have a database of n images (objects). The information about the images is given

Email addresses: `hennig@math.uni-hamburg.de`,
`latecki@math.uni-hamburg.de` (Longin Jan Latecki).
URLs: `http://www.math.uni-hamburg.de/home/hennig`,
`http://www.math.uni-hamburg.de/home/latecki` (Longin Jan Latecki).

in form of $n(n-1)/2$ dissimilarities between them. For a new image, we would like to select the most similar image of our database. This requires the computation of n dissimilarities. Suppose that there is some computational effort to calculate a single dissimilarity, and that it is feasible to calculate a small number of dissimilarities, say 20 or 40, but not all n. Vleugels and Veltkamp (2002) suggest the following strategy: Choose a suitable number p of objects from the database as "vantage objects". Calculate the dissimilarities between the new object and the vantage objects. Interpret every object in the database, as well as the new object, as a p-dimensional vector in the Euclidean space, namely as the vector of dissimilarities to the vantage objects. Select the object in the database, whose vector of dissimilarities to the vantage objects has the smallest Euclidean distance to the vector of the new image. This means that for the classification of the new image only p dissimilarity calculations are needed.

The question arises how to choose the vantage objects. Vleugels and Veltkamp (2002) suggest some heuristic strategies. We present here a data driven approach to measure the quality of a set of vantage objects and we suggest and compare some strategies to find high quality sets. If p would be so small that evaluation of the quality of all $\binom{n}{p}$ vantage sets of size p would be possible, the quality measure could be optimized directly.

The quality measure and some of the suggested strategies are inspired by an old data-analytic idea, namely the principle of cross-validation (A general account is given by Stone (1974)). In general, this means that a rule for statistical prediction or choice can be assessed by dividing a dataset in two parts. One is used for the development of the rule, the other for the assessment of its quality. "Leave-one-out", a more refined form, is used frequently in classification, in particular for the selection of optimal features for discrimination, beginning with Lachenbruch and Mickey (1968). The principle here is as follows: If we have a training sample of n objects, each of them belonging to a known of k possible classes, and we want to assess the quality of a classification rule based on p of q features to classify a new observation into one of the classes, we divide the dataset n times (for each object) in parts of 1 and $n-1$ objects, respectively. Then we treat the class of the single object as unknown and classify it on the base of the other $n-1$ objects. The success rate gives a good quality measure for the discriminant rule, and features can be chosen by minimizing this experimental error rate.

The vantage object problem is familiar, but somewhat different from the feature selection problem in classification, insofar as the possible "features", i.e., the vantage objects, are the members of the dataset themselves, and the classification error rate has to be replaced by another loss function. The definition of the loss measure is treated in Section 2. Furthermore, as long as no simple probability models for object distances are available, there are no standard statistical competitors for data driven methods such as cross-validation.

For large p, the loss measure cannot be optimized directly. In Section 3 we discuss strategies to find vantage objects which optimize the quality measure locally on the base of a stepwise forward algorithm as used often for variable selection in discriminant analysis and linear regression, see e.g. Chapter 15 of Mosteller and Tukey (1977), and on the base of cross-validation. They can be compared with alternative strategies as that of Vleugels and Veltkamp (2002), which try to find objects which, in some manner, represent the whole database. Such techniques are discussed in Section 4. In Section 5, the strategies are applied to two database of 1189 shape images, 8090 respectively, where the dissimilarity is defined as given by Latecki and Lakämper (2000). The direct optimization of the loss measure on the database leads to a certain bias, if it is interpreted as a measure of the quality of the selection of similar images for new objects. This bias is assessed in Section 6. The approach is discussed in Section 7.

2 The quality measure

Let A be our database of $|A| = n$ objects. The aim of this section is to define a measure for the selection quality of a vantage object set $V \subset A$ with $|V| = p < n$.

First we give a mathematical model of the selection procedure of the most similar objects from the database A for a given fixed set $V \subset A$ with $|V| = p < n$ of vantage objects.

Let $D(q,r)$ with $D(q,q) = 0$ and $D(q,r) = D(r,q)$ denote the dissimilarity between the objects r and q. An example is given by Latecki and Lakämper (2000). Let $\vec{v}(q) = (D(q,a))_{a \in V}$ the vector of dissimilarities from object q to the vantage objects ($v_i(q)$, $i = 1, \ldots, p$, denotes the distance to the vantage object of ith smallest index), and let $d_V(q,r) = \|\vec{v}(q) - \vec{v}(r)\|$ the Euclidean distance between $\vec{v}(q)$ and $\vec{v}(r)$.

When the retrieval in the image database A is based on the set of vantage objects V, then for a query image q the best matching image $s_1(q,A)$ in A to q is usually chosen as the image with the smallest Euclidean distance $d_V(q,r)$ for $r \in A$ (for example, this is the case in Vleugels and Veltkamp (2002)). Thus,

$$s_1(q,A) = \arg\min_{r \in A} d_V(q,r),$$

where $\arg\min_{r \in A}$ denotes the element in A for which the minimum value $min_{r \in A} d_V(q, r)$ is reached.

Our approach generalizes to other selection functions based on vantage objects:

$$s_1(q, A) = \arg\min_{r \in A} s_V(q, r),$$

where we call s_V "selection function". The easiest choice is $s_V(q, r) = d_V(q, r)$. However, we observed that the weighted distance

$$s_V(q, r) = w_V(q, r) = \sum_{i=1}^{p} \frac{|v_i(q) - v_i(r)|}{0.3 + v_i(q)}$$

led to somewhat better results for the databases discussed in Section 5. This is reasonable because it may be expected that near vantage objects, i.e., objects with small $v_i(q)$, give better information about q.

For some applications it may be of interest to retrieve more than one similar image from the database for a given query image q. For this purpose we define $s_i(q, A)$ as the object $r \in A$ with ith smallest value[1] of $s_V(q, r)$, $i = 1, \ldots, n$. Let k be the number of selected objects of interest and let $S_V(q, A) = (s_1(q, A), \ldots, s_k(q, A))$ denote the whole selection. Thus, $S_V(q, A)$ is the list of the first k best matching images in A for a given query image q.

If the retrieval in the image database A is based on some comparison on the set of vantage objects V instead on the whole database A, then there is a loss in the retrieval performance. To model the extent of this loss, we define a "loss function" $l(q, S_V(q, A))$ that specifies the loss for the selection corresponding to a query object q. Our approach allows for various loss functions. We worked with two possible loss functions defined below.

One clearly would like that the dissimilarity value D between the query image q and the most similar object $s_1(q, A)$ retrieved for q is as low as possible. Ideally $s_1(q, A)$ should be the element with the smallest dissimilarity value for all elements of A, i.e., $D(q, s_1(q, A))$ should be lower or equal to $D(q, r)$ for all $r \in A$ and $r \neq s_1(q, A)$. This leads us to the following definition of the loss function.

If we want to retrieve only one most similar object for a query image q, we define

$$l_1(q, S_V(q, A)) = \min[D(q, s_1(q, A)), c].$$

[1] For ease of notation we assume that the object leading to the ith smallest value is always unique.

This is a refinement of the most intuitive choice $l_0(q, S_V(q, A)) = D(q, s_1(q, A))$ just described, where c is some cutoff value with the interpretation that c should be the smallest distance value such that a chosen object r with $D(q, r) = c$ is considered as a definitely inadequate matcher for q. The reason for introducing c is that from the viewpoint of application, it does not matter if $D(q, s_1(q, A)) = c$ or $D(q, s_1(q, A)) = 1000c$, if $s_1(q, A)$ is inadequate in each case, but the difference in distance would strongly affect the overall loss L defined in (1), if l_0 would be chosen as the loss function. The cutoff idea stems from robust statistics and was previously used in a cross-validation context by Ronchetti et al. (1997).

If k selected objects are of interest, but the importance of these objects decreases from the first to the last, we define

$$l_k(q, S_V(q, A)) = (\sum_{i=1}^{k} \frac{1}{i})^{-1} \sum_{i=1}^{k} \frac{\min[D(q, s_i(q, A)), c]}{i}.$$

Further reasonable loss functions can be imagined easily.

Given a loss function l, the quality of a selection rule S_V based on vantage objects V can be assessed by the overall loss function

$$L(S_V) = \frac{1}{n-p} \sum_{a \in A \setminus V} l(a, S_V(a, A \setminus \{a\})), \tag{1}$$

that is, $L(S_V)$ is the average loss over all $n-p$ non-vantage objects of A under selection from A, where a is left out. Here, as in leave-one-out cross-validation, every single a mimics a query object while $A \setminus \{a\}$ mimics the database. This is why the vantage objects, which are part of the original database, are not used as input of the selection rule, but may be selected to match a non-vantage a.

3 Stepwise approximate minimization of the overall loss

We now consider the form of the selection function s_V and the loss function l as given, but not the vantage set V. The natural approach to find V would be to optimize the quality measure L subject to $|V| = p$ or $|V| \leq q$ for some upper bound q. But this requires the evaluation of L for $\binom{n}{p}$ candidate vantage sets, which is computationally intractable even for moderate p. Another approach would be to choose the p best objects a according to $L(S_{\{a\}})$, but this cannot be expected to lead to satisfactory results because usually many of the found objects represent about the same selection information and thus most of them could be omitted without considerable loss.

To find a vantage set which minimizes L approximately, we adopt another strategy from the feature selection problem in data-analytic setups like discriminant analysis and regression, namely the stepwise forward selection (SFS; see e.g. Mosteller and Tukey (1977, Chapter 15)). The idea is that we search for the optimal set with one element first. Then we look for the optimal pair including the optimal first element and so on. Formally:

$$V = \{a_1, \ldots, a_p\}, \quad a_i = \mathop{\arg\min}_{a \in A \setminus \{a_1, \ldots, a_{i-1}\}} L(S_{\{a_1, \ldots, a_{i-1}, a\}}). \tag{2}$$

This reduces the number of evaluations of L to $n + (n-1) + \ldots + (n-p+1)$, while the resulting $L(S_V)$ can be expected to be a reasonable approximation to the global minimum. We call the whole strategy defined now SFLOO (stepwise forward leave-one-out).

Up to now we have considered the case of a pre-specified fixed number p of vantage objects. In linear regression the SFS can be complemented by an automatic choice of the number of features based on statistical tests. This is not possible here. An idea is to stop the enlargement of V when L is increased for the first time by the addition of the best new vantage object. As can be seen in Section 5, our experience is that sometimes surprisingly early some V is found where L does not get smaller by the addition of any single object. However, in such situations it makes almost always sense to add further vantage objects because L is again decreased later, so that we cannot recommend to stop if L is increased for the first time. For similar reasons, it does not seem to be worth the effort to adopt some modifications to SFS, e.g., combination of SFS and backward elimination as suggested in Mosteller and Tukey (1977, Chapter 15).

A better idea would be to specify a penalty term $C(p)$, increasing in p, which specifies the "cost" to have p vantage objects in a manner comparable to the selection loss. Then SFS may be used to minimize approximately $L(S_V)+C(p)$ subject to $|V| \leq q$. In our example, however, we restricted our attention to a pre-chosen p.

4 Alternative strategies

The calculation of the qth vantage object among the remaining $n-q+1$ non-vantage objects according to SFLOO requires $n-q+1$ times the calculation of the selection functions for $(n-q)(n-1)$ objects, i.e. it is of order n^3. Even though the vantage set has to be determined only once for a given database, this may take too much time for larger databases.

Here are some useful strategies to determine p vantage objects with smaller computational effort. Their results can be compared by calculation of the overall loss function L, as is done for our example in Section 5.

CV(t): This strategy replaces the leave-one-out cross-validation by a less computer intensive cross-validation scheme. Draw randomly without replacement two disjoint samples T_1 and T_2 of test objects from A, both of size t. Then, L from (1) can be replaced by

$$\hat{L}(S_V) = \frac{1}{t} \sum_{a \in T_1} l(a, S_V(a, T_2)), \tag{3}$$

that is, instead of selecting an object from $A \setminus \{a\}$ for each $a \in A$, only objects from a selection set T_2 are selected for objects from a test set T_1. The vantage objects can now be chosen according to (2) with L replaced by $\hat{L}$. A second difference is that only images from $A \setminus (T_1 \cup T_2)$ should become vantage objects, because the values of $\hat{L}$ are not comparable for the objects of T_1 and T_2 to those for the rest of A. The order of the number of selection function evaluations is nt^2, and test sets of size t between 100 and 1000 should work well and much faster than SFLOO. One could wonder why we do not take a single test sample T and assign all $a \in T$ in the LOO style to the elements of $T \setminus \{a\}$. In our experience, the relation between quality and computing time is more favorable when operating with disjoint test and selection sets T_1 and T_2.

NCV(t): The CV(t)-strategy has the disadvantage that not all elements of A can get into V. A slight variation consists in drawing new test and selection sets $T_1(V),\ T_2(V)$ from $A \setminus V$ for each candidate vantage set V occurring during the SFS. If we introduce $T_1(V),\ T_2(V)$ in (3), we can again optimize (2) over all $V = \{a_1, \ldots, a_{p-1}, a\},\ a \in A \setminus \{a_1, \ldots, a_{p-1}\}$. The advantage is paid by the fact that the estimated quality for the choice of a as a new vantage object does not base on the same test sets for all a, and some images may be favored by the drawing of a test and selection set which fit extraordinary well. The effect of this problem may however be small for a not too small p and t.

MAXIMIN: Vleugels and Veltkamp (2002) suggest to take a first vantage object randomly. Then, the second object should be the object with maximum distance to the first, and the further vantage objects are chosen in order to maximize the minimum distance to one of the previous vantage objects. The idea behind this strategy is that the vantage objects are thought to represent as good as possible the variety of the objects of the database.

CLUSTER: The same goal could be attained by performing a cluster analysis on the objects of A and choosing representative objects for the clusters. There are many methods of cluster analysis, some of them operating on dissimilarity matrices, others on Euclidean vectors. We have chosen a clustering algorithm which fits directly to our problem, i.e., a method that extracts p

Fig. 1. Some shapes from the manual chosen set of vantage objects.

medoid objects $m_1, \ldots, m_p \in A$ in order to minimize the objective function

$$\sum_{a \in A} \min_{j=1,\ldots,p} D(a, m_j),$$

that is, every object is assigned to the nearest medoid. The method was introduced in Kaufman and Rousseeuw (1990, Chapter 2) and is implemented in the statistical packages SPLUS and R. Unfortunately the computational effort for large datasets is high and we were only able to apply the method to the smaller of our two example databases. The result is somewhat disappointing, and therefore we refused to try other clustering approaches which may be computationally easier, but match our purposes less clearly.

1LOO: The last strategy is included only for comparison and consists of taking the best p objects from the first step of SFLOO, that is, the objects with lowest $L(S_{\{a\}})$. This requires as well $O(n^3)$ evaluations of the selection function, but it is more than p times faster than SFLOO because our selection functions increase in complexity with $|V|$. As remarked earlier, 1LOO cannot be expected to be a good strategy, because the vantage objects may get too similar.

Note that only SFLOO, CLUSTER and 1LOO lead to a deterministic choice of the vantage set. The other methods depend on random initializations.

5 An example

We compared the vantage sets V from several choice strategies and the selection functions $s_V = d_V$ and $s_V = w_V$ by evaluation of the overall loss function $L(S_V)$ for two databases A_1 and A_2. Both databases are composed from images retrieved from the Internet. Some images are shown in Figure 1. The dissimilarities between images were computed on the base of their shapes using the measure defined in Latecki and Lakämper (2000). For the A_1 database, we tried to chose the objects so that we cover as many as possible different shapes of objects. We also tried to cover all classes of the common everyday shapes of man-made and natural objects, like shape of cars, tools, humans, animals, and plants. A_1 consists of 1189 shapes. The A_2 database is a simple extension of A_1 to 8090 images.

The results are given in Table 1 for the database A_1 with loss function l_1, in Table 2 with loss function l_{20} ("l_1-SFLOO" refers to the vantage objects from SFLOO calculated with loss function l_1, but evaluated with l_{20}) and in Table 3 for the database A_2. In the latter case, only l_1 has been considered.

Table 1
Database A_1: Comparison of several strategies for vantage object choice (loss function l_1).

Strategy	selection function	p	$L(S_V)$	min	max
Optimal			0.4207		
$V = A$	d_V	1189	0.5896		
$V = A$	w_V	1189	0.5927		
Manual	d_V	40	0.6845		
Manual	w_V	40	0.6774		
SFLOO	d_V	40	0.6258		
SFLOO	w_V	40	0.6207		
MAXIMIN	d_V	40	0.6593	0.6556	0.6625
MAXIMIN - stopped	d_V	9	0.7653		
MAXIMIN	w_V	40	0.6387		
CLUSTER	d_V	40	0.6783		
CLUSTER	w_V	40	0.6864		
CV(100)	d_V	40	0.6726	0.6560	0.6963
CV(100) - stopped	d_V	9	0.7550		
CV(100)	w_V	40	0.6699		
CV(MAXIMIN)	d_V	40	0.6879		
CV(CLUSTER)	d_V	40	0.6812		
CV(200)	d_V	40	0.6743	0.6692	0.6820
CV(200) - stopped	d_V	17	0.7161		
NCV(100)	d_V	40	0.6673	0.6604	0.6706
NCV(100) - stopped	d_V	24	0.6991		
NCV(100)	w_V	40	0.6763		
NCV(200)	d_V	40	0.6720	0.6606	0.6816
NCV(200) - stopped	d_V	13	0.7405		
1LOO	d_V	40	0.7365		
1LOO - stopped	d_V	6	0.8184		

For all setups we calculated further the optimal possible value of $L(S_V)$, i.e. the average loss under selection of the image $b_0 \in A$ for a with $b_0 = \arg\min_{b \neq a} D(a, b)$. We also calculated the overall loss for $V = A$, i.e., the set of vantage objects

Table 2
Database A_1: Comparison of several strategies for vantage object choice (loss function l_{20}).

Strategy	selection function	p	$L(S_V)$
Optimal			0.5206
$V = A$	d_V	1189	0.7021
Manual	d_V	40	0.7691
SFLOO	d_V	40	0.7101
l_1-SFLOO	d_V	40	0.7382
MAXIMIN	d_V	40	0.7439
CLUSTER	d_V	40	0.7681
CV(100)	d_V	40	0.7371
NCV(100)	d_V	40	0.7748

is the whole database. Because of $n - p = 0$ in this case, the sum in (1) has to be taken over all $a \in A$ and has to be divided by n. The obtained value is an approximation for the best value of $L(S_V)$ which can be attained by use of the method of vantage objects. Note that it cannot be expected in general that $V = A$ minimizes the loss over all V, but we do not believe that there is some $V \subset A$ for which the value is considerably smaller.

The entry "Manual" refers to a vantage set of 40 shapes from A_1 which was chosen manually and intuitively to represent the database well, including the shapes of Figure 1. We do not have the "Manual" entry for A_2, since choosing manually 40 shapes that optimally represent the database of 8090 shapes seems to be impossible.

We performed some extra comparisons with database A_1, loss function l_1 and selection function d_V. The non-deterministic strategies were applied five times to assess their variability. In these cases, the "overall loss" value is a mean, and the minimum and maximum values are also given. Furthermore, we tested the idea if properly chosen test sets T_1 and T_2 can improve the quality of the CV method. "CV(MAXIMIN)" indicates that 200 objects were chosen according to the MAXIMIN strategy, 100 assigned randomly to T_1 and 100 to T_2, by analogy "CV(CLUSTER)".

For some of the strategies in the A_1/l_1-setup, we looked for the value of the overall loss for the lowest number p of vantage objects such that $L(S_V)$ had been increased for $p+1$ objects. This is called the "stopped" variant in Table 1. For the non-deterministic strategies, only one run was examined. The stopping rule did generally not lead to good values of the overall loss, because $L(S_V)$ was always again decreased for more than one further object. The general tendency

Table 3
Database A_2: Comparison of several strategies for vantage object choice (loss function l_1).

Strategy	selection function	p	$L(S_V)$
Optimal			0.3113
$V = A$	d_V	8090	0.4779
$V = A$	w_V	8090	0.4859
SFLOO	d_V	10	0.5837
MAXIMIN	d_V	10	0.6491
CV(200)	d_V	10	0.6451
NCV(200)	d_V	10	0.6377
MAXIMIN	d_V	40	0.5455
MAXIMIN	w_V	40	0.5339
CV(200)	d_V	40	0.5505
CV(200)	w_V	40	0.5495
CV(500)	d_V	40	0.5410
NCV(200)	d_V	40	0.5448
NCV(200)	w_V	40	0.5530
NCV(500)	d_V	40	0.5442

that the larger p is, the smaller is the overall loss, is not changed substantially by the use of the stopping rule. SFLOO was monotonely decreased for the first 40 vantage objects.

Because of the computational effort, we determined only 10 vantage objects with SFLOO in A_2. The results of the other strategies are given for $p = 10$ and $p = 40$.

The main results are as follows: SFLOO leads clearly to the best results and should be applied if it is computationally feasible. MAXIMIN is the fastest method and yields in most cases the second best vantage set. The CV and NCV methods are similar to MAXIMIN in quality. The results of the A_1/d_W setup indicate that the quality differences between MAXIMIN, CV and NCV could possibly be explained by random variation only. MAXIMIN has the lowest variation, CV has the largest, so that MAXIMIN is to be preferred when only one vantage set should be calculated, but if one would like to take the best vantage set from 20 or 50 runs of a method, say, the best CV run may outperform the best MAXIMIN run. All these methods are better than the manual choice.

Table 4
Quality of vantage objects from sub-database A_{11} on A_{11} and A_{12} (loss function l_1, selection function d_V).

Strategy	p	$L_{11}(S_{V_{11}})$	$L_{12}(S_{V_{11}})$
Optimal		0.4697	0.4604
$V = A_{11}$	595	0.6480	0.6308
SFLOO	20	0.6878	0.7257
MAXIMIN	20	0.7500	0.7330
CLUSTER	20	0.7601	0.7455
CV(50)	20	0.7727	0.7490
NCV(50)	20	0.7533	0.7412

The size of the test samples for CV and NCV used here does not seem to matter, while, of course, a very small test sample will not work well, and very large test samples make vanishing the speed advantage over SFLOO.

The results of CLUSTER, CV(CLUSTER) and CV(MAXIMIN) are not as good as we hoped. It remains a problem for further research if there is a better clustering method for this purpose. At least for the databases treated here it does not pay to use "representative" test samples for CV. As expected, 1LOO performed badly.

The weighted selection function w_V performs a little bit better in most cases than d_V, but not always. At least in combination with MAXIMIN it seems to be preferable to d_V.

The results with l_{20} differ somewhat from the results with l_1. The objects from l_1-SFLOO do not perform badly, but the performance is still considerably improved if SFLOO is carried out with the same loss function which is used for the calculation of the overall loss. We think that the loss function should always be tailored as exactly as possible to the application.

6 Assessment of the selection bias

The comparison of the different strategies by means of the values of $L(S_V)$ leads to the so-called "selection bias" (see e.g. Berk (1978)). This means that the value for $L(S_V)$ can be too optimistic as an estimate for the assignment quality of new images, if the vantage objects V were obtained by direct optimization of $L(S_V)$ for the old images, which is done by SFLOO at least approximately.

The only way to assess the selection bias is the use of independent images. If the images of our database are the only images at hand, the database must be split into two parts. The first part can be used to perform the search for good vantage objects, and the second part can be used for the assessment of the quality of the vantage sets applied to independent images. However, it can be expected to be more efficient to use them for the improvement of the vantage set instead of the assessment of the selection bias.

While we suggest to take finally the vantage objects based on the whole database, we performed a study to assess the selection bias at least for the smaller database A_1. For this sake A_1 was split randomly in two disjunct parts A_{11} and A_{12} with $|A_{11}| = 595$ and $|A_{12}| = 594$. The strategies SFLOO, CV, NCV, MAXIMIN and CLUSTER were again performed on A_{11} to yield vantage sets V_{11} independent of A_{12}, and the quality of selecting images from A_{11} for the new images from A_{12} was measured by

$$L_{12}(S_{V_{11}}) = \frac{1}{|A_{12}|} \sum_{a \in A_{12}} l(a, S_{V_{11}}(a, A_{11})). \tag{4}$$

The absolute values of $L_{12}(S_{V_{11}})$ cannot estimate properly the quality of a vantage set V chosen with the same strategy on A_1, because $|A_{11}| < |A_1|$, and therefore it can be expected that better matchers for new images can be found in A_1 than in A_{11}. But it is of interest if the ranking of strategies with respect to $L_{12}(S_{V_{11}})$ remains the same as for $L_{11}(S_{V_{11}})$ on A_{11}, which is the analogon of $L(S_V)$ on A_1, and in particular if SFLOO remains to be the best. To keep the circumstances comparable, $t = 50$ for CV and NCV and $p = 20$ were chosen.

The results are given in Table 4. They show that SFLOO still produces the best vantage set, but with much smaller lead. MAXIMIN seems to be better than CV and NCV now, and CLUSTER is no longer the worst, but still not very good. The ranking of strategies w.r.t. $L_{12}(S_{V_{11}})$ is the same than w.r.t. $L_{11}(S_{V_{11}})$. Again it must be noted that the results depend on chance. In general, $L_{11}(S_{V_{11}})$ should be a more optimistic estimate than $L_{12}(S_{V_{11}})$ for all strategies carried out on A_{11}, so that it is surprising that $L_{11}(S_{V_{11}})$ is larger than $L_{12}(S_{V_{11}})$ for all strategies except of SFLOO here. This indicates that the images of A_{11} are somewhat more difficult to match than those of A_{12}, which is confirmed by the optimal possible average loss values to assign images from A_{11}, A_{12}, respectively, to images of A_{11}, given again under the entry “Optimal”, and by the results of $V = A_{11}$. It is an artifact of the random selection of A_{11}. The results must be interpreted with care.

7 Discussion

The approach developed here is neither restricted to a particular dissimilarity nor to a particular loss or selection function. It is always possible to compare vantage object sets stemming from various choice strategies by a quality measure based on the "leave-one-out" principle. If n is not too large, it is promising to optimize the quality locally by stepwise forward selection. In larger databases, other approaches like maximin distance or strategies based on cross-validation with random test sets are to be compared. In our examples, the maximin distance method led to good results, is fast and easily implemented, so that it looks like a good choice, but this finding cannot necessarily be generalized.

From a statistical point of view, the overall loss function (1) is only an estimation of the "real" selection loss of a selection rule for new unknown images. In our setup there is no statistical model, and therefore the theoretical properties of this estimation cannot be analyzed. But there are some results for "leave-one-out" cross-validation in more accessible problems. In general it can be said that cross-validation leads to almost unbiased estimations of a prediction error in discriminant analysis and regression and is clearly superior to naive approaches. The assumptions for cross-validation to work are listed clearly in Stone (1978), where the author refers as well to theoretical results about some situations where the principle may lead to suboptimal decisions. If features are selected by the optimization of a CV-based measure, the measure has a selection bias, which can be assessed by performing a "double cross-validation" (Mosteller and Tukey, 1977, pp. 36f.), i.e. a split of the database into two parts, one of which is only used to assess the selection bias. We suggest to do this only for the illustration of the bias, and to use the features chosen by use of the whole database finally.

For some data-analytical problems, cross-validation can be replaced or complemented by more refined data driven techniques such as "bootstrap" (Efron and Tibshirani (1993) refer to some papers about comparing simulations between cross-validation and bootstrap), but it is not clear how to adapt them to our setup, and the authors continue to acknowledge cross-validation as a widely applicable and useful approach.

An alternative approach to the selection of a similar image could be to build a decision tree on the database based on vantage objects that decide between the branches of the tree. It would be of interest to compare the resulting selection method with ours by means of the leave-one-out quality measure.

References

B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap* (Chapman & Hall, New York, 1993).

K. N. Berk, Comparing subset regression procedures. *Technometrics* **20** (1978) 1–6.

L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis* (Wiley, New York, 1990).

P. A. Lachenbruch and M. R. Mickey, Estimation of Error Rates in Discriminant Analysis. *Technometrics* **10** (1968) 1–11.

L. J. Latecki and R. Lakämper, Shape Similarity Measure Based on Correspondence of Visual Parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000) 1–6.

F. Mosteller and J. W. Tukey, *Data Analysis and Regression* (Addison-Wesley, Reading, 1977).

E. Ronchetti, C. Field and W. Blanchard, Robust Linear Model Selection by Cross-Validation, *Journal of the American Statistical Association* **92** (1997) 1017–1023.

M. Stone, Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B* **36** (1974) 111–133.

M. Stone, Cross-validation: A review. *Statistics (Mathematische Operationsforschung und Statistik)* **9** (1978) 127–139.

J. Vleugels and R. Veltkamp, Efficient Image Retrieval through Vantage Objects. *Pattern Recognition* **35** (2002) No. 1 - to appear.