Complexity Curve: a Graphical Measure of Data Complexity and Classifier Performance

Supplementary document S2:
Evaluating Classifier Performance with Generalisation Curves

Julian Zubek[1,2], Dariusz Plewczynski[2]

1. Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
2. Centre of New Technologies, University of Warsaw, Warsaw, Poland

We discussed the role of data complexity measures in the evaluation of classification algorithms performance. Knowing characteristics of benchmark data sets it is possible to check which algorithms perform well in the context of scarce data. To fully utilise this information, we present a graphical performance measure called generalisation curve. It is based on learning curve concept and allows to compare the learning process of different algorithms while controlling the variance of the data. To demonstrate its validity we apply it to a set of popular algorithms. We show that the analysis of generalisation curves points to important properties of the learning algorithms and benchmark data sets, which were previously suggested in the literature.

Evaluating classifier performance

The basic schema of classifier evaluation is to train a model on one data sample (training set) and then collect its predictions on another, independent data set (testing set). Overall performance is then calculated using some measure taking into account errors made on the testing set. The most intuitive measure is accuracy, but other measures such as precision, recall or F-measure are widely used. When we are interested in comparing classification algorithms, not just trained classifiers, this basic schema is limited. It allows only to perform a static comparison of different algorithms under specified conditions. All algorithms' parameters are fixed, so are the data sets. The results may not be conclusive since the same algorithm may perform very well or very poor depending on the conditions. Such analysis provides a static view of classification task – there is little to be concluded on the dynamics of the algorithm: its sensitivity to the parameter tuning, requirements regarding the sample size etc.

A different approach, which preserves some of the dynamics, is receiver operating characteristic (ROC) curve (Fawcett, 2006). It is possible to perform ROC analysis for any binary classifier, which returns continuous decisions. The fraction of correctly classified examples in class A is plotted against the fraction of incorrectly classified in class B for different values of the classification threshold. The ROC curve captures not only the sole performance of a classifier, but also its sensitivity to the threshold value selection.

Another graphical measure of classifier performance, which visualises its behaviour depending on a threshold value, is cost curve introduced by Drummond and Holte (2006). They claim that their method is more convenient to use because it allows to visualise confidence intervals and statistical significance of differences between classifiers. However, it still measures the performance of a classifier in a relatively static situation where only threshold value changes.

Both ROC curves and cost curves are applicable only to classifiers with continuous outputs and to two class problems, which limits their usage. What is important is the key idea behind them: instead of giving the user a final solution they give freedom to choose an optimal classifier according to some criteria from a range of options.

The learning curve technique presents in a similar fashion the impact of the sample size on the classification accuracy. The concept itself originates from psychology. It is defined as a plot of learner's performance against the amount of effort invested in learning. Such graphs are widely used in medicine (Schlachta et al., 2001), economics (Nemet, 2006), education (Karpicke and Roediger, 2008), or engineering (Jaber and Glock, 2013). They allow to describe the amount of training required for an employee to perform certain job. They are also used in entertainment industry to scale difficulty level of video games (Sweetser and Wyeth, 2005). In machine learning context they are sometimes referred to as the performance curve (Sing et al., 2005). The effort in such curve is measured with the number of examples in the training set.

Learning curve is a visualisation of an incremental learning process in which data is accumulated and the accuracy of the model increases. It captures the algorithm's generalisation capabilities: using the curve it is possible to estimate what amount of data is needed to successfully train a classifier and when collecting additional data does not introduce any significant improvement. This property is referred to in literature as the sample complexity – a minimal size of the training set required to achieve acceptable performance.

As it was noted above, standard learning curve in machine learning expresses the effort in terms of the training set size. However, for different data sets the impact of including an additional data sample may be different. Also, within the same set the effect of including first 100 samples and last 100 samples is very different. Generalisation curve – an extension of learning curve proposed in this article – deals with these problems by using an effort measure founded on data complexity instead of raw sample size.

Generalisation curve

Generalisation curve is the proposed variant of learning curve based on data set complexity. It is the plot presenting accuracy of a classifier trained on a data subset versus subset's information content, i.e. its Hellinger distance from the whole set. To construct the plot, a number of subsets of a specified size are drawn, the mean Hellinger distance and the mean classifier accuracy are marked on the plot. Trained classifiers are always evaluated on the whole data set, which represents the source of full information. Using such resubstitution in the evaluation procedure may be unintuitive since the obtained scores do not represent true classifier performance on independent data. However this strategy corresponds to information captured by complexity curve and allows to utilise full data set for evaluation without relying on additional splitting procedures. It still allows for a meaningful classification algorithm comparison: the final part of the plot promotes classifiers which fit to the data completely, while the initial part favours classifiers with good generalisation capabilities.

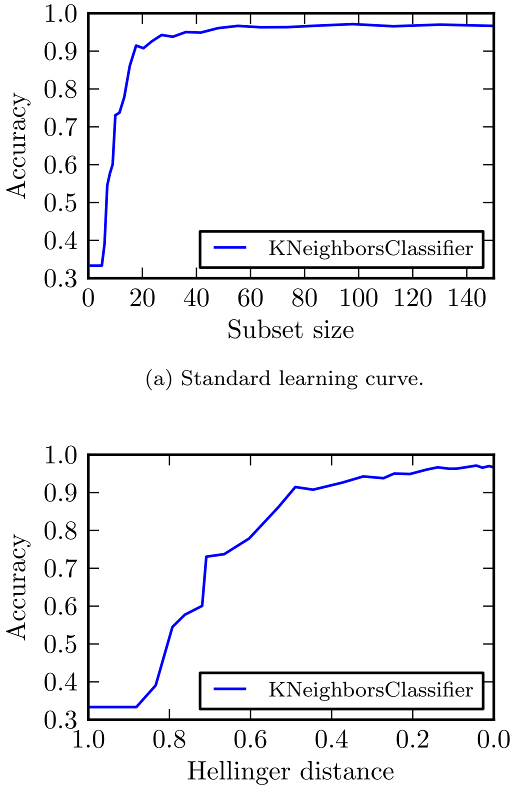
Algorithm 1 presents the exact procedure of calculating generalisation curve.

Standard learning curve and generalisation curve for the same data and classifier are depicted in Figure 1. The generalisation curve gives more insight into algorithm learning dynamics, because it emphasises initial learning phases in which new information is acquired. In the case of k-neighbours classifier we can see that it is unable to generalise if the training sample is too small. Then it enters a rapid learning phase which gradually shifts to a final plateau, when the algorithm is unable to incorporate any new information into the model.

In comparison with standard learning curve, generalisation curve should be less dependent on data characteristics and more suitable for the comparison of algorithms. Again the score, which can be easily obtained from such plot is the area under the curve.

Algorithm 1 Procedure for calculating generalisation curve.

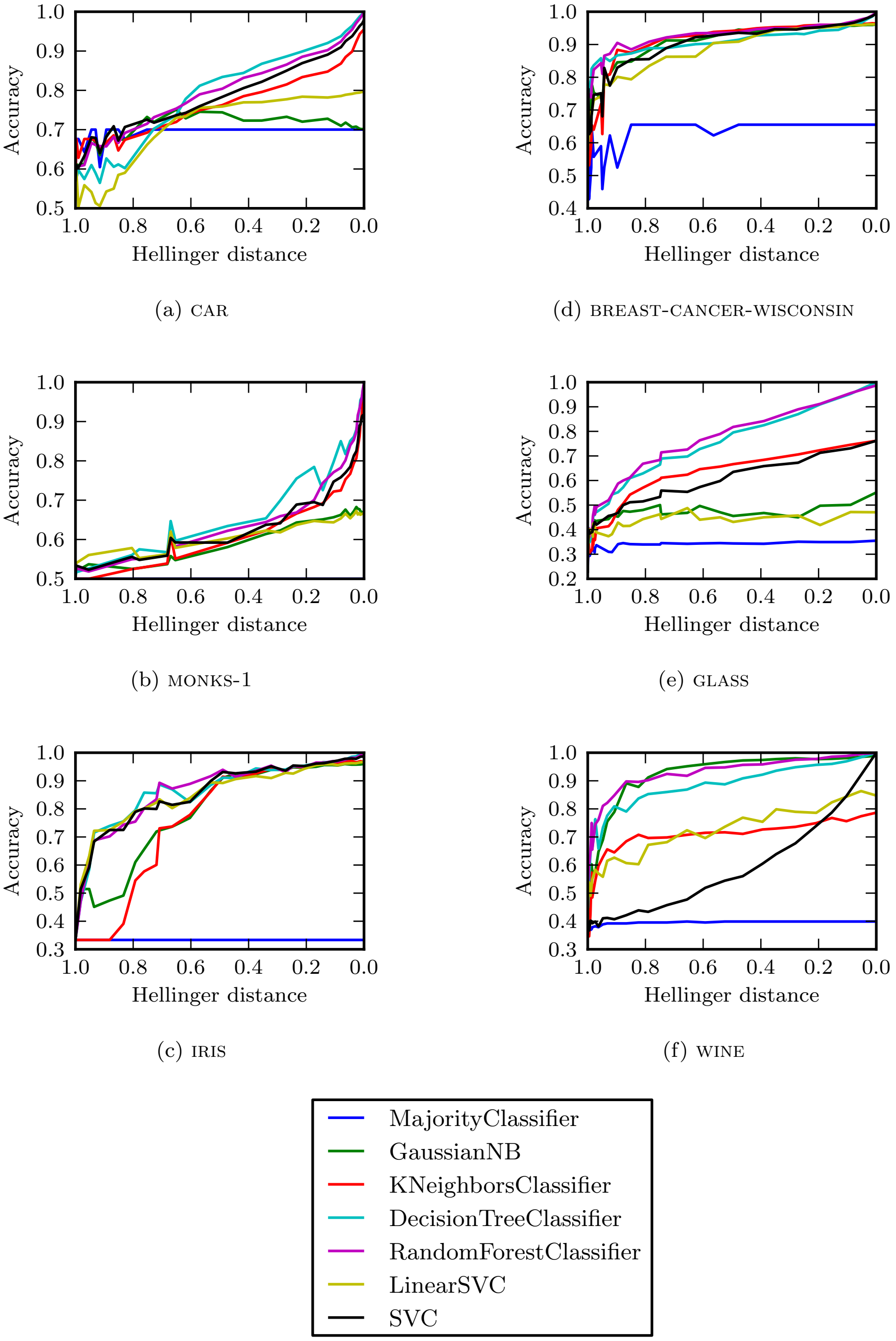
```
$D$ – original data set, $K$ – number of samples.

1. Transform $D$ with whitening transform and/or ICA to obtain $D_I$.
2. Estimate probability distribution(s) from $D_I$.
3. For $i$ in $1 \dots |D|$:
    (a) For $j$ in $1 \dots K$:
        i. Draw subset $S_i^j \subseteq D$ such that $|S_i^j| = i$ and its analogous subset $O_i^j \subseteq D_I$.
        ii. Calculate distance $l_i^j$ between $O_i^j$ and $D_I$ according to the standard or conditional formula.
        iii. Train the classifier on $S_i^j$ and evaluate it on $D$ to get its accuracy $a_i^j$.
    (b) Calculate mean $l_i$ and mean $a_i$:
        $$l_i = \frac{1}{K}\sum_{j=1}^{K} l_i^j \qquad a_i = \frac{1}{K}\sum_{j=1}^{K} a_i^j$$

Generalisation curve is a plot of $a_i$ vs $l_i$.
```



(a) Standard learning curve.

(b) Generalisation curve.

Figure 1: Learning curve and generalisation curve for data set IRIS and K-neighbours classifier ($k = 5$).



(a) CAR

(d) BREAST-CANCER-WISCONSIN

(b) MONKS-1

(e) GLASS

(c) IRIS

(f) WINE

Figure 2: Generalisation curves for various classification algorithms.

Generalisation curves for benchmark data sets

Another application of the proposed methodology is comparison of classification algorithms based on generalisation curves. We evaluated a set of standard algorithms available in scikit-learn library (Pedregosa et al., 2011). As benchmark data sets we used the same sets from UCI repository as in section demonstrating interpretability of complexity curves. The following classification algorithms were evaluated:

- MajorityClassifier – artificial classifier which always returns the label of the most frequent class in the training set,
- GaussianNB – naïve Bayes classifier with Gaussian kernel probability estimate,
- KNeighborsClassifier – k-nearest neighbours, $k = 5$,
- DecisionTreeClassifier – CART decision tree algorithm,
- RandomForestClassifier – random forest with 10 CART trees,
- LinearSVC – linear spport vector machine (without kernel transformation), cost parameter $C = 1$,
- SVC – support vector machine with radial basis function kernel (RBF): $exp(-\frac{1}{n}|x - x'|^2)$, n – number of features, cost parameter $C = 1$.

Generalisation curves were calculated for all classifiers with the same random seed, to make sure that the algorithms are trained on exactly the same data samples. The obtained curves are presented in Figure 2.

The performance of the majority classifier is used as a baseline. We expect that the worst-case performance of any classifier should be at least at the baseline level. This is indeed observed in the plots: most classifiers start at the baseline level and then their accuracy steadily increases as more data are accumulated. The notable exception is the CAR data set, where the accuracy of decision tree and linear SVM stays below the accuracy of the majority classifier in the initial phase of learning. We attribute this to the phenomena known as anti-learning (Kowalczyk and Chapelle, 2005). It occurs in certain situations, when the sample size is smaller than the number of attributes, and correct classification of the examples in the training set may lead to an inverted classification of the examples in the testing set.

In an ideal situation the learning algorithm is able to utilise every bit of additional information identified by the complexity curve to improve the classification and the accuracy gain is linear. The generalisation curve should be then a straight line. Convex generalisation curve indicates that complexity curve is only a loose upper bound on classifier variance, in other words algorithm is able to fit a model using less information than indicated by the complexity curve. On the other hand, concave generalisation curve corresponds to a situation when the independence assumption is broken and including information on attributes interdependencies, not captured by complexity curve, is necessary for successful classification.

On most of the benchmark data sets generalisation curves are generally convex, which means that the underlining complexity curves constitute proper upper bounds on the variance error component. The bound is relatively tight in the case of GLASS data set, looser in the case of IRIS, and the loosest for WINE and BREAST-CANCER-WISCONSIN data. A natural conclusion is that a lot of variability contained in this last data set and captured by the Hellinger distance is irrelevant to the classification task. The most straightforward explanation would be the presence of unnecessary attributes uncorrelated with class, which can be ignored altogether. This is consistent with the results of various studies in feature selection. Choubey et al. (1996) identified that in GLASS data 7-8 attributes (78-89%) are relevant, in IRIS data 3 attributes (75%), and in BREAST-CANCER-WISCONSIN 5-7 attributes (56-78%). Similar results were obtained for BREAST-CANCER-WISCONSIN in other studies, which found that only 4 of the original attributes (44%) contribute

	Classifiers						
Data set	M	NB	kNN	DT	RF	SVM$_l$	SVM$_r$
car	0.70 (7)	0.71 (5.5)	0.76 (4)	0.79 (2.5)	**0.80 (1)**	0.71 (5.5)	0.79 (2.5)
monks-1	0.50 (7)	0.57 (6)	0.58 (5)	**0.63 (1)**	0.61 (2)	0.59 (4)	0.60 (3)
iris	0.33 (7)	0.79 (5)	0.76 (6)	**0.87 (1.5)**	**0.87 (1.5)**	0.85 (4)	0.86 (3)
bcw	0.64 (7)	0.91 (4)	0.92 (2)	0.91 (4)	**0.93 (1)**	0.89 (6)	0.91 (4)
glass	0.34 (7)	0.47 (5)	0.64 (3)	0.76 (2)	**0.78 (1)**	0.44 (6)	0.61 (4)
wine	0.40 (7)	**0.93 (1.5)**	0.71 (5)	0.90 (3)	**0.93 (1.5)**	0.73 (4)	0.60 (6)
Avg. rank	7	4.5	4	2.33	1.33	4.92	3.75

Table 1: Areas under generalisation curves for various algorithms. Values given in brackets are ranks among all algorithms (ties solved by ranking randomly and averaging ranks). M – majority classifier, NB – naïve Bayes, kNN – k-nearest neighbours, DT – decision tree, RF – random forest, SVM$_r$ – support vector machine with RBF kernel, SVM$_l$ – linear support vector machine.

to the classification (Ratanamahatana and Gunopulos, 2003; Liu et al., 1998). Dy and Brodley (2004) obtained best classification results for wine data set with 7 attributes (54%).

On monks-1 and car data generalisation curves for all algorithms besides naïve Bayes and linear SVM are concave. This is an indication of models relying heavily on attribute interdependencies to determine the correct class. This is not the case for naïve Bayes and linear SVM because these methods are unable to model attribute interactions. This is not surprising: both monks-1 and car are artificial data sets with discrete attributes devised for evaluation of rule-based and tree-based classifiers Thrun et al. (1991); Bohanec and Rajkovič (1988). Classes are defined with logical formulas utilising relations of multiple attributes rather than single values – clearly the attributes are interdependent.

An interesting case is RBF SVM on wine data set. Even though it is possible to model the problem basing on a relatively small sample, it overfits strongly by trying to include unnecessary interdependencies. This is a situation when variance of a model is greater than indicated by the complexity curve.

To compare performance of different classifiers, we computed areas under generalisation curves (AUGC) for all data sets. Results are presented in Table 1. Random forest classifier obtained the highest scores on all data sets except monks-1 where single decision tree performed the best. On wine data set naïve Bayes achieved AUGC comparable with random forest.

AUGC values obtained on different data sets are generally not comparable, especially when the base level – majority classifier performance – differs. Therefore, to obtain a total ranking we ranked classifiers separately on each data set and averaged the ranks. According to this criteria random forest is the best classifier on these data sets, followed by decision tree and support vector machine with radial basis function kernel.

Comparison of algorithms using AUGC favours an algorithm which is characterised simultaneously by good accuracy and small sample complexity (ability to draw conclusions from a small sample). The proposed procedure helps to avoid applying an overcomplicated model and risking overfitting when a simpler model is adequate. It takes into account algorithm properties ignored by standard performance metrics.

References

Bohanec, M. and Rajkovič, V. (1988). Knowledge Acquisition and Explanation for Multi-Attribute Decision. In *Making, 8 th International Workshop "Expert Systems and Their Applications.*

Choubey, S. K., Deogun, J. S., Raghavan, V. V., and Sever, H. (1996). A comparison of feature selection algorithms in the context of rough classifiers. In *, Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, 1996*, volume 2, pages 1122–1128 vol.2.

Drummond, C. and Holte, R. C. (2006). Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, 65(1):95–130.

Dy, J. G. and Brodley, C. E. (2004). Feature selection for unsupervised learning. *The Journal of Machine Learning Research*, 5:845–889.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874.

Jaber, M. Y. and Glock, C. H. (2013). A learning curve for tasks with cognitive and motor elements. *Computers & Industrial Engineering*, 64(3):866–871.

Karpicke, J. D. and Roediger, H. L. (2008). The Critical Importance of Retrieval for Learning. *Science*, 319(5865):966–968.

Kowalczyk, A. and Chapelle, O. (2005). An analysis of the anti-learning phenomenon for the class symmetric polyhedron. In *Proceedings of the 16th International Conference on Algorithmic Learning Theory*. Springer.

Liu, H., Motoda, H., and Dash, M. (1998). A monotonic measure for optimal feature selection. In *Machine Learning: ECML-98*, pages 101–106. Springer.

Nemet, G. F. (2006). Beyond the learning curve: factors influencing cost reductions in photovoltaics. *Energy Policy*, 34(17):3218 – 3232.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ratanamahatana, C. a. and Gunopulos, D. (2003). Feature selection for the naive bayesian classifier using decision trees. *Applied artificial intelligence*, 17(5-6):475–487.

Schlachta, C. M., Mamazza, J., Seshadri, P. A., Cadeddu, M., Gregoire, R., and Poulin, E. C. (2001). Defining a learning curve for laparoscopic colorectal resections. *Diseases of the Colon & Rectum*, 44(2):217–222.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). ROCR: visualizing classifier performance in R. *Bioinformatics*, 21(20):3940–3941.

Sweetser, P. and Wyeth, P. (2005). GameFlow: a model for evaluating player enjoyment in games. *Comput. Entertain.*, 3(3):3–3.

Thrun, S. B., Bala, J. W., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., Jong, K. A. D., Dzeroski, S., Fisher, D. H., Fahlman, S. E., Hamann, R., Kaufman, K. A., Keller, S., Kononenko, I., Kreuziger, J. S., Michalski, R. S., Mitchell, T. A., Pachowicz, P. W., Vafaie, H., Welde, W. V. d., Wenzel, W., Wnek, J., and Zhang, J. (1991). The MONK's problems: A Performance Comparison of Different Learning Algorithms. Technical Report CMU-CS-91-197, Carnegie Mellon University.