Non-Euclidean Problems in Pattern Recognition Related to Human Expert Knowledge

Robert P.W. Duin

Pattern Recognition Laboratory, Delft Univ. of Technology, Delft, The Netherlands
r.p.w.duin@ieee.org
http://prlab.tudelft.nl

Abstract. Regularities in the world are human defined. Patterns in the observed phenomena are there because we define and recognize them as such. Automatic pattern recognition tries to bridge human judgment with measurements made by artificial sensors. This is done in two steps: representation and generalization.

Traditional object representations in pattern recognition, like features and pixels, either neglect possibly significant aspects of the objects, or neglect their dependencies. We therefor reconsider human recognition and observe that it is based on our direct experience of dissimilarities between objects. Using these concepts, pattern recognition systems can be defined in a natural way by pairwise object comparisons. This results in the dissimilarity representation for pattern recognition.

An analysis of dissimilarity measures optimized for performance shows that they tend to be non-Euclidean. The Euclidean vector spaces, traditionally used in pattern recognition and machine learning may thereby be suboptimal. We will show this by some examples. Causes and consequences of non-Euclidean representations will be discussed. It is conjectured that human judgment of object differences result in non-Euclidean representations as object structure is taken into account.[1]

1 Introduction

Pattern recognition is an intrinsic human ability. Even young children are able to recognize patterns in the surrounding objects. During the whole life we are guided by pattern recognition. It constitutes implicitly a base for our judgements. Scientists make explicitly use of this ability in their professional life.

Science usually starts with a categorization of the phenomena. The differences between the various pattern classes are, at least initially, defined by the human observer based on his personal interest, e.g. following from the utility. Later they may explicitly be related to observable properties.

The research area of automatic pattern recognition studies the design of systems that are able to simulate this human ability. In some applications it is aimed to simulate an expert, e.g. a medical doctor, performing a recognition task. The design is based on an analysis of recognized examples and is guided by expert knowledge of the observations and of the procedures that are followed.

[1] This paper is an extended version of [1].

J. Filipe and J. Cordeiro (Eds.): ICEIS 2010, LNBIP 73, pp. 15–28, 2011.

In order to learn from examples it is necessary to represent them such that they can easily be compared. A statistical analysis of a set of examples (the training set) should be possible in order to pave the ground for an appropriate assignment of the pattern class to new examples. We distinguish in this process two steps:

Representation. In this first stage real world objects, observed by sensors, are represented such that the comparison with other objects is enabled. All available knowledge about the objects, their properties and the pattern classes to be distinguished should be used here.

Generalization. Using the representation, sets of objects (classes) or discriminant functions between them are modeled from their statistics. This is based on statistical estimators and machine learning procedures. The goal is to create the models in such a way that the assignment of class membership of new, incoming objects is facilitated (classification).

In the first step the emphasis is on the use of existing knowledge. In the second step 'new' knowledge is generated from observations (learning from examples). Occasionally it happens as well that the representation is optimized by observations and that additional knowledge is used during statistical modelling.

It is the purpose of this paper to discuss conditions and problems in obtaining good representations. It will be shown that proper representations, in agreement with human observations and judgements, may be in conflict with the demands for the next step, the generalization. In some problems (and perhaps in many) such a proper representation is non-Euclidean, but the present set of generalization tools is based on the assumption of an Euclidean space. Examples will be discussed as well as possibilities to solve this problem.

2 The Representation Problem

The purpose of the representation step is that it should enable use to compare sets or classes of objects in a numerical way. It should be possible to build models for such a class or to construct decision functions between classes. The dominant, favorite way in pattern recognition is the vector space. Objects are represented as points in such a space and operations on sets of points (classes of objects) result in functions that describe domains for the classes or separation functions between them.

The multi-dimensional vector space representation enables the application of many tools as developed in linear algebra, multi-variate statistics and machine learning. A point of concern however is whether it pays respect to the objects and their relations. If we want to learn from the representation of a set of objects that is given to us as examples for the recognition (classification) of new objects then a demand is that a small variation in one of the example objects should result in a small variation in its representation. If this is not the case, if the representation jumps in space, how can we expect that we learn from the set of examples in terms of the construction of class domains or separation boundaries?

This demand, the continuity of the variation in the representation as a result of a variation in the object, is directly related to what is called in the early literature on pattern recognition as *compactness*: classes of similar objects cover a finite domain in the representation space. We reformulate the demand as: similar real world objects should have similar representations.

Are similar representations thereby also related to similar objects? Not necessarily. If two-dimensional objects like hand-written characters are represented by area and perimeter then the representation is compact: small changes in the shape of a character will result in small changes in area and perimeter. Objects with entirely different shapes however may have the same area and perimeter and thereby the same representation. An additional demand for representations, usually not fulfilled, is that similar representations should refer to similar objects. If this is the case, the representation is a *true representation*.

If the representation is not true entirely different objects may be close in the representation space. They may even belong to different classes. This is the cause of class overlap. Given the representation classes may not be fully separated anymore. In spite of the fact that an expert observes essential differences and assigns them to different classes, they may be represented on the same place in the representation space. This can only be solved by statistics: in this area objects should be assigned to the most probable pattern class. Consequently, it is needed to use statistics as probability densities have to be estimated.

We like to emphasize that the need of using statistics in pattern recognition is caused by class overlap resulting from a non-true representation. If the representation would have been a true representation then class differences observed by a human expert would have been reflected in the representation and objects of different classes would not have been represented on the same place. The intrinsic amount of class overlap, in pattern recognition called the Bayes error, is the result of the representation. A different representation will yield a different Bayes error. A true representation will result in a zero Bayes error.[2]

3 Feature Representation

For a long time the feature representation has been the only vector representation used in pattern recognition. It is still dominant and the vector spaces resulting from other representations are often even called 'feature spaces', neglecting their different origin.

Features are object properties that contribute to the distinction of classes. They are defined or suggested by the experts that are also able to determine the true class membership (class label) of objects. For many problems it appears to be difficult to define exactly what the feature are. For instance, doctors cannot always exactly defined what should be measured in a lung X-ray or in a ECG signal for the recognition of some disease. Also in daily life it is for humans not easy to describe explicitly how to recognize a particular person.

If an expert has good knowledge about the physical background of a pattern class he may well be able to define a small set of powerful features that can be used to construct

[2] In this reasoning we neglect here the fact that some objects are ambiguous and can belong to more than a single class, e.g. the digit '0' and the letter 'O' in some fonts. We also assumed that the class labels are assigned without any noise.

a well performing recognition system. If he is hesitative however he may supply long lists of possible measurements that might be used as features. Obtaining many features is the result of a lack of knowledge. This should be compensated by many, well labeled examples to be used by the pattern recognition analyst to train a classification system and possibly to determine a small set of good features.

In application areas where many features have been proposed, e.g. in OCR, optical character recognition, this is the result of a lack of knowledge. We don't know, in the sense that we cannot make it explicit, how we recognize characters. Such applications become only successful if large amounts of data become available to compensate this lack of knowledge.

4 Pixel Representation

If good features cannot be found for objects like images, time signals and spectra, an obvious alternative is to take everything: just to sample the object. For images these are the pixels and we will use that word for the resulting representation: the pixel representation. It has the advantage that it still contains everything, seemingly no information is lost (see below for a discussion), but it is not specific. Many pixels may be needed to generate a good result.

In the above mentioned OCR application area, a break-through was established when pixel representations became possible due to the availability of large datasets and big and fast computers to handle them. OCR systems are usually based on a combination of many approaches, including pixel based ones.

There is a paradox related to this development. High resolution images yield high dimensional vector spaces resulting from the pixel representation. To build classification systems for such spaces many examples (large training sets) are needed. For a given, limited size of the training set, it may be better (yielding a higher performance) to reduce the dimensionality by taking less pixels, e.g. by sub-sampling the images. This is entirely different from the human recognition. It is certainly not true that human recognition is improved by the use of low-resolution images. This points to a possible defect of this whole approach: the representation and/or the classification schemes used in it are not appropriate.

What is definitely wrong with the pixel representation is that the pixel connectivity, the relations between neighboring pixels in the image, is lost. From the representation it cannot be retrieved anymore which axes that participate in constituting the space corresponding to neighboring pixels. We have cut the objects in pieces, have put them on a heap and we try now to use this heap for recognizing the object. In other words, we have lost ourselves in many minor details and the sight on the entire object is completely gone. This has already been observed and discussed extensively by Goldfarb [2].

5 Structural Representations

An approach to pattern recognition that definitely respects that objects should be considered in their entirety and that it takes into account that it is dangerous to break them down into unrelated sets of properties is structural pattern recognition. Unfortunately, this approach does not produce a vector space, but represents objects by strings or graphs.

Generalization from sets of strings or graphs has been done for a long time by template matching. E.g. a dissimilarity measure between graphs is defined and by the resulting graph match procedure new objects are classified to the class of the object with the most similar graph. Much work has been done on the improvement of the representation as well as on the matching procedure. Classification itself relied for a long time just on template matching, corresponding to the nearest neighbor rule in statistical pattern recognition.

6 Dissimilarity Representation

Between the above representations clearly a gap can be observed. For vector spaces very nice sets of tools are available to describe class domains or to construct classification functions. The feature and pixel representations however that apply such vector spaces suffer from the fact that they describe the objects just partially, resulting in strong class overlap, or cut them entirely in pieces by which their structure is lost. The structural representations respect object structure but fail to construct a good representation space for which a broad collection of tools is available.

The dissimilarity representation [3] tries to bridge this gap. It is based on the observation that a comparison of objects is basic in the constitution of classes in human recognition [4] In the constitution of a dissimilarity representation on top of structural pattern recognition pairwise dissimilarities between objects are found by matching structural object descriptions. They are used to construct a vector space in which every object is represented as a point. Instead of template matching now classifiers in such vector spaces can be considered. The two main approaches to construct a vector space from a given set dissimilarities, the dissimilarity matrix, will be shortly treated. There are many references that describe these in mathematical terms, e.g. [3],[5].

6.1 The Dissimilarity Space

In the first approach the dissimilarity matrix is considered as a set of row vectors, one for every object. They represent the objects in a vector space constructed by the dissimilarities to the other objects. Usually, this vector space is treated as a Euclidean space.

If there are m objects given to construct the space, then each of them is given by m dissimilarities (including the dissimilarity with itself, usually zero). The initial dissimilarity space is thereby given as a m-dimensional vector space with m objects in it. This is a degenerate situation in which many classifiers yield bad results due to overtraining or the the curse of dimensionality [6]. Some classifiers like the SVM can still produce good results in this situation, but for many it may be better either to fill the space with more objects, or to reduce the dimensionality, e.g. by some procedure for prototype selection or feature selection (which coincides here). Even random selection of objects works well as nearby objects are similar and a random selection produces some sampling of total set.

The result is a vector space built by a so called representation set of objects and which is filled by an appropriate training set. The standard tools of statistical pattern recognition can be used to construct classifiers. New objects are mapped into the space by just measuring their dissimilarities to the representation set.

It should be realized that the Euclidean distances between objects in the dissimilarity space are only in very special cases identical to the given dissimilarities. In general they are different. However, it is expected that almost identical object have very similar dissimilarities to all representation objects, so they will be very close in the dissimilarity space and have thereby a small distance. Consequently the dissimilarity representation is compact. If the dissimilarity measure that is used is appropriate then the reverse is also true: different objects will have different dissimilarities to the representation objects under the condition that this set is sufficiently large and well distributed over the domain of objects. The dissimilarity representation has thereby the potential to be a true representation.

6.2 Embedding the Dissimilarity Matrix

In the second approach, an attempt is made to embed the dissimilarity matrix in a Euclidean vector space such that the distances between the objects in this space are equal to the given dissimilarities. This can only be realized error free, of course, if the original set of dissimilarities are Euclidean themselves. If this is not the case, either an approximate procedure has to be followed or the objects should be embedded into a non-Euclidean vector space. This is a space in which the standard inner product definition and the related distance measure are changed (among others, resulting in indefinite kernels). It appears that an exact embedding is possible for every symmetric dissimilarity matrix with zeros on the diagonal. The resulting space is the so-called pseudo-Euclidean space [3].

The pseudo-Euclidean space consist of two orthogonal subspaces, a 'positive' subspace and a 'negative' subspace. Every object has a representation in both subspaces. Both subspaces are normal Euclidean spaces. The squared distance between two objects represented in the pseudo-Euclidean space has to be determined by subtracting the squared distances between their representations in the two subspaces instead of adding them is in a ordinary Euclidean space. The negative subspace can be considered as a correction of the given dissimilarities w.r.t. proper Euclidean distances.

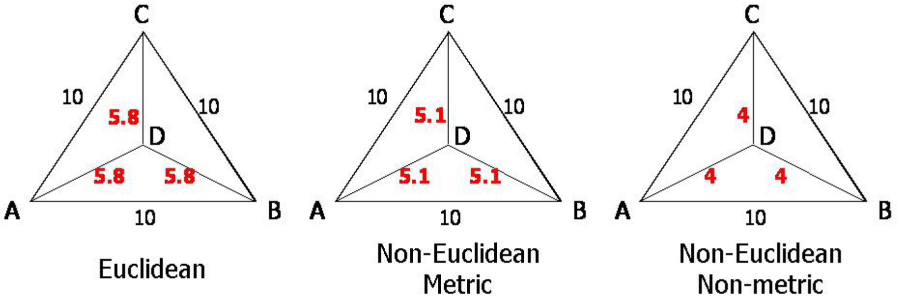
Many of the dissimilarity measures used in the pattern recognition practice appear to be indefinite: they cannot be understood as distances in a Euclidean vector space, they are sometimes even not metric and they do not satisfy the Mercer conditions that are needed for optimizing the SVM classifier [7].

A small but growing number of classifiers can be trained in the pseudo-Euclidean space [8], but a general toolbox is not yet available. For this reason and others Euclidean corrections are studied: ways to transform the given dissimilarity matrix or the pseudo-Euclidean embedding in such a way that an Euclidean vector space is constructed that is as close as possible to the original one. This is useful if the cause of the non-Euclidean characteristic of the data is non-informative, i.e. that it is unrelated to the class differences. Measurement noise and approximate optimizations in determining the dissimilarities may result in non-Euclidean relations between objects. Such noise may be removed by Euclidean corrections. In case however the non-Euclidean characteristics are informative Euclidean corrections will deteriorate the performance.

In the present state-of-the-art the dissimilarity space has to be preferred over embedding combined with corrections. It is from a computational point much more feasible and it does not suffer from non-Euclidean problem. The dissimilarity space however, treats dissimilarities as properties and neglects their distance character. For that reason research into embedding approaches continuous from the perspective that may preserve better the information contained in the dissimilarity measurements.

7 Non-Euclidean Human Pattern Recognition

In [1] we extensively studied the causes of the non-Euclidean characteristics of many real world datasets. We will summarize some results here and then discuss this topic from a slightly shifted point of view in order to gather support for our main conjecture.



Euclidean

Non-Euclidean Metric

Non-Euclidean Non-metric

Fig. 1. Illustration of the difference between Euclidean, metric, but non-Euclidean and non-metric dissimilarities. If the distances between the four points A, B, C and D are given as in the left plot then an exact 2-dimensional Euclidean embedding is possible. If the distances are as given as in the middle plot, the triangle inequality is obeyed. So the given distances are metric. but no isometric Euclidean embedding exist. The distances in the right plot are non-Euclidean as well as non-metric.

In fig. 1 the difference between non-metric and non-Euclidean distances is illustrated. Distances can be metric and still non-Euclidean. Non-metric relations constitute a strong example of non-Euclidean relations, but if the distances are metric it is still possible that the distances between more than three points do not fit in a Euclidean space. In fact this is common. In many applications the analyst defines a distance measure that is metric while he demands that the direct distance between objects is always smaller than any detour[3].

It is not always possible to avoid non-metric relations. Suppose we have to define a dissimilarity measure between real world objects like (images of) cups. They may

[3] For local consistency we used in this example everywhere the word ’distance’ instead of dissimilarity. On other place again ’dissimilarity’ will be used to emphasized that we are discussing distance-like relations that are possibly sloppy defined.

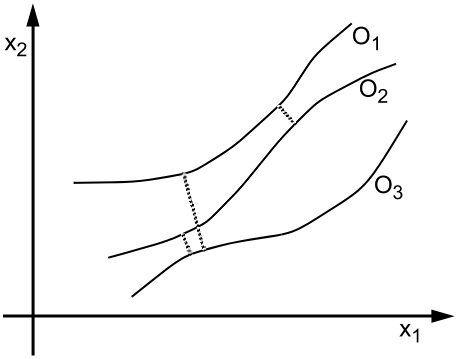


Fig. 2. Vector space with the invariant trajectories for three objects O_1, O_2 and O_3. If the chosen dissimilarity measure is the minimal distance between these trajectories, triangle inequality can easily be violated, i.e. $d(O_1, O_2) + d(O_1, O_3) < d(O_1, O_3)$.

be observed from different orientations, having different sizes that should not result in contributions to the dissimilarity as they are invariants for the class memberships. So in a pairwise comparison transformations for all orientations and sizes are considered and the smallest dissimilarity that is found is defined as the correct one, made insensitive for the invariants. In other pairwise comparisons this process is repeated for other pairs of cups. Observed in some high-dimensional parameter space a situation as sketched in fig. 2 may exist, showing that in case the transformations for removing the invariants are non-linear the triangle inequality may be violated.

Another example is given in fig. 3 which illustrates how an artificial dataset has been generated that we used for studying non-Euclidean data. In a multi-dimensional cube two sets of non-overlapping balls are positioned at random. The balls in the two sets have different radii. Their values are assumed to be unknown For every ball all distances to all other balls are measured from surface to surface. We asked ourselves the question whether it is possible to distinguish the two sets, e.g. can we determine whether an arbitrary ball belongs to the class of large balls or to the class of small balls if we just measure the distances as defined and if the labels of all other balls are given. This appears to be possible by making use of the negative part of the pseudo-Euclidean space. Without it, it is impossible. The surprising result was that if the positive part is neglected and just the negative part is given the separation is even much better.

This example makes clear how we may interpret the negative subspace of the pseudo-Euclidean space. If all balls would have had zero radii then we just had a collection of proper Euclidean distances. Because the balls have a size the given distances are somewhat shorter. A small value is missing and as a result the negative subspace is needed as a compensation. To phrase it somewhat poetic: as the objects have an inner life that cannot be observed directly, but that influences the measured dissimilarities, we end up with non-Euclidean data.

Let us now return to recognition problems for which features are difficult to define, like characters and medical images. Euclidean distances can be defined for such objects, e.g. by putting them on top of each other and adding the squared differences pixel

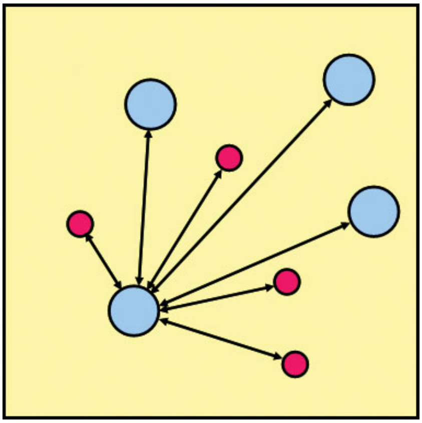


Fig. 3. Illustration of an artificial experiment in which sets of balls with different radii are distinguished by the distances between their surfaces

by pixel. Researchers trying to improve this create different dissimilarity measures, e.g. by a non-linear deformation of the pixel grid, see [9]. They thereby try to simulate the human way of observing objects implicitly, as they aim to improve the performance of the automatic recognition system such that it approximates the human recognition. This is done by deviating from the Euclidean distance measure.

There are many examples in the literature of non-Euclidean dissimilarity measures [3]. In particular in relation with shape recognition the dissimilarity approach using such measures produces good results. This brings us to the following conjecture:
The way humans judge differences between real world objects is non-Euclidean. This is caused by the fact that they include object structure next to object features in their judgement.

The above mentioned 'inner life' of objects is thereby identified as structure.

8 Examples

The differences between the representations discussed in this paper will be illustrated by two experiments. They are based on real world datasets. It is not our intention to claim that a particular approach outperforms other ones for these applications. A much more extensive set of experiments and discussions would be needed for that. Here we will restrict ourselves to show the performances of a rather arbitrary chosen classifier (LIBSVM with a linear kernel, [10]), which is generally recognized as very good, for the various representations. Other classifiers and other parameter settings may show different results.

8.1 Handwritten Digits

We took a part of the classic NIST database of handwritten numbers [11] and selected at random subsets of 500 digits for the ten classes 0-9. They were resampled to images of

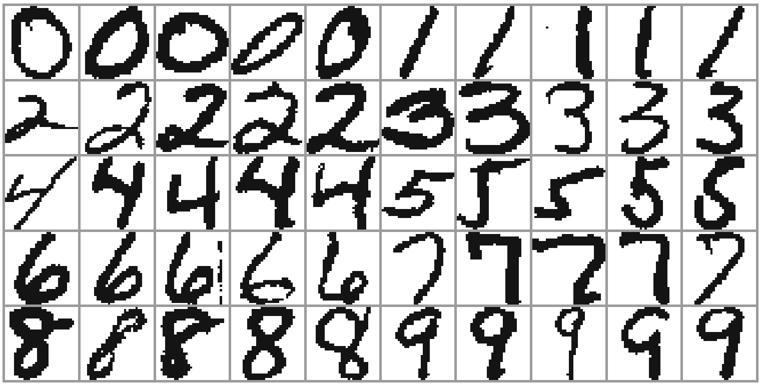


Fig. 4. Examples of the images used for the digit recognition experiment

32×32 pixels in such a way that the digits fit either horizontally or vertically. In figure 4 some examples are given: black is '1' and white is '0'. The dataset was repeatedly split in sets for training and testing. In every split the ten classes were evenly represented. The following representations are used:

Features. We used 10 moments: the 7 rotations invariant moments and the moments [0 0], [0 1], [1 0], measuring the total number of black pixels and the centers of gravity in the horizontal and vertical directions.

Pixels. Every digit is represented by a vector in $32 \times 32 = 1024$ dimensional vector space.

Dissimilarities to the training set. Every object is represented by the Euclidean distances to all objects in the training set.

Dissimilarities to blurred digits in the training set. As the pixels in the digit images are spatially connected blurring may emphasize this. In this way the distances between slightly rotated, shifted or locally transformed but otherwise identical digits becomes small.

The results are shown in figure 5. They show that for large training sets the pixel representation is superior. This is to be expected as this representation stores asymptotically the unverse of possible digits. For small training sets a proper set of features may perform better. The moments we used here are very general features. Much better ones may be found for describing digits. As explained a feature description reduces the objects: it may be insensitive for some object differences. The dissimilarity representation for sufficiently large representation sets may see all object differences and may thereby perform better.

8.2 Flow Cytometer Histograms

This dataset is based on 612 FL3-A DNA flow cytometer histograms from breast cancer tissues in 256 resolution. The initial data were acquired by M. Nap and N. van Rodijnen of the Atrium Medical Center in Heerlen, The Netherlands, during 2000-2004, using

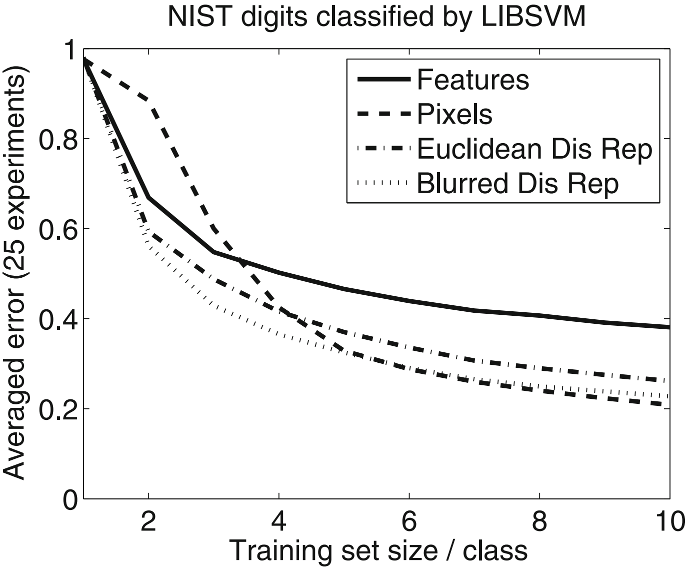


Fig. 5. Learning curves for the digit recognition experiment

the four tubes 3-6 of a DACO Galaxy flow cytometer. Histograms are labeled in 3 classes: aneuploid (335 patients), diploid (131) and tetraploid (146). We averaged the histograms of the four tubes thereby covering the DNA contents of about 80000 cells per patient. We removed the first and the last bin of every histogram as here outliers are collected, thereby obtaining 254 bins per histogram.

Examples of histograms are shown in fig. 6. The following representations are used:

Histograms. Objects (patients) are represented by the normalized values of the histograms (summed to one) described by a 254 dimensional vector. This representation is similar as the pixel representation used for images as it is based on just a sampling of the measurements.

Euclidean distances. These dissimilarities are computed as the Euclidean distances (L2 norm) in the above mentioned vector space. Every object is represented by its distances to the objects in the training set.

Calibrated distances. As the histograms may suffer from an incorrect calibration in the horizontal direction (DNA content) we computed for every pairwise dissimilarity between two histograms the multiplicative correction factor for the bin positions that minimizes their dissimilarity. Here we used the L1 norm. This representation makes use of the shape structure of the histograms and removes an invariant (the wrong original calibration) as symbolically illustrated in figure 2.

Again a linear Support Vector Machine was used as a classifier using a fixed trade-off parameter C. The learning curves for the three representations are shown in figure 7. They clearly illustrate how for this classifier the dissimilarity representation outperforms the 'pixel' representation (sampling of the histograms) and that using background knowledge on the definition of the dissimilarity measure improves the results further.

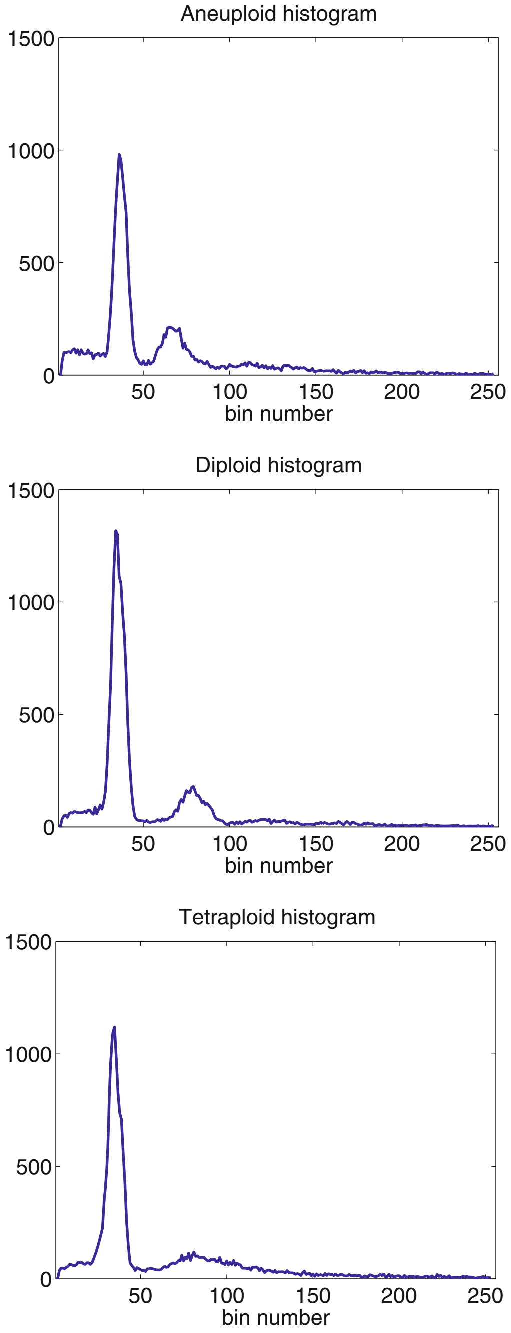


Fig. 6. Examples of some flow cytometer histograms: aneuploid (top), diploid(middle) and tetraploid (bottom)

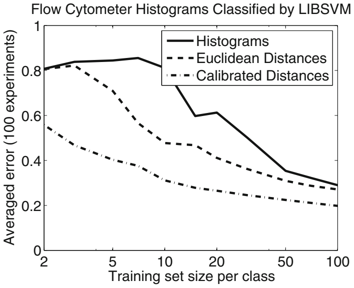


Fig. 7. Learning curves for the flow cytometer histogram recognition experiment

9 Discussion and Conclusions

For the recognition of real world objects measured by images, time signals and spectra, simple features or samples may not be sufficient. They neglect the internal structure of objects. Structural descriptions like graphs and strings lack the possibility of the use of an appropriate vector space. The dissimilarity representation bridges this gap, but has thereby to be able to deal with non-Euclidean dissimilarities. We conjecture that this deviation from the Euclidean distance measure is caused by the inclusion of structure in the human judgement of object differences which is lacking in the traditional feature representations.

Acknowledgements. This research is financially supported by the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

1. Duin, R.P.W., Pęalska, E.: Non-euclidean dissimilarities: Causes and informativeness. In: Hancock, E.R., Wilson, R.C., Windeatt, T., Ulusoy, I., Escolano, F. (eds.) SSPR&SPR 2010. LNCS, vol. 6218, pp. 324–333. Springer, Heidelberg (2010)
2. Goldfarb, L., Abela, J., Bhavsar, V., Kamat, V.: Can a vector space based learning model discover inductive class generalization in a symbolic environment? Pattern Recognition Letters 16(7), 719–726 (1995)
3. Pęalska, E., Duin, R.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore (2005)
4. Edelman, S.: Representation and Recognition in Vision. MIT Press, Cambridge (1999)
5. Pęalska, E., Duin, R.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38(6), 729–744 (2008)
6. Jain, A.K., Chandrasekaran, B.: Dimensionality and sample size considerations in pattern recognition practice. In: Krishnaiah, P.R., Kanal, L.N. (eds.) Handbook of Statistics, vol. 2, pp. 835–855. North-Holland, Amsterdam (1987)
7. Cristianini, N., Shawe-Taylor, J.: Support Vector Machines and other kernel-based learning methods. Cambridge University Press, UK (2000)

8. Pęalska, E., Haasdonk, B.: Kernel discriminant analysis with positive definite and indefinite kernels. IEEE Transactions on Pattern Analysis and Machine Intelligence 31(6), 1017–1032 (2009)
9. Jain, A., Zongker, D.: Representation and recognition of handwritten digits using deformable templates. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(12), 1386–1391 (1997)
10. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using second order information for training support vector machines. Journal of Machine Learning Research 6, 1889–1918 (2005)
11. Wilson, C., Garris, M.: Handprinted character database 3. Technical Report, National Institute of Standards and Technology (February 1992)