PERGAMON

Pattern Recognition 36 (2003) 2187–2196

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

The choice of vantage objects for image retrieval

Christian Hennig[a,b,*], Longin Jan Latecki[c]

[a] *ETH Zürich, Seminar für Statistik, Zürich CH-8092, Switzerland*
[b] *Fachbereich Mathematik, Universität Hamburg, Hamburg 20146, Germany*
[c] *Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122, USA*

Received 2 November 2001; received in revised form 26 August 2002; accepted 7 October 2002

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 $p \ll n$ "vantage objects" (Pattern Recognition 35 (2002) 69) 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 loss measure to assess the quality of a set of vantage objects: For every image, we select a matching image from the remaining images of the database by use of the vantage set, and we average the resulting dissimilarities. We compare two classes of choice strategies: The first one is based on a stepwise forward selection of vantage objects to optimize the loss measure. The second is to choose objects as representative as possible for the whole range of the database.

Keywords: 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 in form of $n(n-1)/2$ dissimilarities between them. For a new image, we would like to select the most similar image from 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 [1] 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 [1] suggest some heuristic strategies. We present here a data driven approach to measure the quality of a set of vantage objects by means of a loss function and we suggest and compare some strategies to find high quality sets. If p would be so small that evaluation of the loss of all $\binom{n}{p}$ vantage sets of size p would be possible, the loss function could be optimized directly.

* Corresponding author. Seminar fur Statistic, ETH Zurich (LEO), Zurich CH-8092, Switzerland. Tel.: +41-632-6184; fax: +41-632-1228.

E-mail addresses: henni@math.uni-hamburg.de (C. Hennig), latecki@temple.edu (L.J. Latecki).

URLs: http://www.math.uni-hamburg.de/home/hennig, http://www.cis.temple.edu/latecki/

0031-3203/03/$30.00
PII: S0031-3203(02)00326-6

There are two well-known problems in multivariate statistics that have some similarities to the vantage object problem: As in the variable selection problem in classification [2, Chapter 6], a subset of a "feature set" (namely the set of objects, variables, respectively) is to be constructed in order to perform a computationally easier classification on the base of this subset. Secondly, there are some methods to look for "representative objects" of a dataset in the context of cluster analysis, see [3, Chapter 2].

The loss function 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 [4]). In general, this means that a rule for statistical prediction or classification 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 for variable selection in classification, beginning with Lachenbruch and Mickey [5]. 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 misclassification rate gives a good loss measure for the discriminant rule, and features can be chosen by minimizing this experimental error rate.

The vantage object problem needs another kind of loss measure, which is defined in Section 2. 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 a strategy to find vantage objects which optimize the loss 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. [6, Chapter 15]; [2, Chapter 6], and on the base of cross-validation. They can be compared with alternative strategies, which try to find objects which, in some manner, represent the whole database. The strategy of Vleugels and Veltkamp [1] belongs to this class as well as the search for objects representing clusters in the data. Such techniques are discussed in Section 4. Some extensions of our approach are introduced in Section 5.

In Section 6, we apply the strategies to four databases where three different dissimilarity measures between the images are used.

The direct optimization of the loss measure on the database leads to a certain bias, if it is interpreted as an estimator for the loss of the selection of similar images for new objects. This bias is assessed in Section 7. Some discussion is given in Section 8.

2. The loss measure

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

Firstly 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 [7]. Let $\vec{v}(q) = (D(q,a))_{a \in V}$ the vector of dissimilarities from object q to the vantage objects, i.e., $v_i(q)$, $i = 1, \ldots, p$, denotes the dissimilarity 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 [1]). Thus,

$$s_1(q,A) = \arg\min_{r \in A} \mathrm{d}_V(q,r), \tag{1}$$

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

Compared to the selection of the best matching image from A for some query image, the vantage object approach leads to some loss in the retrieval performance. To model the extent of this loss, we define a "loss function" $l(q, s_1(q,A))$ that specifies the loss for the selection corresponding to a query object q.

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:

$$l_0(q, s_1(q,A)) = D(q, s_1(q,A)). \tag{2}$$

Note that a loss of 0 can only be reached if identical images, i.e., images with dissimilarity 0 from the query objects, are present in the dataset. Therefore, in most cases the optimal possible value of l_0 will be larger.

Alternatives for the selection rule (1) and the loss function (2) are given in Section 5.

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

$$L(s_1) = \frac{1}{n} \sum_{a \in A} l(a, s_1(a, A \backslash \{a\})), \tag{3}$$

that is, $L(s_1)$ is the average loss over all n 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 $A\backslash\{a\}$ mimics the database.

3. Stepwise approximate minimization of the overall loss

We now consider the form of the selection function h_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 overall loss L subject to $|V| = p$ or $|V| \leqslant q$ 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 variable selection problem in data-analytic setups like discriminant analysis and regression, namely the stepwise forward selection (SFS); (see e.g. [6, 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

$$V = \{a_1, \ldots, a_p\}, \quad a_i = \operatorname*{arg\,min}_{a \in A\backslash\{a_1,\ldots,a_{i-1}\}} L(S_{\{a_1,\ldots,a_{i-1},a\}}). \tag{4}$$

This reduces the number of evaluations of L to $n + (n-1) + \cdots + (n-p+1)$, 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 stepwise forward leave-one-out (SFLOO).

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 6, 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 [6, 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 $|V| \leqslant q$. In our examples, 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 examples in the Sections 6 and 7.

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 (3) can be replaced by

$$\hat{L}(S_V) = \frac{1}{t} \sum_{a \in T_1} l(a, S_V(a, T_2)), \tag{5}$$

that is, instead of selecting an object from $A\backslash\{a\}$ 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 (4) with L replaced by $\hat{L}$. A second difference is that only images from $A\backslash(T_1 \cup T_2)$ 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 $T\backslash\{a\}$. 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 $A\backslash V$ for each candidate vantage set V occurring during the SFS. If we introduce $T_1(V)$, $T_2(V)$ in (5), we can again optimize (4) over all $V = \{a_1, \ldots, a_{p-1}, a\}$, $a \in A\backslash\{a_1, \ldots, a_{p-1}\}$. The advantage is paid by the fact that the estimated loss 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 [1] 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 medoid objects $m_1,\dots,m_p \in A$ in order to minimize the objective function

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

that is, every object is assigned to the nearest medoid. The method was introduced in [3, 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. Extensions

The selection rule (1) and the loss function (2) from Section 5 are not the only reasonable choices. In this section, we discuss some alternatives.

Rule (1) has the form

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

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

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

led to somewhat better results for the database A_3 discussed in Section 6. 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, and such objects get a larger weight in the computation of w_V.

The loss function l_0 may be replaced by a more refined version as well:

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

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 (3), 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. [8]. It is most useful if the value range of the dissimilarities is very large.

For some applications it may be of interest to retrieve more than one similar image from the database for a given query image q. In this case, we may wish to find vantage objects that minimize a loss function depending not only on the first selected image. For this purpose we define $s_i(q,A)$ as the object $r\in A$ with ith smallest value[1] of $h_V(q,r)$, $i = 1,\dots,n$. Let k be the number of selected objects of interest and let $S_V(q,A) = (s_1(q,A),\dots,s_k(q,A))$ 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.

For the case that k selected objects are of interest, we define

$$l_k(q,S_V(q,A)) = \left(\sum_{i=1}^{k} z_i\right)^{-1} \sum_{i=1}^{k} z_i h_V(q,s_i(q,A)), \qquad (6)$$

where $z_i > 0,\ i = 1,\dots,k$, are weights corresponding to the relative importance of the ith best selected object. Further reasonable loss functions can be imagined easily.

6. Examples

We compared the vantage sets V from several choice strategies applied to four databases by evaluation of the overall loss function $L(S_V)$, where $S_V(q,A) = s_1(q,A)$ unless stated explicitly.

The first example database A_1 consists of 1100 shapes of fishes form

`http://www.ee.surrey.ac.uk/Research/VSSP/imagedb/demo.html`.

The database has been used in [9], where the use of a new dissimilarity measure between shapes is proposed. This measure is a pseudo-metric, i.e., it is symmetric and satisfies the triangle inequality and each object has dissimilarity 0 to itself. The measure is applied to boundary contours of 2D objects. First feature points are extracted from boundary contours, such as edge and corner points. The set of features of one object is then translated, rotated, and scaled so as to minimize some similarity function with respect to the features from the other object.

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

Table 1
Database A_1: Comparison of several strategies for vantage object choice (loss function l_0, selection function d_V)

Strategy	p	$L(S_V)$
Optimal		0.1171
$V = A$	1100	0.1345
Demo default	12	0.1535
SFLOO	12	0.1410
MAXIMIN	12	0.1544
CLUSTER	12	0.1552
CV(100)	12	0.1538
NCV(100)	12	0.1516

Table 2
Database A_2: Comparison of several strategies for vantage object choice (loss function l_0, selection function d_V)

Strategy	p	$L(S_V)$	$R(S_V)$
Optimal		0.0458	0.951
$V = A$	100	0.0516	0.927
SFLOO-l_0	4	0.0480	0.925
SFLOO-l_9	4	0.0546	0.920
MAXIMIN	4	0.0552	0.910
CLUSTER	4	0.0729	0.898

The results from our strategies are given in Table 1. A demonstration of image retrieval with A_1 can be found on http://give-lab.cs.uu.nl/Matching/ptd/. The vantage object approach is demonstrated as well, based on 12 given default vantage objects (which can be changed manually). These objects are referred to as "demo default",[2] and this has been the reason why our comparisons are based on 12 vantage objects.

Further, we calculated 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 is the whole database. It could be believed that the obtained value approximates the best value of $L(S_V)$ to be attained by use of the method of vantage objects. However, it cannot be expected in general that $V = A$ minimizes the loss over all V, and sometimes much smaller vantage sets can do better, as can be seen in the second example (Table 2).

SFLOO gives clearly the best results for this database, while the differences between the losses of the other strategies are small and might be explained by the random variations of MAXIMIN, CV and NCV alone.

The database A_2 contains only 100 images from movies. The images can be seen on www.cis.temple.edu/~latecki/ImSim. The advantage of this database is that a ground truth retrieval rate is known. There are 10 images from each sequence, so that there are 10 known classes of images. That is, the results obtained from our strategies can be compared with the true retrieval rate.

Let $k = 10$ be the size of the classes and $m = 10$ be the number of classes. With $S_V(q,A) = (s_1(q,A), \ldots, s_k(q,A))$, and $g(q) \in \{1, \ldots, m\}$ denoting the class of object q, the retrieval rate is defined as the proportion of objects of the correct class among the first k retrieved objects:

$$R(S_V) = \frac{1}{nk} \sum_{a \in A} |i \in \{1, \ldots, k\}: g(a) = g(s_i(a,A))|. \quad (7)$$

By convention, $a = s_1(a,A)$ is included in the calculation of R.

To generate the database A_2, we used the metric of Hu and Mojsolovic [10], which is applied to measure dissimilarity of digital color images. The first step of distance computation is to obtain a compact, perceptually relevant representation of the color content of an image. This representation is obtained by a kind of rough segmentation of a given image to obtain perceptually dominant colors. Once the dominant colors are extracted, the image is represented as a vector of pairs (I_i, P_i), where I_i is the index to a color in a particular color codebook and P_i is the area percentage occupied by that color. The actual distance between two images is computed by finding the optimal mapping function between their vector representations that minimizes the overall mapping distance.

We decided to work with 4 vantage objects because the sequences stem from 4 movies. The results are shown in Table 2. The entry for "optimal $R(S_V)$" is based on the 10 most similar images for every image; the theoretically maximal possible value for $R(S_V)$ is of course 1.000. "SFLOO-l_9" means that the SFLOO strategy is used to optimize the loss based on l_9 as defined in (6). The weights are chosen as $z_1 = \cdots = z_9 = 1$. The reason for using this loss function here is that the retrieval rate is based on the choice of the 9 best matching images for a query image (plus the query image itself). We try to simulate the situation that we do not know the correct classes, but we know that 9 matching images (all of the same importance) for a query image are to be found. $L(S_V)$ is nevertheless computed on the base of l_0 for all strategies. In this example, the use of l_9 does not pay, because SFLOO-l_0 is not only the best with respect to $L(S_V)$, but also with respect to the retrieval rate. Its value of $L(S_V)$ is even better than that of $V = A$. However, SFLOO-l_9 gives almost the same retrieval rate, and the idea of using more than one selected object is supported by the results of Table 6 in Section 7. Note that Table 2 shows a good correspondence between the quality ranking in terms of our loss function $L(S_V)$ and in terms of the retrieval rate.

[2] As far as we know, the demo default objects result from an application of MAXIMIN. The result may differ from our MAXIMIN result because MAXIMIN depends on chance. The demo default objects are 2, 12, 22, 107, 307, 427, 450, 648, 800, 861, 999, 1025. We obtained 157, 565, 757, 592, 1090, 34, 434, 653, 600, 942, 49, 266 from SFLOO.

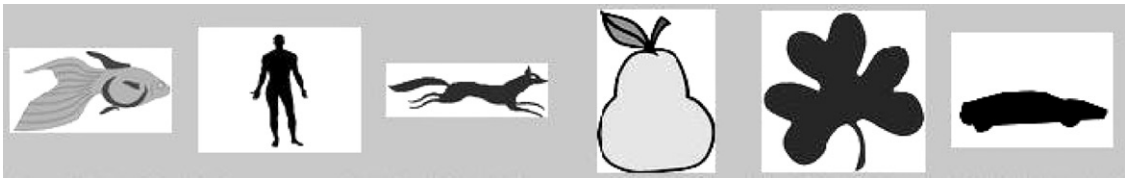

Fig. 1. Database A_3: Some shapes from the manual chosen set of vantage objects.

This can be seen as a justification for the choice of our loss function.

MAXIMIN and CLUSTER are clearly worse with respect to both criteria. The strategies CV(t) and NCV(t) are intended to reduce the computational effort of SFLOO. Therefore they are only recommended for t much smaller than $n/2$. We did not apply them to A_2 because for such t this database is too small to form reliable subsets and SFLOO is fairly fast.

The two databases A_3 and A_4 are composed from images retrieved from the Internet. Some images are shown in Fig. 1. The dissimilarities between images were computed on the base of their shapes using the shape similarity measure defined in [7]. It is designed to compare the shapes of silhouettes of 2D objects. To reduce influence of digitization noise as well as segmentation errors the shapes are first simplified by a novel process of digital curve evolution. To compute the similarity measure, the best possible correspondence of visual parts is established first. Then the similarity between corresponding parts is computed and aggregated. The obtained shape similarity measure does not obey the triangle inequality. It achieved an excellent retrieval performance in Core Experiment CE-Shape-1 of the MPEG-7 standard [11].

For the A_3-database, we tried to choose the objects so that we cover as many different shapes of objects as possible. We also tried to cover all classes of the common everyday shapes of man-made and natural objects, like shapes of cars, tools, humans, animals, and plants. A_3 consists of 1189 shapes. The A_4 database is an extension of A_3 to 8090 images.

The results[3] for database A_3 are given in Table 3. We applied the loss function l_c with $c = 2$. This means that two images with dissimilarity larger than 2 are considered as definitely inadequate matchers for each other. While about 97.5% of the dissimilarities are smaller than 2, the remaining values go up to a maximum of 8.32. The use of l_c should prevent that a single selection of a very bad matcher for a query image during the LOO procedure excludes an otherwise good candidate image from getting a vantage object.

[3] Note that the entries for $L(S_V)$ in Tables 3 and 4 have been computed with a slight modification of (3): The loss has only been averaged over the non-vantage objects (except of the case "$V = A$"). As far as we have checked it, the differences are negligible. But the use of form (3) is definitely recommended for small databases such as A_2.

The entry "Manual" refers to a vantage set of 40 shapes from A_3 which was chosen manually and intuitively to represent the database well, including the shapes of Fig. 1. We do not have the "Manual" entry for A_4, since it seems to be impossible to choose manually shapes that optimally represent such a large database.

We performed some extra comparisons with database A_3, loss function l_c 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.

For some of the strategies in the A_3/l_c-setup, we looked for the first number p of vantage objects such that $L(S_V)$ for $p+1$ objects is larger than for p. The value of $L(S_V)$ for such p is listed as "stopped" variant in Table 3. 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 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 the computational effort for SFLOO becomes horrible for A_4, we computed only 10 vantage objects with SFLOO. This needed more than a week of computation time. The present implementation of CLUSTER causes memory problems. Note that the authors Kaufman and Rousseeuw [3] acknowledge the problems for such large datasets, but their recommended alternative does not work for distance data. The other strategies have been evaluated with $p = 10$ and $p = 40$. The results from A_4 are given in Table 4.

The main results from A_3 and A_4 are as follows: SFLOO leads clearly to the best results as long as 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 loss. The results of the A_3/d_W setup indicate that the loss 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 3
Database A_3: Comparison of several strategies for vantage object choice (loss function l_c)

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(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		

Table 4
Database A_4: Comparison of several strategies for vantage object choice (loss function l_c)

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

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 are not as good as we hoped. It remains a problem for further research if there is a better clustering method for this purpose. 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.

7. Assessment of the selection bias

$L(S_V)$ can be interpreted as an estimator for the expected loss of the selection from A for a new query object. For a fixed set of vantage objects V, this estimator is only biased very weakly, because $L(S_V)$ is computed based on selections out of $n-1$ images ($A\backslash\{a\}$ for all $a \in A$), while we select from all n images for a new query object.

But if V stems from the optimization of $L(S_V)$, which is done at least approximately by SFLOO, this can induce a severe bias into $L(S_V)$, the so-called "selection bias" (see e.g. [12]). The reason can be understood easily: Consider a perfect dice. Of course the relative frequency of throwing a "5" will be an unbiased estimator for the probability of this event, which is $\frac{1}{6}$. However, the relative frequency of the fewest thrown number can be expected to be more or

Table 5
Average loss from 10 random splits of database A_1 into training sample A_{11} and validation sample A_{12} with 550 objects each (loss function l_0, selection function d_V)

Strategy	p	$L_{11}(S_{V_{11}})$	$L_{12}(S_{V_{11}})$
SFLOO	6	0.1618	0.1659
MAXIMIN	6	0.1725	0.1722
CLUSTER	6	0.1751	0.1746
CV(50)	6	0.1731	0.1727
NCV(50)	6	0.1713	0.1715

Table 6
Average loss from 10 random splits of database A_2 into training sample A_{21} and validation sample A_{22} with 50 objects each (loss function l_0, selection function d_V)

Strategy	p	$L_{21}(S_{V_{21}})$	$R_{21}(S_{V_{21}})$	$L_{22}(S_{V_{21}})$	$R_{22}(S_{V_{21}})$
SFLOO-l_0	2	0.0845	0.899	0.0935	0.885
SFLOO-l_4	2	0.0887	0.912	0.0964	0.901
MAXIMIN	2	0.1162	0.843	0.1144	0.818
CLUSTER	2	0.1166	0.805	0.1054	0.774

Table 7
Loss from random split of database A_3 into training sample A_{31} with 595 objects and validation sample A_{32} with 594 objects (loss function l_c, selection function d_V)

Strategy	p	$L_{31}(S_{V_{31}})$	$L_{32}(S_{V_{31}})$
SFLOO	20	0.6897	0.7332
MAXIMIN	20	0.7495	0.7580
CLUSTER	20	0.7470	0.7691
CV(50)	20	0.7447	0.7662
NCV(50)	20	0.7396	0.7785

less smaller than $\frac{1}{6}$ (depending on the number of throws), and if this number is "5" by chance, the (minimal) relative frequency of "5" must be a downward biased estimator for its probability. The selection bias problem is analogous.

This means that the value for $L(S_V)$ can be too optimistic for SFLOO and also, but to a smaller extent, for CV and NCV.

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 ("training sample"), and the second part can be used for the assessment of the loss of the vantage sets applied to independent images ("validation sample"). Because cross-validation has already been applied on the training sample, this principle is called "double cross-validation" by Mosteller and Tukey [6, pp. 36f.].

However, the images of the validation sample can be expected to be more efficient to be used 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 for the databases A_1, A_2 and A_3 by performing such a double cross-validation.

For this sake A_i, $i = 1, 2, 3$, was split randomly in two disjunct parts A_{i1} (training sample) and A_{i2} (validation sample). The sizes have been $|A_{11}| = |A_{12}| = 550$, $|A_{21}| = |A_{22}| = 50$, $|A_{31}| = 595$ and $|A_{32}| = 594$. Additionally, we restricted A_{21} and A_{22} so that both sets had to contain exactly 5 images from each of the 10 classes.

The strategies SFLOO, MAXIMIN, CLUSTER, CV and NCV[4] were again performed on A_{i1} to yield vantage sets V_{i1} independent of A_{i2}, and the loss of selecting images from A_{i1} for the new images from A_{i2} was measured by

$$L_{i2}(S_{V_{i1}}) = \frac{1}{|A_{i2}|} \sum_{a \in A_{i2}} l(a, S_{V_{i1}}(a, A_{i1})). \tag{8}$$

The absolute values of $L_{i2}(S_{V_{i1}})$ cannot estimate properly the loss of a vantage set V chosen with the same strategy on A_i, because $|A_{i1}| \approx |A_i|/2$, and therefore it can be expected that better matchers for new images can be found in A_i than in A_{i1}. But it is of interest if the ranking of strategies with respect to $L_{i2}(S_{V_{i1}})$ remains the same as for $L_{i1}(S_{V_{i1}})$ on A_{i1}, which is the analogue of $L(S_V)$ on A_i, and in particular if SFLOO remains to be the best. To keep the circumstances comparable, t for CV and NCV and p have been divided by 2 compared to the computations in Section 6. SFLOO-l_9 has been replaced by SFLOO-l_4 for A_2. R_{21} and R_{22} have been defined by analogy to L_{i1} and L_{i2} with the difference that $k = 5$ has been used in (7) because the class sizes in A_{21} are shrunken to 5.

The results of such a double cross-validation depend on chance because of the random split of the database. To get more reliable results, we repeated the double cross-validation 10 times for the databases A_1 and A_2. The results are given in Tables 5 and 6. While there has been considerable variation in the overall loss values during the 10 replications, the rankings among the strategies remained fairly stable. Note that the retrieval rate $R_{22}(S_{V_{21}})$ must be expected to be larger than $R_{21}(S_{V_{21}})$ in most cases, because the query object itself, which is always correctly classified, is included in the calculation of $R_{21}(S_{V_{21}})$, but not in $R_{22}(S_{V_{21}})$.

The results for A_3 are given in Table 7. Generally, the results show the expected tendency that SFLOO suffers from the largest selection bias, but still remains the best strategy. CV and NCV also try to optimize the overall loss approximately, but except of NCV in Table 7 they did not produce a significantly larger selection bias than MAXIMIN and CLUSTER.

[4] CV and NCV again have not been applied to A_2.

As opposed to Table 2, SFLOO-l_4 outperforms SFLOO-l_0 in terms of the retrieval rate in Table 6. This happened for 8 out of 10 random splits for $R_{21}(S_{V_{21}})$ and for 7 out of 10 random splits for $R_{22}(S_{V_{21}})$. With respect to the overall loss, MAXIMIN performed worse in Table 6 compared to the other situations. Because the first vantage object of MAXIMIN is chosen randomly, $p = 2$ may be too small for this strategy.

8. 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 loss measure based on the "leave-one-out" principle. Our suggestions for loss function are not meant to be optimal in a general sense. We think that the loss measure should be tailored to the concrete database and the objective of the image retrieval. However, our suggestions may fit many situations.

If n is not too large, it is promising to optimize the loss 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 unless the number of vantage objects was too small. It is fast and easily implemented, so that it looks like a good choice. The strategy of taking representative objects of clusters as vantage objects led to disappointing results in almost all examples. However, we generally recommend the comparison of vantage objects from more than one strategy.

The results of our examples do not show considerably different tendencies between the two image dissimilarity measures that obey the triangle inequality and the one of Latecki and Lakämper [7], but more experiments would be needed to examine in depth the dependence of our approach on the properties of the dissimilarity measure.

From a statistical point of view, the overall loss function (3) is only an estimate 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 estimate 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 in [13], where the author refers as well to theoretical results about some situations where the principle may lead to suboptimal decisions. If vantage objects 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. We suggest to do this only for the illustration of the bias, and to use the objects 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" [14] 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 loss measure.

9. Summary

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 $p \ll n$ "vantage objects" [1] 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 loss measure to assess the quality of a set of vantage objects: For every image, we select a matching image from the remaining images of the database by use of the vantage set, and we average the resulting dissimilarities. This principle is referred to as "leave-one-out cross-validation" is statistics. We suggest and compare some choice strategies: The first class of choice strategies is based on a stepwise forward selection of vantage objects to optimize the loss measure. This can be done by using the whole dataset to estimate the loss measure, or, to save computing time, to draw random subsets on which the measure is evaluated. The second class is the choice of objects as representative as possible for the whole range of the database. From this class, a cluster analysis method and the stepwise forward maximization of the minimum dissimilarity inside the vantage set are compared. Some modifications are suggested to adapt the loss function and the selection criterion for vantage objects better to a given database. We apply the strategies to four example databases. We study not only the performance of the strategies, but also the selection bias by splitting the databases in two parts and testing the quality vantage objects selected from one part for the other. In the examples, stepwise forward optimization of the loss measure on the whole database performs best. The strategy to maximize the minimum dissimilarity inside the vantage set shows a good relation between performance and computing time.

References

[1] J. Vleugels, R. Veltkamp, Efficient image retrieval through vantage objects, Pattern Recognition 35 (2002) 69–80.
[2] D.J. Hand, Discrimination and Classification, Wiley, Chichester, 1981.
[3] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: an Introduction to Cluster Analysis, Wiley, New York, 1990.
[4] M. Stone, Cross-validatory choice and assessment of statistical predictions, J. Roy. Stat. Soc. B 36 (1974) 111–133.
[5] P.A. Lachenbruch, M.R. Mickey, Estimation of error rates in discriminant analysis, Technometrics 10 (1968) 1–11.
[6] F. Mosteller, J.W. Tukey, Data Analysis and Regression, Addison-Wesley, Reading, MA, 1977.
[7] L.J. Latecki, R. Lakämper, Shape similarity measure based on correspondence of visual parts, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 1185–1190.
[8] E. Ronchetti, C. Field, W. Blanchard, Robust linear model selection by cross-validation, J. Amer. Stat. Assoc. 92 (1997) 1017–1023.
[9] P. Giannopoulos, R.C. Veltkamp, A pseudo-metric for weighted point sets, in: Proceedings European Conference on Computer Vision (ECCV 2002), Lecture Notes in Computer Science, Vol. 2352, Springer, Heidelberg, 2002, pp. 715–730.
[10] J. Hu, A. Mojsolovic, Optimal color composition matching of images, in: Proceedings of ICPR, Vol. 4, Barcelona, 2000, pp. 47–51.
[11] L.J. Latecki, R. Lakämper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, South Carolina, 2000, pp. 424–429.
[12] K.N. Berk, Comparing subset regression procedures, Technometrics 20 (1978) 1–6.
[13] M. Stone, Cross-validation: a review, Statistics (Mathematische Operationsforschung und Statistik) 9 (1978) 127–139.
[14] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, Chapman & Hall, New York, 1993.

About the Author—CHRISTIAN HENNIG received a diploma in Mathematics 1993 and a Ph. D. 1997, both at the University of Hamburg. He is research assistant at the University of Hamburg since 1997 and temporarily 2001–2003 at the Seminar for Statistics, ETH Zürich. His research areas are multivariate data analysis, especially classification and clustering, and philosophical foundations of statistics.

About the Author—LONGIN JAN LATECKI is the winner of the 25th Pattern Recognition Society Award together with Azriel Rosenfeld for the best paper in Pattern Recognition published in 1998. He received the main annual award from the German Society for Pattern Recognition (DAGM), the 2000 Olympus Award, for his contributions on Shape Description and Similarity of 2D Objects. He is an associated editor of Pattern Recognition and chairs the annual conference series Vision Geometry organized by the Society for Optical Engineering (SPIE). He is an associate professor at the Dept. of Computer and Information Sciences, Temple University, Philadelphia, USA. His main research areas with over 75 publications are shape representation and shape similarity, video mining, image and video databases, video compression and transmission over IP, and digital geometry and topology.