THE ROLE OF SEQUENCES FOR INCREMENTAL LEARNING

Susanne Wenzel
Institute of Geodesy and Geoinformation, Department of Photogrammetry, University of Bonn, Germany
susanne.wenzel@uni-bonn.de

Lothar Hotz
HITeC e.V. c/o Department of Computer Science, University of Hamburg, Germany
hotz@informatik.uni-hamburg.de

Keywords: Incremental Learning, Ordering Effects, Version Space Learning, Scene Interpretation

Abstract: In this paper, we point out the role of sequences of samples for training an incremental learning method. We define characteristics of incremental learning methods to describe the influence of sample ordering on the performance of a learned model. We show the influence of sequence for two different types of incremental learning. One is aimed on learning structural models, the other on learning models to discriminate object classes. In both cases, we show the possibility to find good sequences before starting the training.

1 INTRODUCTION

For the purpose of scene interpretation, e. g. the interpretation of facade images it is impossible to capture the huge variability of facades with one dataset. Instead we need a continuous learning system that is able to improve already learned models using new examples. This process in principle does not stop. For this intention of infinitely long learning, there exists a number of incremental learning methods. But one can easily show that the performance of these methods depends on the order of examples for training, also this is not mentioned by most other authors.

Imagine little kids, they are sophisticated incremental learners. They permanently increase their knowledge just by observing their surrounding. But obviously they learn much better when examples are presented in a meaningful order.

Therefore teachers design a curriculum for teaching their pupils in the beginning of a school year. This way they ensure that the topics are presented to the pupils in a well structured order depending of the complexity and challenge of each single topic.

In this paper, we want to point out the relevance of sequence for training incremental learning methods. Furthermore, we want to show the possibility to define curricula for training incremental learning methods. That means, we want to define good sequences of examples to train such methods such that they always perform best.

We show this by means of two different learning methods. One is aimed on learning models to discriminative object classes. The other is based on Version Space Learning and aims on structural learning.

Existing work about ordering effects in incremental learning is quite rare. A good overview of existing work and results in this field is given by (Langley, 1995).

(Fisher, 1987) proposed with COBWEB an unsupervised incremental learning method based on hierarchical conceptual clustering. He already mentioned the role of sequence for training.

In the following some other authors try to overcome this problem. (Talavera and Roure, 1998) introduced a buffering strategy, where they store difficult examples for later evaluation. Thus, they postpone their evaluation to a more qualified state of the learned model. But this can lead to an oversized sample storage. This means just collecting all examples and evaluating them in a batch procedure. (McKusick and Langley, 1991) made some empirical experiments on ACHACHNE, a modification of COBWEB. For this algorithm they got good results by choosing examples altering from all classes instead of a sequence of samples from single classes.

Recently, the term curriculum learning was used by (Bengio et al., 2009) for finding sequences for training neural networks.

The rest of the paper is structured as follows. We first characterize incremental learning methods regarding the effects of sample ordering in Section 2. Thereafter, we start description of ordering effects on learning models to discriminate object classes in Section 3. Here, we propose experiments to explore these effects and propose a method to define suitable class sequences for training based on the estimate of bounds of the Bayes error in Section 3.2. Section 4 argues on the role of sequences for learning structural models.

2 CHARACTERIZING INCREMENTAL LEARNING METHODS

To characterize a certain learning method we need to provide a measure for success. In this work, we measure performance in terms of the error rate related to number of samples used for learning so far to characterize the performance of an incremental learning method. This depend on various conditions, e. g. the classes, the samples, the classifier, and possibly the sequence of samples.

A further measure used in this work is the number of examples k that are used by a learning method for correctly interpreting a fixed number n of given examples. The goal would be to minimize the number of needed examples k through selecting a best sequence of examples.

As main characteristic of an incremental learning method we define its dependence on the training sequence as follows:

Definition 2.1: ***Ordering sensitivity*.** We call an incremental learning method ordering sensitive, if its performance is effected by the sequence of training examples.

In other words if there exist a training sequence that yields a different curve of error rates or a different number of examples than the method is ordering sensitive.

Now, one might tend to use this to decide whether a learning method is good or not. Thus, one could say a good incremental learning method is not effected by the training sequence, thus, is not ordering sensitive (Giraud-Carrier, 2000).

But we argue vice versa. If a method is ordering sensitive and we can assume that we are able to identify the best training sequence, thus, getting always the best performance for this method, then we possibly can outperform any method that is not ordering sensitive.

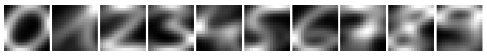
Figure 1: Mean images per class of samples from used digits dataset.

3 THE ROLE OF SEQUENCES WHEN LEARNING OBJECT CLASSES

In this section, the goal of learning is identifying and discriminating instances of classes.

In the following, we will concentrate on the qualitative characteristics of an incremental learning method, thus, the classification power measured in terms of error rates.

To measure ordering sensitivity, we just train a method with several sequences of training samples and record the error rates per used sample. The variance of these error rates indicates the ordering sensitivity. The higher the variance of error rates the higher the ordering sensitivity.

We need to distinguish to different training scenarios.

1. Over the whole training sequence there are samples from all classes simultaneously available.
2. Samples from different classes become available sequentially, one class after the other.

In the first case, we ask whether there is an influence of the training sequence at all, thus, over all samples from all classes. Here, one can imagine either to start with some very typical examples from each class and increase their complexity afterwards. Or vice versa start with complex examples to rapidly fill the feature space. Obviously the best strategy strongly depends on the used method.

For the second case, we assume the possibility to learn one class after the other, i. e. all about number windows followed by all about doors and so on. The question is, which class sequence is suited to learn most about facade elements. We call ordering sensitivity in the 2nd scenario class ordering sensitivity.

3.1 Experiments to explore ordering effects

To investigate possible ordering effects we have used the dataset of handwritten digits from (Seewald, 2005). These are handwritten numbers form 0 to 9, thus, 10 classes. That are intensity images, normalized to 16×16 pixel. Per class there are 190 samples for training and 180 for testing, thus, 3700 images in total. Figure 1 shows mean images for each class

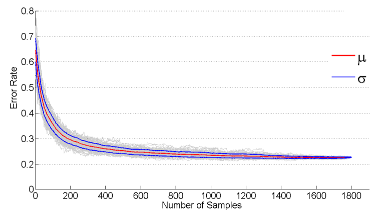

Figure 2: Ordering Sensitivity. Results of experiments with 100 different training sequences with samples randomly chosen from all classes. The graph shows error rates vs. number of used samples. Gray: error rates of each individual sequence, red: mean error rate of the according experiment, blue: standard deviation of the according experiment.

of these dataset. As features, we use HOG-Features (Dalal and Triggs, 2005).

As learning method we use an incremental formulation of Linear Discriminant Analysis (LDA), called iLDAaPCA as it is described by (Uray et al., 2007). This is a combination of two linear subspace methods, an incremental Principal Component Analysis (PCA) and the LDA, thus, uses the generative power of PCA and the discriminative power of LDA. Whereby the use of PCA enables the incremental update of the LDA space.

Experiment 1: Sequences including samples from all classes First, we need to know, if an incremental learning method is ordering sensitive. Therefore, we just try different sequences of training samples. For each new sample, we update the learned model, evaluate it on the test set and record the error rate. This way we get a curve of error rates depending on the number of used samples. The variance of these error rates over the whole sequence indicates the sensitivity against the training sequence. Results are shown in Figure 2.

The variances of gray curves, which are the individual curves of error rates, show the great sensitivity with respect to the ordering of training samples.

We observe a decrease of variances down to equal results of 23% error rate for all trials after processing all samples.

Experiment 2: Special ordering of classes The second type of experiments tries to evaluate the dependency of ordering the classes during training. Assume training samples of different classes become available over time, thus, we start training with some initial classes and add samples of a new class afterwards. Hence, the experiment is defined with first choosing the classes randomly and second choose the order of samples of this class again randomly. Results are shown in Figure 3.

Here, we started sequential training with samples

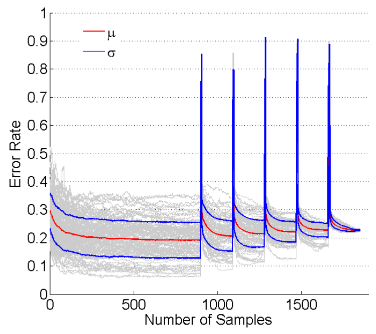

Figure 3: Class ordering sensitivity for 100 different orderings of learned classes. Same meaning of axes and lines as in Figure 2. Here we started with samples from 5 randomly chosen classes and continued with samples from new classes afterwards.

from 5 randomly chosen classes out of 10. Then, we added samples from single again randomly chosen new classes. At this point, the error rates increase drastically. This is caused by the fact that one new sample from a new class is not sufficient to train the model on this new class, thus, the evaluation on the test set fails on this class. But by adding more samples from the new class the model becomes better and the error rates decreases again. The overall performance decreases a bit with every new class as the complexity of the classification problem increases with the number of classes.

Again, we observe a high variability of error rates between different trials with different class sequences. During the initialisation this goes up to 30%. Again, we reach same results for each trial after processing all samples.

Hence, the result of this experiment shows a strong relation between class sequence and the performance of the learned model. Furthermore, it might be possible to use less samples when using the best sequence of classes, as the best possible error rate is already reached after using just a part of available samples.

Of course, at the end we get same results independent of the used training sequence. It tells: this method behaves well, thus, we have learned the same model after processing all available training samples. But on the other hand, with the scenario of life long learning we will never reach that end, the point, were we have seen all possible examples. Hence, the results of this experiments show the significance of having a curriculum, i. e. we need to know the best order of examples before starting the training. This way we would have an error rate always on bottom of the graph in Figure 3, hence, always reach best possible performance.

We are now able to characterize an incremental learning method in the sense of ordering sensitivity regarding samples from all classes or regarding the order of classes. However, by now we need to know how to define the best sequence for training. The following section proposes one criterion to choose a good sequence of classes just depending on the data.

Up to now, we have not found a similar criterion to get best suited within class sequences.

3.2 How to define a curriculum?

The task now is to find from all permutations of examples the one that produces the best performance. Related to the property of class ordering sensitivity of an incremental learning method defined in Section 2 we propose here a way to get good training sequences regarding this property.

For this purpose, we considered different measures for separating classes, e. g. distance measures as the Kullback-Leibler-distance or the Bhattacharyya distance. But both assume knowledge about the underlying class generating distributions. We do not want to make any assumptions about that. Hence, we looked for criteria that are independent from any assumption about the data distribution.

For this, we found the Bayes error as a general measure for the separability of classes that can be estimated without any knowledge about the data distribution. Thus, it defines the best achievable error rate in a classification procedure. We can estimate bounds of the Bayes error using the k-nearest-neighbour classificator. This is shown by (Fukunaga, 1972, chap. 6).

We did some experiments that we do not report in detail, at which we compared the error rates of different class sequences with the according bounds of Bayes error. Our empirical results indicate a strong relation between the Bayes error and the performance of an incremental learning method given a particular training sequence. We got good results in terms of error rates using the following two rules:

- Choose class combinations for initialisation with lowest bounds of Bayes error.
- Choose remaining classes with lowest bounds of Bayes error regarding the complete set of learned classes up to now.

In order to estimate good class sequences according this rules, which are just based on the data, we have used a greedy search procedure. Thereby we estimate bounds of Bayes error for all possible combinations of classes and search for sequences with the overall best performance. For description of the whole procedure see (Wenzel and Förstner, 2009).

The resulting list $\mathcal{L}$ consists of pairs of sets $\{\text{initialisation classes}, \text{single classes to add}\}$.

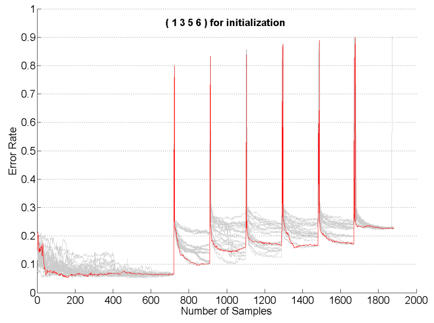

Figure 4: Results of testing predefined class ordering. Experiment with a fixed set of classes for initialisation.

Thereby the number of initialisation classes increases from 2 to any number M of classes that is still suited to initialise the learning method.

Experiment 3: Confirm best class sequence of classes determined from bounds of Bayes error We used the greedy search to find good sequences for our test dataset of handwritten digits. This results in lists of good class sequences which are shown in detail in (Wenzel and Förstner, 2009).

To verify this way of getting good class sequences we show the results of one selected experiment, shown in Figure 4. Primarily the experiment is defined as described in Section 3.1. Again, we trained the learning method with different randomly chosen sequences of classes. We started training with a fixed set of classes based on the results of searching for good sequence. Here we used classes $1,3,5,6$ for initialisation. In addition to the randomly chosen sequences afterwards we did an additional trial where we continued with classes $4,2,9,7,8,10$, which was one of 11 found suited sequences. The error rates of this trial is shown as red line in Figure 4. Again, more results, for other sequences found by greedy search can be found in (Wenzel and Förstner, 2009).

In fact, we get always almost best performance for classes sequences given by the bounds of Bayes error.

First, we observe that error rates from all trials during the initialisation are on the bottom area of those from Experiment 2, shown in Figure 3.

Second, after initialisation, thus, when adding samples from new classes, error rates from previous defined class sequences are always almost below error rates from random trials, shown as gray lines.

Hence, with the estimate of bounds of the Bayes error we actually have found an appropriate indicator for finding good sequences of classes for training an incremental learning method.

4 COMPUTING THE BEST IMAGE SEQUENCE

For the next experiment, we use a Version Space Learning (VSL) module for exploring good sequences of examples. These experiments are processed in the domain of interpreting facade scenes (Foerstner, 2007). By giving annotated primitive objects like door, railing, and window the interpretation system will combine those to aggregates like balconies or entrances. Models describing spatial relations, number, type, and size of parts of aggregates are learned through the VSL method.

Learning method The VSL module generates a set of possible concept hypotheses for positive examples of a given aggregate (see (Mitchell, 1977)). A *concept hypothesis* represents a possible conceptual description of real-world aggregates presented as learning examples. By increasing the example set, a concept hypothesis might change. In VSL, the space of possible concept hypotheses VS is implicitly represented through an upper and a lower bound on their generality. The General Boundary GB contains all maximally general members of VS, the Specific Boundary SB contains all maximally specific members of VS.

Experiment For analysing the influence of the order of presented examples on the number of needed examples, we processed the following experiment. Given a set S of 8 images, where each has a number of examples of the aggregate `balcony` (see Figure 5). Each sequence of images induced a certain number k of examples that are needed for recognizing all balconies in the images. In general, one could create every sequence, count k for each sequence, and select the one with the smallest k. This can of course lead to a large number of computations. By introducing following improvements to this brute-force approach, we reduced the needed effort:

- For a subsequence like $[F074\ F072\ F058\ F057\ F041]$, the resulting conceptual models are stored.
- If two subsequence s_1 and s_2 of S have a common starting subsequence s_s (e.g. $s_1 = [F074\ F072\ F058\ F057\ F041]$ and $s_2 = [F074\ F072\ F058\ F057\ F036]$ have $s_s = [F074\ F072\ F058\ F057]$ as a common starting subsequence) this subsequence have to be interpreted only once. We say s_1 and s_2 are *follow-ups* of s_s. This reduces the effort for determining k for s_1 and s_2.

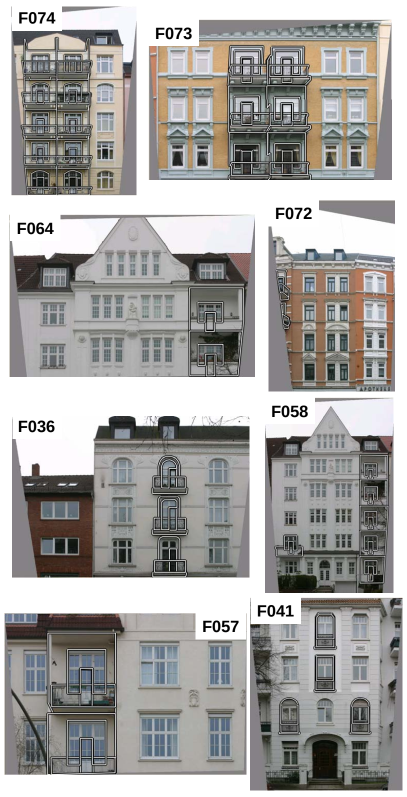

Figure 5: Images with annotated primitives (framed).

- If a best order s_l is found for a subsequence of a certain length l, a next image i is taken and positioned after each image of s_b, thus, leading to subsequences of length $l+1$. For each s_{l+1} k is computed by taking follow-ups into account. In other words, the i_n is moved from start to the end of s_l. For each subsequences of a certain length, a minimum k_m is stored. Thus, if k of a subsequence of that length exceeds k_m, one can interrupt further computations of that subsequence. This reduces the number of sequences needed for interpretation.

In Figure 6, two subsequences are presented with $k = 12$ and $k = 11$. The difference is the order of the last two images, which determines that the right ordering is better than the left one.

For the images in Figure 5, we compute the final ordering shown in Figure 7.

Discussion The experiment shows that the order of presented examples is crucial for the number of iterations needed for learning a certain aggregate. The most important criterion seems to be an early set of examples covering a preferably large variety of aggregate instances. Furthermore, if this early set contains examples which cover extrem properties, like

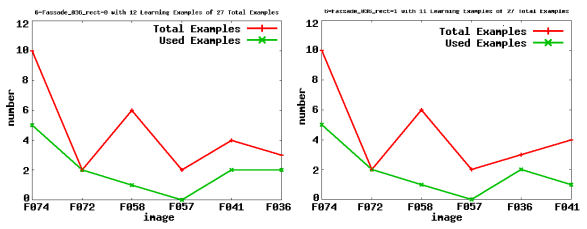

Figure 6: Example for reduced number of examples: when learning from F036 in F041 only one example is used for learning, if F036 does not precede F041 (left diagram) two examples are taken from F041.

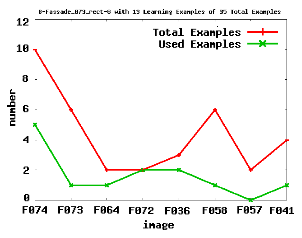

Figure 7: Sequence for the given images which uses fewest examples for learning.

small and large objects, smallest and largest number of a certain object type, the number of iterations is reduced. This is strongly related to the generalization method of the learning module which creates concept descriptions which cover all presented examples. If extremes are selected, regular examples are covered through this generalization. This does not only hold for size and number restrictions but also for spatial relations. Because spatial relations are organized in a taxonomical hierarchy, generalizing relations between objects of distinct examples lead to the computation of the least common subsumes (*lcs*) of these relations. If the *lcs* has low taxonomical depth, it will cover more examples which may come up in future iterations.

5 CONCLUSION

In this work, we have shown the relevance of sample ordering for training on the performance of an incremental learning method. We called this effect the ordering sensitivity of an incremental learning method.

We have shown experiments to evaluate ordering sensitivity related to the tasks of learning models for object classification and learning structural models describing spatial relations.

For the task to find good class sequences for training a classification model, we have shown the possibility to get those sequences beforehand just based on the data. Therefore, we identified the Bayes error as appropriate measure to find class sequences that cause best possible error rates.

For learning structural models we found rules to empirically describe good image sequences.

This way, we can achieve always best performance given a particular learning method. Furthermore, we can reduce the number of samples for learning if the error rate during training a particular class does not drop anymore.

Thus, we have shown the relevance of curriculum learning that should be investigated on any type of incremental learning methods. Further research should concentrate on finding more measures to define curricula before starting training just based on the data but related to the particular learning method.

REFERENCES

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In Bottou, L. and Littman, M., editors, *ICML'09*, pages 41–48, Montreal. Omnipress.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR'05*, volume 1, pages 886–893. IEEE Computer Society.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172.

Foerstner, W. (2007). Annotated image database. *eTRIMS EU-Project, Deliverable D1.1.*

Fukunaga, K. (1972). *Introduction to Statistical Pattern Recognition*. Academic Press, first edition.

Giraud-Carrier, C. (2000). A note on the utility of incremental learning. *AI Communications*, 13(4):215–223.

Langley, P. (1995). Order effects in incremental learning. *Learning in humans and machines: Towards an interdisciplinary learning science.*

McKusick, K. B. and Langley, P. (1991). Constraints on tree structure in concept formation. In *IJCAI'91*, pages 810–816.

Mitchell, T. (1977). Version spaces: A candidate elimination approach for rule learning. In *IJCAI'77*, pages 305–310.

Seewald, A. K. (2005). Digits - a dataset for handwritten digit recognition. Technical Report TR-2005-27, Austrian Research Institut for Artificial Intelligence.

Talavera, L. and Roure, J. (1998). A buffering strategy to avoid ordering effects in clustering. In *ECML'98*, pages 316–321.

Uray, M., Skocaj, D., Roth, P. M., Bischof, H., and Leonardis, A. (2007). Incremental LDA Learning by Combining Reconstructive and Discriminative Approaches. In *BMVC'07*, University of Warwick, UK.

Wenzel, S. and Förstner, W. (2009). The role of sequences for incremental learning. Technical Report TR-IGG-P-2009-04, Department of Photogrammetry, University of Bonn.