Complexity curve: a graphical measure of data complexity and classifier performance

Julian Zubek, Dariusz M Plewczynski

We describe a method for assessing data set complexity based on the estimation of the underlining probability distribution and Hellinger distance. Contrary to some popular measures it is not focused on the shape of decision boundary in a classification task but on the amount of available data with respect to attribute structure. Complexity is expressed in terms of graphical plot, which we call complexity curve. We use it to propose a new variant of learning curve plot called generalisation curve. Generalisation curve is a standard learning curve with x-axis rescaled according to the data set complexity curve. It is a classifier performance measure, which shows how well the information present in the data is utilised. We perform theoretical and experimental examination of properties of the introduced complexity measure and show its relation to the variance component of classification error. We compare it with popular data complexity measures on 81 diverse data sets and show that it can contribute to explaining the performance of specific classifiers on these sets. Then we apply our methodology to a panel of benchmarks of standard machine learning algorithms on typical data sets, demonstrating how it can be used in practice to gain insights into data characteristics and classifier behaviour. Moreover, we show that complexity curve is an effective tool for reducing the size of the training set (data pruning), allowing to significantly speed up the learning process without reducing classification accuracy. Associated code is available to download at: https://github.com/zubekj/complexity_curve (open source Python implementation).

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

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

[1]**Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland, Centre of New Technologies, University of Warsaw, Warsaw, Poland, j.zubek@cent.uw.edu.pl**
[2]**Centre of New Technologies, University of Warsaw, Warsaw, Poland, d.plewczynski@cent.uw.edu.pl**

ABSTRACT

We describe a method for assessing data set complexity based on the estimation of the underlining probability distribution and Hellinger distance. Contrary to some popular measures it is not focused on the shape of decision boundary in a classification task but on the amount of available data with respect to attribute structure. Complexity is expressed in terms of graphical plot, which we call complexity curve. We use it to propose a new variant of learning curve plot called generalisation curve. Generalisation curve is a standard learning curve with x-axis rescaled according to the data set complexity curve. It is a classifier performance measure, which shows how well the information present in the data is utilised.
We perform theoretical and experimental examination of properties of the introduced complexity measure and show its relation to the variance component of classification error. We compare it with popular data complexity measures on 81 diverse data sets and show that it can contribute to explaining the performance of specific classifiers on these sets.
Then we apply our methodology to a panel of benchmarks of standard machine learning algorithms on typical data sets, demonstrating how it can be used in practice to gain insights into data characteristics and classifier behaviour. Moreover, we show that complexity curve is an effective tool for reducing the size of the training set (data pruning), allowing to significantly speed up the learning process without reducing classification accuracy.
Associated code is available to download at: `https://github.com/zubekj/complexity_curve` (open source Python implementation).

Keywords: Learning curves, Data complexity, Data pruning, Hellinger distance, Bias-variance decomposition, Performance measures

INTRODUCTION

It is common knowledge in machine learning community that the difficulty of classification problems varies greatly. Sometimes it is enough to use simple out of the box classifier to get a very good result and sometimes careful preprocessing and model selection are needed to get any non-trivial result at all. The difficulty of a classification task clearly stems from certain properties of the data set, yet we still have problems with defining those properties in general.

Bias-variance decomposition (Domingos, 2000) demonstrates that the error of a predictor can be attributed to three sources: bias, coming from inability of an algorithm to build an adequate model for the relationship present in data, variance, coming from inability to estimate correct model parameters from an imperfect data sample, and some irreducible noise. Following this line of reasoning, difficulty of a classification problem may come partly from the complexity of the relation between dependent variable and explanatory variables, partly from the scarcity of information in the training sample, and partly from an overlap between classes. This is identical to sources of classification difficulty identified by Ho and Basu (2002), who labelled the three components: ‘complex decision boundary’, ‘small sample size and dimensionality induced sparsity’ and ‘ambiguous classes’.

In this article we introduce a new measure of data complexity targeted at sample sparsity, which is mostly associated with variance error component. We aim to measure information saturation of a data set without making any assumptions on the form of relation between dependent variable and the rest of variables, so explicitly disregarding shape of decision boundary and classes ambiguity. Our complexity measure takes into account the number of samples, the number of attributes and attributes internal structure, under a simplifying assumption of attribute independence. The key idea is to check how well a data set can be approximated by its subsets. If the probability distribution induced by a small data sample is very similar to the probability distribution induced by the whole data set we say that the set is saturated with information and presents an opportunity to learn the relationship between variables without promoting the variance. To operationalise this notion we introduce two kinds of plots:

- Complexity curve – a plot presenting how well subsets of growing size approximate distribution of attribute values. It is a basic method applicable to clustering, regression and classification problems.
- Conditional complexity curve – a plot presenting how well subsets of growing size approximate distribution of attribute values conditioned on class. It is applicable to classification problems and more robust against class imbalance or differences in attributes structure between classes.

Since the proposed measure characterise the data sample itself without making any assumptions as to how that sample will be used it should be applicable to all kinds of problems involving reasoning from data. In this work we focus on classification tasks since this is the context in which data complexity measures were previously applied. We compare area under the complexity curve with popular data complexity measures and show how it complements the existing metrics. We also demonstrate that it is useful for explaining classifier performance by showing that the area under the complexity curve is correlated with the area under the receiver operating characteristic (AUC ROC) for popular classifiers tested on 81 benchmark data sets.

We propose two immediate applications of the developed method. The first one is connected with the fundamental question: how much of the original sample is needed to build a successful predictor? We pursue this topic by proposing a data pruning strategy based on complexity curve and evaluating it on large data sets. We show that it can be considered as an alternative to progressive sampling strategies (Provost et al., 1999).

The second proposed application is classification algorithm comparison. 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.

RELATED LITERATURE

Problem of measuring data complexity in the context of machine learning is broadly discussed. Our beliefs are similar to Ho (2008), who stated the need for including data complexity analysis in algorithm comparison procedures. The same need is also discussed in fields outside machine learning, for example in combinatorial optimisation (Smith-Miles and Lopes, 2012).

The general idea is to select a sufficiently diverse set of problems to demonstrate both strengths and weaknesses of the analysed algorithms. The importance of this step was stressed by Macià et al. (2013), who demonstrated how algorithm comparison may be biased by benchmark data sets selection, and showed how the choice my guided by complexity measures. Characterising problem space with some metrics makes it possible to estimate regions in which certain algorithms perform well (Luengo and Herrera, 2013), and this opens up possibilities of meta-learning (Smith-Miles et al., 2014).

In this context complexity measures are used not only as predictors of classifier performance but more importantly as diversity measures capturing various properties of the data sets. It is useful when the measures themselves are diverse and focus on different aspects of the data to give as complete characterisation of the problem space as possible. In the later part of the article we demonstrate that complexity curve fits well into the landscape of currently used measures, offering new insights into data characteristics.

Measuring data complexity

A set of practical measures of data complexity with regard to classification was introduced by Ho and Basu (2002), and later extended by Ho et al. (2006) and Orriols-Puig et al. (2010). It is routinely used in tasks involving classifier evaluation (Macià et al., 2013; Luengo and Herrera, 2013) and meta-learning (Díez-Pastor et al., 2015; Mantovani et al., 2015). Some of these measures are based on the overlap of values of specific attributes, examples include Fisher's discriminant ratio, volume of overlap region, attribute efficiency etc. The others focus directly on class separability, this groups includes measures such as the fraction of points on the boundary, linear separability, the ratio of intra/inter class distance. In contrast to our method, such measures focus on specific properties of the classification problem, measuring decision boundary and class overlap. Topological measures concerned with data sparsity, such as ratio of attributes to observations, attempt to capture similar properties as complexity curve.

Li and Abu-Mostafa (2006) defined data set complexity in the context of classification using the general concept of Kolmogorov complexity. They proposed a way to measure data set complexity using the number of support vectors in support vector machine (SVM) classifier. They analysed the problems of data decomposition and data pruning using above methodology. A graphical representation of the data set complexity called the complexity-error plot was also introduced. The main problem with their approach is the selection of very specific and complex machine learning algorithms, which may render the results in less universal way, and which is prone to biases specific for SVMs. This make their method unsuitable for diverse machine learning algorithms comparison.

Another approach to data complexity is to analyse it on instance level. This kind of analysis is performed by Smith et al. (2013) who attempted to identify which instances are misclassified by various classification algorithm. They devised local complexity measures calculated with respect to single instances and later tried to correlate average instance hardness with global data complexity measures of Ho and Basu (2002). They discovered that is mostly correlated with class overlap. This makes our work complementary, since in our complexity measure we deliberately ignore class overlap and individual instance composition to isolate another source of difficulty, namely data scarcity.

Yin et al. (2013) proposed a method of feature selection based on Hellinger distance (a measure of similarity between probability distributions). The idea was to choose features, which conditional distributions (depending on the class) have minimal affinity. In the context of our framework this could be interpreted as measuring data complexity for single features. The authors demonstrated experimentally that for the high-dimensional imbalanced data sets their method is superior to popular feature selection methods using Fisher criterion, or mutual information.

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.

DEFINITIONS

In the following sections we define formally all measures used throughout the paper. Basic intuitions, assumptions, and implementation choices are discussed. Finally, algorithms for calculating complexity curve, conditional complexity curve, and generalisation curve are given.

Measuring data complexity with samples

In a typical machine learning scenario we want to use information contained in a collected data sample to solve a more general problem which our data describe. Problem complexity can be naturally measured by the size of a sample needed to describe the problem accurately. We call the problem complex, if we need to collect a lot of data in order to get any results. On the other hand, if a small amount of data suffices we say the problem has low complexity.

How to determine if a data sample describes the problem accurately? Any problem can be described with a multivariate probability distribution P of a random vector X. From P we sample our finite data sample D. Now, we can use D to build the estimated probability distribution of X – P_D. P_D is the approximation of P. If P and P_D are identical we know that data sample D describes the problem perfectly and collecting more observations would not give us any new information. Analogously, if P_D is very different from P we can be certain that the sample is too small.

To measure similarity between probability distributions we use Hellinger distance. For two continuous distributions P and P_D with probability density functions p and p_D it is defined as:

$$H^2(P,P_D) = \frac{1}{2}\int\left(\sqrt{p(x)}-\sqrt{p_D(x)}\right)^2 dx$$

The minimum possible distance 0 is achieved when the distributions are identical, the maximum 1 is achieved when any event with non-zero probability in P has probability 0 in P_D and vice versa. Simplicity and naturally defined 0–1 range make Hellinger distance a good measure for capturing sample information content.

In most cases we do not know the underlining probability distribution P representing the problem and all we have is a data sample D, but we can still use the described complexity measure. Let us picture our data D as the true source of knowledge about the problem and the estimated probability distribution P_D as the reference distribution. Any subset $S \subset D$ can be treated as a data sample and a probability distribution P_S estimated from it will be an approximation of P_D. By calculating $H^2(P_D, P_S)$ we can assess how well a given subset represent the whole available data, i.e. determine its information content.

Obtaining a meaningful estimation of a probability distribution from a data sample poses difficulties in practice. The probability distribution we are interested in is the joint probability on all attributes. In that context most of the realistic data sets should be regarded as extremely sparse and naïve probability estimation using frequencies of occurring values would result in mostly flat distribution. This can be called the curse of dimensionality. Against this problem we apply a naïve assumption that all attributes are independent. This may seem like a radical simplification but, as we will demonstrate later, it yields good results in practice and constitute a reasonable baseline for common machine learning techniques. Under the independence assumption we can calculate the joint probability density function f from the marginal density functions $f_1, \dots, f_n$:

$$f(x) = f_1(x_1) f_2(x_2) \dots f_n(x_n)$$

We will now show the derived formula for Hellinger distance under the independence assumption. Observe that the Hellinger distance for continuous variables can be expressed in another form:

$$\frac{1}{2}\int \left(\sqrt{f(x)} - \sqrt{g(x)}\right)^2 dx =$$
$$\frac{1}{2}\int \left(f(x) - 2\sqrt{f(x)g(x)} + g(x)\right) dx =$$
$$\frac{1}{2}\int f(x)\,dx - \int \sqrt{f(x)g(x)}\,dx + \frac{1}{2}\int g(x)\,dx =$$
$$1 - \int \sqrt{f(x)g(x)}\,dx$$

In the last step we used the fact the that the integral of a probability density over its domain must be one.

We will consider two multivariate distributions F and G with density functions:

$$f(x_1,\dots,x_n) = f_1(x_1)\dots f_n(x_n)$$
$$g(x_1,\dots,x_n) = g_1(x_1)\dots g_n(x_n)$$

The last formula for Hellinger distance will now expand:

$$1 - \int \cdots \int \sqrt{f(x_1,\dots,x_n)g(x_1,\dots,x_n)}\,dx_1\dots dx_n =$$
$$1 - \int \cdots \int \sqrt{f_1(x_1)\dots f_n(x_n)g_1(x_1)\dots g_n(x_n)}\,dx_1\dots dx_n =$$
$$1 - \int \sqrt{f_1(x_1)g_1(x_1)}\,dx_1 \dots \int \sqrt{f_n(x_n)g_n(x_n)}\,dx_n$$

In this form variables are separated and parts of the formula can be calculated separately.

Practical considerations

Calculating the introduced measure of similarity between data set in practice poses some difficulties. First, in the derived formula direct multiplication of probabilities occurs, which leads to problems with numerical stability. We increased the stability by switching to the following formula:

$$1 - \int \sqrt{f_1(x_1)g_1(x_1)}\,dx_1 \dots \int \sqrt{f_n(x_n)g_n(x_n)}\,dx_n =$$

$$1-\left(1-\frac{1}{2}\int\left(\sqrt{f_1(x_1)}-\sqrt{g_1(x_1)}\right)^2dx_1\right)\ldots\left(1-\frac{1}{2}\int\left(\sqrt{f_n(x_n)}-\sqrt{g_n(x_n)}\right)^2dx_2\right)=$$
$$1-\left(1-H^2(F_1,G_1)\right)\ldots\left(1-H^2(F_n,G_n)\right)$$

For continuous variables probability density function is routinely done with kernel density estimation (KDE) – a classic technique for estimating the shape continuous probability density function from a finite data sample (Scott, 1992). For sample $(x_1,x_2,\ldots,x_n)$ estimated density function has a form:

$$\hat{f}_h(x)=\frac{1}{nh}\sum_{i=1}^{n}K\left(\frac{x-x_i}{h}\right)$$

where K is the kernel function and h is a smoothing parameter – bandwidth. In our experiments we used Gaussian function as the kernel. This is a popular choice, which often yields good results in practice. The bandwidth was set according to the modified Scott's rule (Scott, 1992):

$$h=\frac{1}{2}n^{-\frac{1}{d+4}},$$

where n is the number of samples and d number of dimensions.

In many cases the independence assumption can be supported by preprocessing input data in a certain way. A very common technique, which can be applied in this situation is the whitening transform. It transforms any set of random variables into a set of uncorrelated random variables. For a random vector X with a covariance matrix Σ a new uncorrelated vector Y can be calculated as follows:

$$\Sigma=PDP^{-1}$$
$$W=PD^{-\frac{1}{2}}P^{-1}$$
$$Y=XW$$

where D is diagonal matrix containing eigenvalues and P is matrix of right eigenvectors of Σ. Naturally, lack of correlation does not implicate independence but it nevertheless reduces the error introduced by our independence assumption. Furthermore, it blurs the difference between categorical variables and continuous variables putting them on an equal footing. In all further experiments we use whitening transform preprocessing and then treat all variables as continuous.

A more sophisticated method is a signal processing technique known as Independent Component Analysis (ICA) (Hyvärinen and Oja, 2000). It assumes that all components of an observed multivariate signal are mixtures of some independent source signals and that the distribution of the values in each source signal is non-gaussian. Under these assumption the algorithm attempts to recreate the source signals by splitting the observed signal into the components as independent as possible. Even if the assumptions are not met, ICA technique can reduce the impact of attributes interdependencies. Because of its computational complexity we used it as an optional step in our experiments.

Machine learning task difficulty

Our data complexity measure can be used for any type of problem described through a multivariate data sample. It is applicable to regression, classification and clustering tasks. The relation between the defined data complexity and the difficulty of a specific machine learning task needs to be investigated. We will focus on supervised learning case. Classification error will be measured as mean 0-1 error. Data complexity will be measured as mean Hellinger distance between real and estimated probability distributions of attributes conditioned on target variable:

$$\frac{1}{m}\sum_{i=1}^{m}H^2(P(X|Y=y_i),P_D(X|Y=y_i))$$

where X – vector of attributes, Y – target variable, $y_1,y_2,\ldots y_m$ – values taken by Y.

It has been shown that error of an arbitrary classification or regression model can be decomposed into three parts:

$$\text{Error}=\text{Bias}+\text{Variance}+\text{Noise}$$

Domingos (2000) proposed an universal scheme of decomposition, which can be adapted for different loss functions. For a classification problem and 0-1 loss L expected error on sample x for which the true label is t, and the predicted label given a traning set D is y can be expressed as:

$$\begin{aligned} & E_{D,t}[\mathbb{1}(t \neq y)] \\ & \quad = \mathbb{1}(E_t[t] \neq E_D[y]) + c_2 E_D[\mathbb{1}(y \neq E_D[y])] + c_1 E_t[\mathbb{1}(t \neq E_t[t])] \\ & \quad = B(x) + c_2 V(x) + c_1 N(x) \end{aligned}$$

where B – bias, V – variance, N – noise. Coefficients c_1 and c_2 are added to make the decomposition consistent for different loss functions. In this case they are equal to:

$$c_1 = P_D(y = E_t[t]) - P_D(y \neq E_t[t]) P_t(y = t \,|\, E_t[t] \neq t)$$

$$c_2 = \begin{cases} 1 & \text{if } E_t[t] = E_D[y] \\ -P_D(y = E_t[t] \,|\, y \neq E_D[y]) & \text{otherwise.} \end{cases}$$

Bias comes from an inability of the applied model to represent the true relation present in data, variance comes from an inability to estimate optimal model parameters from the data sample, noise is inherent to the solved task and irreducible. Since our complexity measure is model agnostic it clearly does not include bias component. As it does not take into account the dependent variable, it cannot measure noise either. All that is left to investigate is the relation between our complexity measure and variance component of the classification error.

The variance error component is connected with overfitting, when the model fixates over specific properties of a data sample and looses generalisation capabilities over the whole problem domain. If the training sample represented the problem perfectly and the model was fitted with perfect optimisation procedure variance would be reduced to zero. The less representative the training sample is for the whole problem domain, the larger the chance for variance error.

This intuition can be supported by comparing our complexity measure with the error of the Bayes classifier. We will show that they are closely related. Let Y be the target variable taking on values $v_1, v_2, \ldots, v_m$, $f_i(x)$ an estimation of $P(X = x|Y = v_i)$ from a finite sample D, and $g(y)$ an estimation of $P(Y = y)$. In such setting 0-1 loss of the Bayes classifier on a sample x with the true label t is:

$$\mathbb{1}(t \neq y) = \mathbb{1}\left(t \neq \arg\max_i \left(g(v_i) f_i(x)\right)\right)$$

Let assume that $t = v_j$. Observe that:

$$v_j = \arg\max_i \left(g(v_i) f_i(x)\right) \Leftrightarrow \forall_i g(v_j) f_j(x) - g(v_i) f_i(x) \geq 0$$

which for the case of equally frequent classes reduces to:

$$\forall_i f_j(x) - f_i(x) \geq 0$$

We can simultanously add and substract term $P(X = x \,|\, Y = v_j) - P(X = x \,|\, Y = v_i)$ to obtain:

$$\begin{aligned} \forall_i \; & (f_j(x) - P(X = x \,|\, Y = v_j)) + \\ & (P(X = x \,|\, Y = v_i) - f_i(x)) + \\ & (P(X = x \,|\, Y = v_j) - P(X = x \,|\, Y = v_i)) \geq 0 \end{aligned}$$

We know that $P(X = x \,|\, Y = v_j) - P(X = x \,|\, Y = v_i) \geq 0$, so as long as estimations $f_i(x)$, $f_j(x)$ do not deviate too much from real distributions the inequality is satisfied. It will not be satisfied (i.e. an error will take place) only if the estimations deviate from the real distributions in a certain way (i.e. $f_j(x) < P(X = x|Y = v_j)$ and $f_i(x) > P(X = x|Y = v_i)$) and the sum of these deviations is greater than

$P(X=x|Y=v_j) - P(X=x|Y=v_i)$. The Hellinger distance between $f_i(x)$ and $P(X=x|Y=v_i)$ measures the deviation. This shows that by minimising Hellinger distance we are also minimising error of the Bayes classifier. Converse may not be true: not all deviations of probability estimates result in classification error.

In the introduced complexity measure we assumed independency of all attributes, which is analogous to the assumption of naïve Bayes. Small Hellinger distance between class-conditioned attribute distributions induced by sets A and B means that naïve Bayes trained on set A and tested on set B will have only very slight variance error component. Of course, if the indepedence assumption is broken bias error component may still be substantial.

Complexity curve

Complexity curve is a graphical representation of a data set complexity. It is a plot presenting the expected Hellinger distance between a subset and the whole set versus subset size:

$$CC(n) = E[H^2(P, Q_n)]$$

where P is the empirical probability distribution estimated from the whole set and Q_n is the probability distribution estimated from a random subset of size $n \leq |D|$. Let us observe that $CC(|D|) = 0$ because $P = Q_{|D|}$. Q_0 is undefined, but for the sake of convenience we assume $CC(0) = 1$.

Algorithm 1 Procedure for calculating complexity curve.

D – original data set, K – number of random subsets of the specified size.

1. Transform D with whitening transform and/or ICA to obtain D_I.
2. Estimate probability distribution for each attribute of D_I and calculate joint probability distribution – P.
3. For i in $1 \ldots |D_I|$ (with an optional step size d):
   (a) For j in $1 \ldots K$:
      i. Draw subset $S_i^j \subseteq D_I$ such that $|S_i^j| = i$.
      ii. Estimate probability distribution for each attribute of S_i^j and calculate joint probability distribution – Q_i^j.
      iii. Calculate Hellinger distance: $l_i^j = H^2(P, Q_i^j)$.
   (b) Calculate mean m_i and standard error s_i:

$$m_i = \frac{1}{K}\sum_{j=1}^{K} l_i^j \qquad s_i = \sqrt{\frac{1}{K}\sum_{j=1}^{K}\left(m_i - l_i^j\right)^2}$$

Complexity curve is a plot of $m_i \pm s_i$ vs i.

To estimate complexity curve in practice, for each subset size K random subsets are drawn and the mean value of Hellinger distance, along with standard error, is marked on the plot. The Algorithm 1 presents the exact procedure. Parameters K (the number of samples of a specified size) and d (sampling step size) controls the trade-off between the precision of the calculated curve and the computation time. In all experiments, unless stated otherwise, we used values $K = 20$, $d = \frac{|D|}{60}$. Regular shapes of the obtained curves did not suggest the need for using larger values.

Figure 1 presents a sample complexity curve. It demonstrates how by drawing larger subsets of the data we get better approximations of the original distribution, as indicated by the decreasing Hellinger distance. The logarithmic decrease of the distance is characteristic: it means that with a relatively small number of samples we can recover general characteristics of the distribution, but to model the details precisely we need a lot more data points. The shape of the curve is very regular, with just minimal variations. It

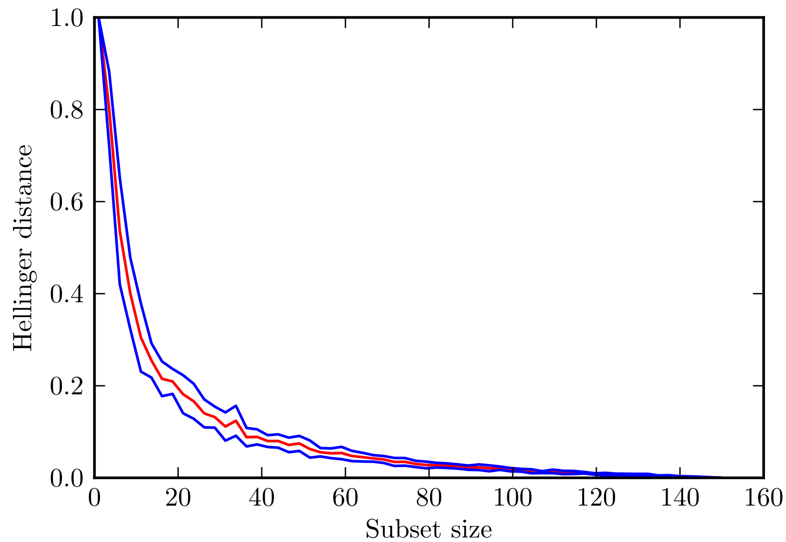

Figure 1. Complexity curve for **iris** data set. Red line represents the mean value, blue lines represent mean $\pm$ standard deviation.

means that the subset size has a far greater impact on the Hellinger distance that the composition of the individual subsets.

The shape of the complexity curve captures the information on the complexity of the data set. If the data is simple, it is possible to represent it relatively well with just a few instances. In such case, the complexity curve is very steep at the beginning and flattens towards the end of the plot. If the data is complex, the initial steepness of the curve is smaller. That information can be aggregated into a single parameter – the area under the complexity curve (AUCC). If we express the subset size as the fraction of the whole data set, then the value of the area under the curve becomes limited to the range $[0,1]$ and can be used as an universal measure for comparing complexity of different data sets.

Conditional complexity curve

The complexity curve methodology presented so far deals with the complexity of a data set as a whole. While this approach gives information about data structure, it may assess complexity of the classification task incorrectly. This is because data distribution inside each of the classes may vary greatly from the overall distribution. For example, when the number of classes is larger, or the classes are imbalanced, a random sample large enough to represent the whole data set may be too small to represent some of the classes. To take this into account we introduce conditional complexity curve. We calculate it by splitting each data sample according to the class value and taking the arithmetic mean of the complexities of each sub-sample. Algorithm 2 presents the exact procedure.

Comparison of standard complexity curve and conditional complexity curve for **iris** data set is given by Figure 2. This data set has 3 distinct classes. Our expectation is that estimating conditional distributions for each class would require larger data samples than estimating the overall distribution. Shape of the conditional complexity curve is consistent with this expectation: it is less steep than the standard curve and has larger AUCC value.

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

Algorithm 2 Procedure for calculating conditional complexity curve.

D – original data set, C – number of classes, N – number of subsets, K – number of samples.

1. Transform D with whitening transform and/or ICA to obtain D_I.
2. Split D_I according to the class into $D_I^1, D_I^2, \ldots, D_I^C$.
3. From $D_I^1, D_I^2, \ldots, D_I^C$ estimate probability distributions $P^1, P^2, \ldots, P^C$.
4. For i in $1 \ldots |D_I|$ with a step size $\frac{|D_I|}{N}$:
   (a) For j in $1 \ldots K$:
      i. Draw subset $S_i^j \subseteq D_I$ such that $|S_i^j| = i$.
      ii. Split S_i^j according to the class into $S_i^{j,1}, S_i^{j,2}, \ldots, S_i^{j,C}$.
      iii. From $S_i^{j,1}, S_i^{j,2}, \ldots, S_i^{j,C}$ estimate probability distributions $Q_i^{j,1}, Q_i^{j,2}, \ldots, Q_i^{j,C}$.
      iv. Calculate mean Hellinger distance: $l_i^j = \frac{1}{C} \sum_{k=1}^{C} H^2(P^k, Q_i^{j,k})$.

   (b) Calculate mean m_i and standard error s_i:

$$m_i = \frac{1}{K} \sum_{j=1}^{K} l_i^j \qquad s_i = \sqrt{\frac{1}{K} \sum_{j=1}^{K} \left(m_i - l_i^j\right)^2}$$

Conditional complexity curve is a plot of $m_i \pm s_i$ vs i.

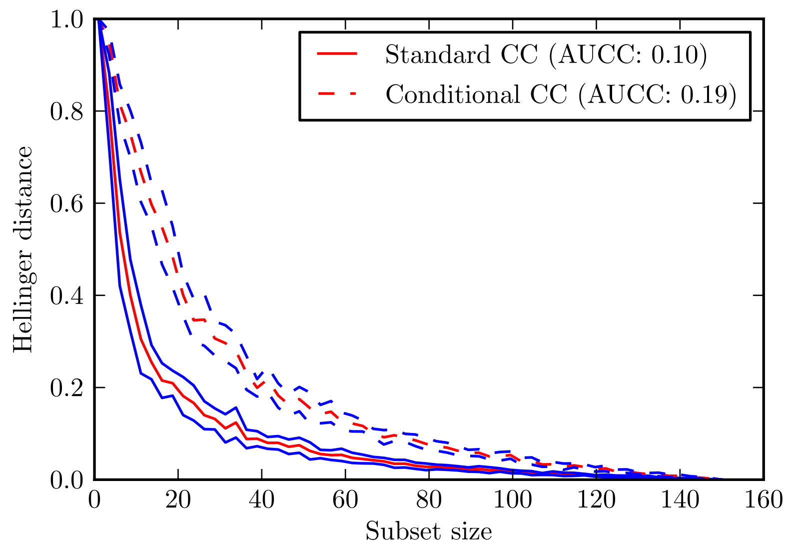

Figure 2. Complexity curve (solid) and conditional complexity curve (dashed) for **iris** data set.

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 3 presents the exact procedure of calculating generalisation curve.

Algorithm 3 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.

Standard learning curve and generalisation curve for the same data and classifier are depicted in Figure 3. 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.

PROPERTIES

To support validity of the proposed method, we perform an in-depth analysis of its properties. We start from purely mathematical analysis giving some intuitions on complexity curve convergence rate and identifying border cases. Then we perform experiments with toy artificial data sets testing basic assumptions behind complexity curve. After that we compare it experimentally with other complexity data measures and show its usefulness in explaining classifier performance.

Mathematical properties

Drawing a random subset S_n from a finite data set D of size N corresponds to sampling without replacement. Let assume that the data set contains k distinct values $\{v_1, v_2, \dots, v_k\}$ occurring with frequencies $P = (p_1, p_2, \dots, p_k)$. $Q_n = (q_1, q_2, \dots, q_k)$ will be a random vector which follows a multivariate hypergeometric distribution.

$$q_i = \frac{1}{n}\sum_{y \in S_n} \mathbf{1}\{y = v_i\}$$

The expected value for any single element is:

$$E[q_i] = p_i$$

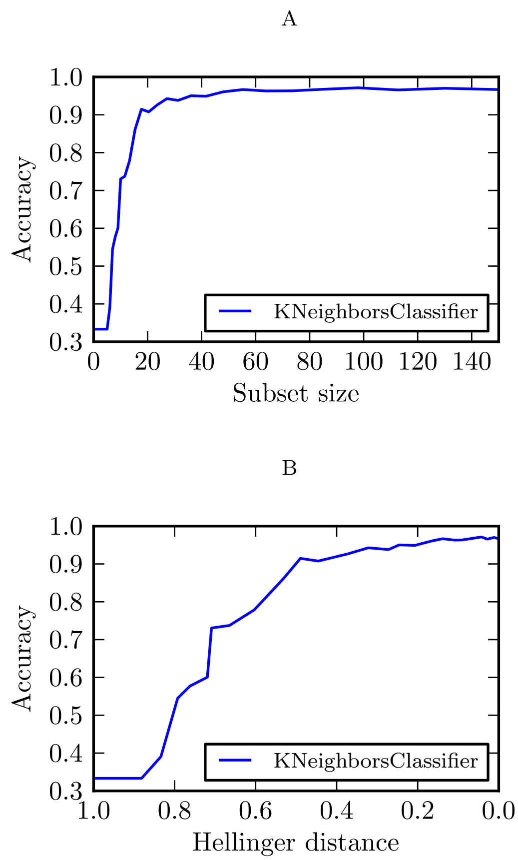

Figure 3. Learning curve (A) and generalisation curve (B) for data set IRIS and k-neighbours classifier ($k = 5$).

The probability of obtaining any specific vector of frequencies:

$$P\left(Q_n = (q_1, q_2, \ldots, q_k)\right) = \frac{\binom{p_1 N}{q_1 n}\binom{p_2 N}{q_2 n}\cdots\binom{p_k N}{q_k n}}{\binom{N}{n}}$$

with $\sum_{i=1}^{k} q_i = 1$.

We will consider the simplest case of discrete probability distribution estimated through frequency counts without using the independence assumption. In such case complexity curve is by definition:

$$CC(n) = E[H^2(P, Q_n)]$$

It is obvious that $CC(N) = 0$ because when $n = N$ we draw all available data. This means that complexity curve always converges. We can ask whether it is possible to say anything about the rate of this convergence. This is the question about the upper bound on the tail of hypergeometric distribution. Such bound is given by Hoeffding-Chvátal inequality (Chvátal, 1979; Skala, 2013). For the univariate case it has the following form:

$$P\left(|q_i - p_i| \geq \delta\right) \leq 2e^{-2\delta^2 n}$$

which generalises to a multivariate case as:

$$P\left(|Q_n - P| \geq \delta\right) \leq 2ke^{-2\delta^2 n}$$

where $|Q_n - P|$ is the total variation distance. Since $H^2(P, Q_n) \leq |Q_n - P|$ this guarantees that complexity curve converges at least as fast.

Now we will consider a special case when $n = 1$. In this situation the multivariate hypergeometric distribution is reduced to a simple categorical distribution P. In such case the expected Hellinger distance is:

$$E[H^2(P, Q_1)] = \sum_{i=1}^{k} \frac{p_i}{\sqrt{2}} \sqrt{\sum_{j=1}^{k} \left(\sqrt{p_j} - \mathbf{1}\{j = k\}\right)^2}$$

$$= \sum_{i=1}^{k} \frac{p_i}{\sqrt{2}} \sqrt{1 - p_i + (\sqrt{p_i} - 1)^2} = \sum_{i=1}^{k} p_i \sqrt{1 - \sqrt{p_i}}$$

This corresponds to the first point of complexity curve and determines its overall steepness.

Theorem: $E[H^2(P, Q_1)]$ is maximal for a given k when P is an uniform categorical distribution over k categories, i.e.:

$$E[H^2(P, Q_1)] = \sum_{i=1}^{k} p_i \sqrt{1 - \sqrt{p_i}} \leq \sqrt{1 - \sqrt{\frac{1}{k}}}$$

Proof: We will consider an arbitrary distribution P and the expected Hellinger distance $E[H^2(P, Q_1)]$. We can modify this distribution by choosing two states l and k occurring with probabilities p_l and p_k such as that $p_l - p_k$ is maximal among all pairs of states. We will redistribute the probability mass between the two states creating a new distribution P'. The expected Hellinger distance for the distribution P' will be:

$$E[H^2(P', Q_1)] = \sum_{i=1, i \neq k, i \neq l}^{k} p_i \sqrt{1 - \sqrt{p_i}} + a\sqrt{1 - \sqrt{a}} + (p_k + p_l - a)\sqrt{1 - \sqrt{p_k + p_l - a}}$$

where a and $p_k + p_l - a$ are new probabilities of the two states in P'. We will consider a function $f(a) = a\sqrt{1 - \sqrt{a}} + (p_k + p_l - a)\sqrt{1 - \sqrt{p_k + p_l}}$ and look for its maxima.

$$\frac{\partial f(x)}{\partial a} = -\sqrt{1 - \sqrt{p_k + p_l - a}} + \frac{\sqrt{p_k + p_l - a}}{4\sqrt{1 - \sqrt{p_k + p_l - a}}} + \sqrt{1 - \sqrt{a}} - \frac{\sqrt{a}}{4\sqrt{1 - \sqrt{a}}}$$

The derivative is equal to 0 if and only if $a = \frac{p_k+p_l}{2}$. We can easily see that:

$$f(0) = f(p_k + p_l) = (p_k + p_l)\sqrt{1-\sqrt{p_k+p_l}} < (p_k + p_l)\sqrt{1-\sqrt{\frac{p_k+p_l}{2}}}$$

This means that $f(a)$ reaches its maximum for $a = \frac{p_k+p_l}{2}$. From that we can conclude that for any distribution P if we produce distribution P' by redistributing probability mass between two states equally the following holds:

$$E[H^2(P',Q_1)] \geq E[H^2(P,Q_1)]$$

If we repeat such redistribution arbitrary number of times the outcome distribution converges to uniform distribution. This proves that the uniform distribution leads to the maximal expected Hellinger distance for a given number of states.

Theorem: Increasing the number of categories by dividing an existing category into two new categories always increases the expected Hellinger distance, i.e.

$$\sum_{i=1}^{k} p_i\sqrt{1-\sqrt{p_i}} \leq \sum_{i=1, i\neq l}^{k} p_i\sqrt{1-\sqrt{p_i}} + a\sqrt{1-\sqrt{a}} + (p_l - a)\sqrt{1-\sqrt{p_l - a}}$$

Proof: Without the loss of generality we can assume that $a < 0.5p_l$. We can subtract terms occurring on both sides of the inequality obtaining:

$$p_l\sqrt{1-\sqrt{p_l}} \leq a\sqrt{1-\sqrt{a}} + (p_l - a)\sqrt{1-\sqrt{p_l - a}}$$

$$p_l\sqrt{1-\sqrt{p_l}} \leq a\sqrt{1-\sqrt{a}} + p_l\sqrt{1-\sqrt{p_l - a}} - a\sqrt{1-\sqrt{p_l - a}}$$

$$p_l\sqrt{1-\sqrt{p_l}} + a\sqrt{1-\sqrt{p_l - a}} \leq a\sqrt{1-\sqrt{a}} + p_l\sqrt{1-\sqrt{p_l - a}}$$

Now we can see that:

$$p_l\sqrt{1-\sqrt{p_l}} \leq p_l\sqrt{1-\sqrt{p_l - a}}$$

and

$$a\sqrt{1-\sqrt{p_l - a}} \leq a\sqrt{1-\sqrt{a}}$$

which concludes the proof.

From the properties stated by these two theorems we can gain some intuitions about complexity curves in general. First, by looking at the formula for the uniform distribution $E[H^2(P,Q_1)] = \sqrt{1-\sqrt{\frac{1}{k}}}$ we can see that when $k = 1$ $E[H^2(P,Q_1)] = 0$ and when $k \to \infty$ $E[H^2(P,Q_1)] \to 1$. The complexity curve will be less steep if the variables in the data set take multiple values and each value occurs with equal probability. This is consistent with our intuition: we need a larger sample to cover such space and collect information. For smaller number of distinct values or distributions with mass concentrated mostly in a few points smaller sample will be sufficient to represent most of the information in the data set.

Complexity curve and the performance of an unbiased model

To confirm validity of the assumptions behind complexity curve we performed experiments with artificial data generated according to the known model. Error of the corresponding classifier trained on such data does not contain bias component, so it is possible to observe if variance error component is indeed upper bounded by the complexity curve. We used the same scenario as when calculating the complexity curve: classifiers were trained on random subsets and tested on the whole data set. We matched first and last points of complexity curve and learning curve and observed their relation in between.

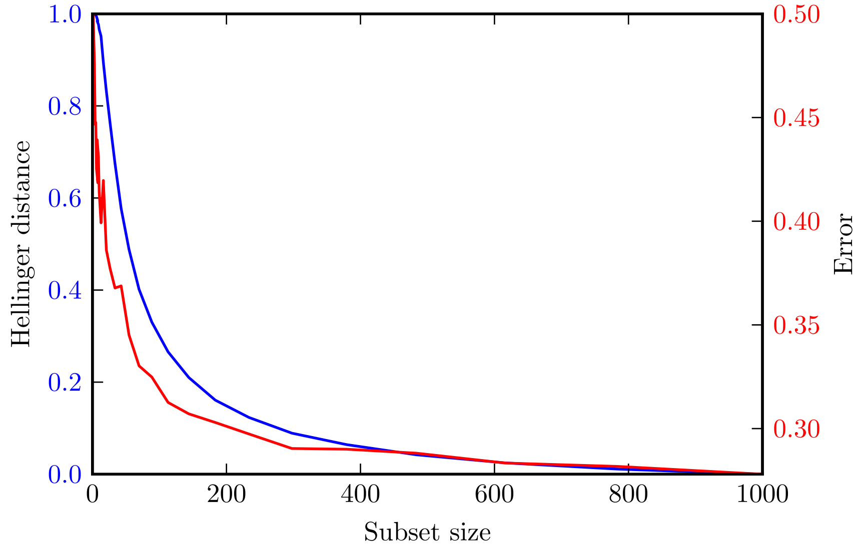

Figure 4. Complexity curve and learning curve of the logistic regression on the **logit** data.

The first kind of data followed the logistic model (**logit** data set). Matrix X (1000 observations, 12 attributes) contained values drawn from normal distribution with mean 0 and standard deviation 1. Class vector Y was defined as follows:

$$P(Y|x) = \frac{e^{\beta' x}}{(1+e^{\beta' x})}$$

where $\beta = (0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0, 0, 0, 0, 0, 0)$. All attributes were independent and conditionally independent. Since Y values were not deterministic, there was some noise present – classification error of the logistic regression classifier trained and tested on the full data set was larger than zero.

Figure 4 presents complexity curve and adjusted error of logistic regression for the generated data. After ignoring noise error component, we can see that the variance error component is indeed upper bounded by the complexity curve.

Different kind of artificial data represented multidimensional space with parallel stripes in one dimension (**stripes** data set). It consisted of X matrix with 1000 observations and 10 attributes drawn from an uniform distribution on range $[0, 1)$. Class values Y dependent only on value of one of the attributes: for values lesser than 0.25 or greater than 0.75 the class was 1, for other values the class was 0. This kind of relation can be naturally modelled by a decision tree, and all the attributes are again independent and conditionally independent.

Figure 5 presents complexity curve and adjusted error of decision tree classifier on the generated data. Once again the assumptions of complexity curve methodology are satisfied and the complexity curve is indeed an upper bound for the error.

What would happen if the attribute conditional independence assumption was broken? To answer this question we generated another type of data modelled after multidimensional chessboard (**chessboard** data set). X matrix contained 1000 observations and 2, 3 attributes drawn from an uniform distribution on range $[0, 1)$. Class vector Y had the following values:

$$\begin{cases} 0 & \text{if } \Sigma_{i=0}^{m} \left\lfloor \frac{x_i}{s} \right\rfloor \text{ is even} \\ 1 & \text{otherwise} \end{cases}$$

where s was a grid step in our experiments set to 0.5. There is clearly strong attribute dependence, but since all parts of decision boundary are parallel to one of the attributes this kind of data can be modelled with a decision tree with no bias.

Figure 6 presents complexity curves and error curves for different dimensionalities of **chessboard** data. Indeed here classification error becomes larger than indicated by complexity curve. The more

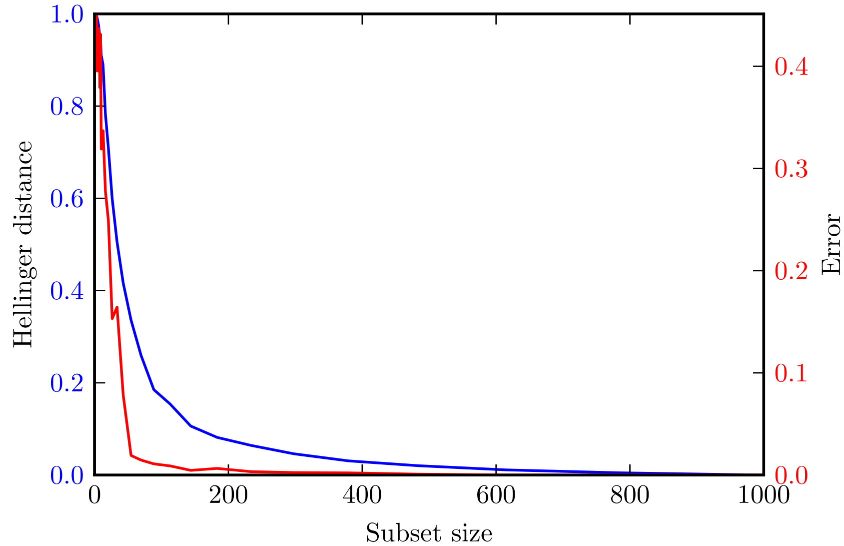

Figure 5. Complexity curve and learning curve of the decision tree on the **stripes** data.

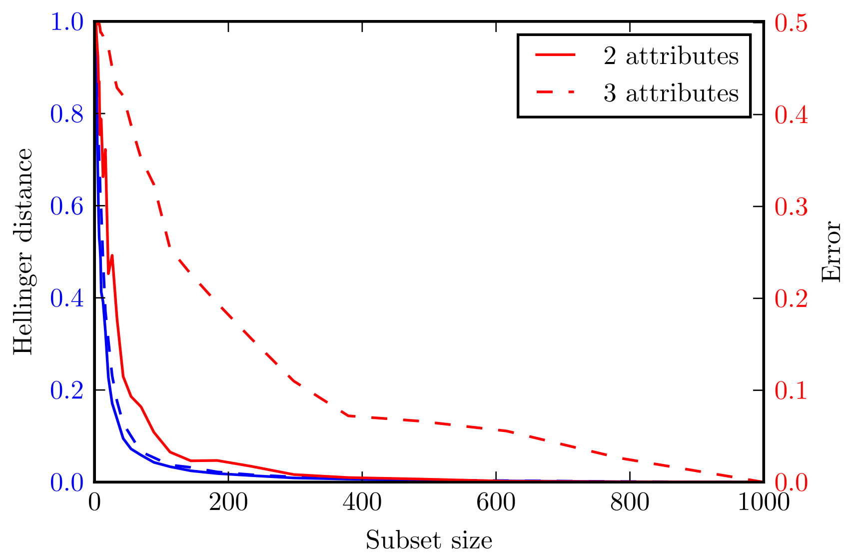

Figure 6. Complexity curve and learning curve of the decision tree on the **chessboard** data.

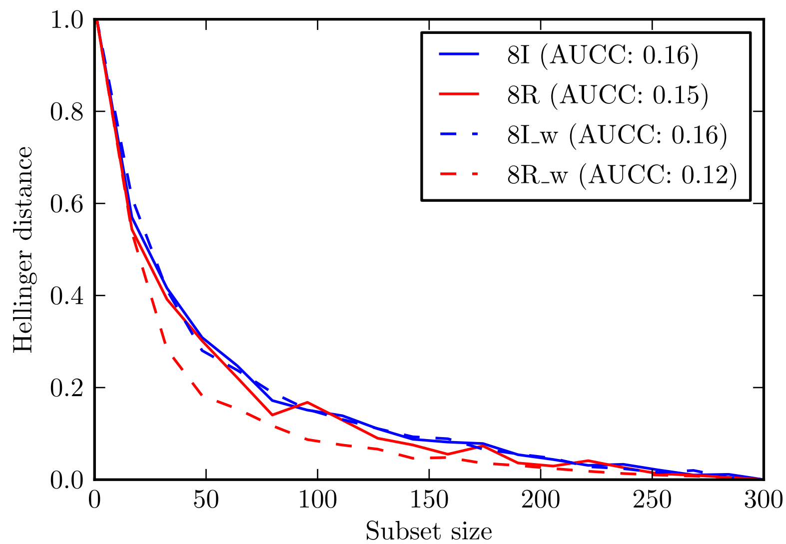

Figure 7. Complexity curves for whitened data (dashed lines) and not whitened data (solid lines). Areas under the curves are given in the legend. 8I – set of 8 independent random variables with Student's t distribution. 8R – one random variable with Student's t distribution repeated 8 times. 8I_w – whitened 8I. 8R_w – whitened 8R.

dimensions, the more dependencies between attributes violating complexity curve assumptions. For 3 dimensional chessboard the classification problem becomes rather hard and the observed error decreases slowly, but the complexity curve remains almost the same as for 2 dimensional case.

Results of experiments with controlled artificial data sets are consistent with our theoretical expectations. Basing on them we can introduce a general interpretation of the difference between complexity curve and learning curve: learning curve below the complexity curve is an indication that the algorithm is able to build a good model without sampling the whole domain, limiting the variance error component. On the other hand, learning curve above the complexity curve is an indication that the algorithm includes complex attributes dependencies in the constructed model, promoting the variance error component.

Impact of whitening and ICA

To evaluate the impact of the proposed preprocessing techniques (whitening and ICA – Independent Component Analysis) on complexity curves we performed experiments with artificial data. In the first experiment we generated two data sets of 300 observations and with 8 attributes distributed according to Student's t distribution with 1.5 degrees of freedom. In one data set all attributes were independent, in the other the same attribute was repeated 8 times. To both sets small Gaussian noise was added. Figure 7 shows complexity curves calculated before and after whitening transform. We can see that whitening had no significant effect on the complexity curve of the independent set. In the case of the dependent set complexity curve calculated after whitening decreases visibly faster and the area under the curve is smaller. This is consistent with our intuitive notion of complexity: data set with repeated attributes should be significantly less complex.

In the second experiment two data sets with 100 observations and 4 attributes were generated. The first data set was generated from the continuous uniform distribution on interval $[0,2]$, the second one from the discrete (categorical) uniform distribution on the same interval. To both sets small Gaussian noise was added. Figure 8 presents complexity curves for original, whitened and ICA-transformed data. Among the original data sets the intuitive notion of complexity is preserved: area under the complexity curve for categorical data is smaller. The difference disappears for the whitened data but is again visible in the ICA-transformed data.

These simple experiments are by no means exhaustive but they confirm usefulness of the chosen signal processing techniques (data whitening and Independent Component Analysis) in complexity curve analysis.

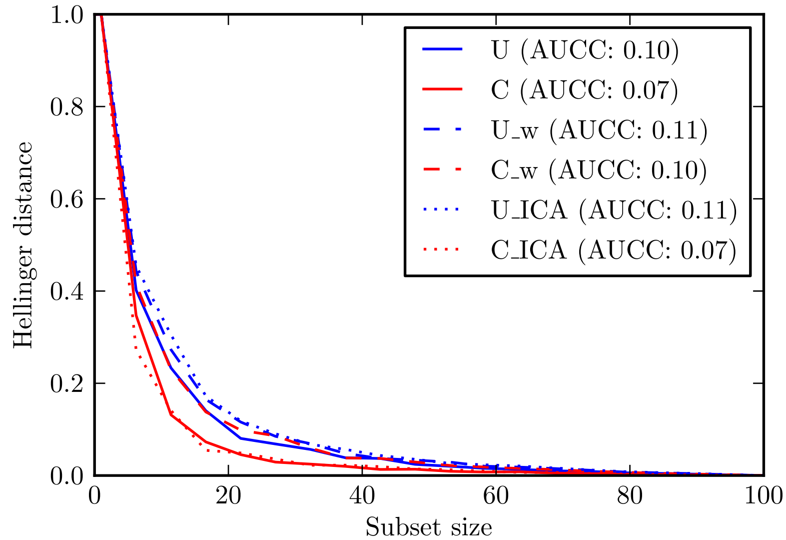

Figure 8. Complexity curves for whitened data (dashed lines), not whitened data (solid lines) and ICA-transformed data (dotted lines). Areas under the curves are given in the legend. U – data sampled from uniform distribution. C – data sampled from categorical distribution. U_w – whitened U. C_w – whitened C. U_ICA – U_w after ICA. C_ICA – C_w after ICA.

Complexity curve variability and outliers

Complexity curve is based on the expected Hellinger distance and the estimation procedure includes some variance. The natural assumption is that the variability caused by the sample size is greater than the variability resulting from a specific composition of a sample. Otherwise averaging over samples of the same size would not be meaningful. This assumption is already present in standard learning curve methodology, when classifier accuracy is plotted against training set size. We expect that the exact variability of the complexity curve will be connected with the presence of outliers in the data set. Such influential observations will have a huge impact depending whether they will be included in a sample or not.

To verify whether these intuitions were true, we constructed two new data sets by introducing artificially outliers to WINE data set. In WINE001 we modified 1% of values by multiplying them by a random number from range $(-10, 10)$. In WINE005 5% of values were modified in such manner.

Figure 9 presents conditional complexity curves for all three data sets. WINE001 curve has indeed a higher variance and is less regular than WINE curve. WINE005 curve is characterised not only by a higher variance but also by a larger AUCC value. This means that adding so much noise increased the overall complexity of the data set significantly.

The result support our hypothesis that large variability of complexity curve signify an occurrence of highly influential observations in the data set. This makes complexity curve a valuable diagnostic tool for such situations. However, it should be noted that our method is unable to distinguish between important outliers and plain noise. To obtain this kind of insight one has to employ different methods.

Comparison with other complexity measures

The set of data complexity measures developed by Ho and Basu (2002) and extended by Ho et al. (2006) continues to be used in experimental studies to explain performance of various classifiers (Díez-Pastor et al., 2015; Mantovani et al., 2015). We decided to compare experimentally complexity curve with those measures. Descriptions of the measures used are given in Table 1.

According to our hypothesis conditional complexity curve should be robust in the context of class imbalance. To demonstrate this property we used for the comparison 88 imbalanced data sets used previously in the study by Díez-Pastor et al. (2015). These data sets come originally from HDDT (Cieslak et al., 2011) and KEEL (Alcalá et al., 2010) repositories. We selected only binary classification problems. The list of data sets with their properties is presented as Tables 2, 3.

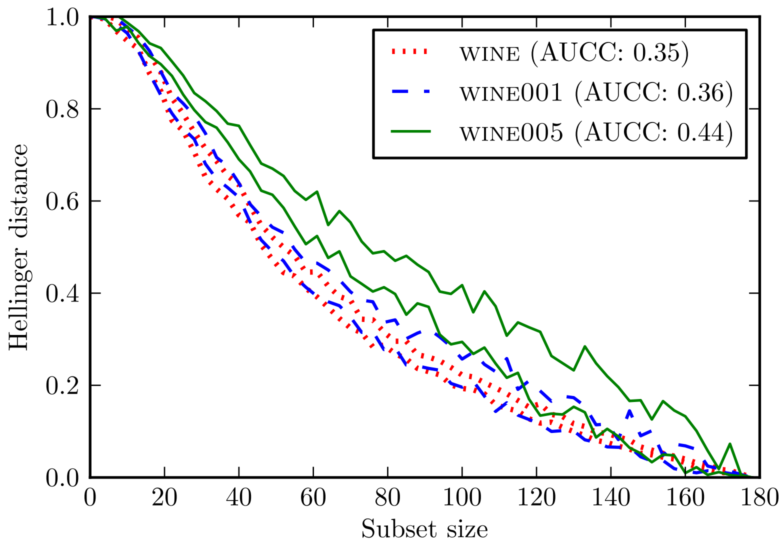

Figure 9. Complexity curves for WINE and its counterparts with introduced outliers. For the sake of clarity only contours were drawn.

Id	Description
F1	Maximum Fisher's discriminant ratio
F1v	Directional-vector maximumline Fisher's discriminant ratio
F2	Overlap of the per-classline bounding boxes
F3	Maximum individual feature efficiency
F4	Collective feature efficiency
L1	Minimized sum of the error distance of a linear classifier
L2	Training error of a linear classifier
L3	Nonlinearity of a linear classifier
N1	Fraction of points on the class boundary
N2	Ratio of average intra/inter class nearest neighbor distance
N3	Leave-one-out error rate of the one-nearest neighbor classifier
N4	Nonlinearity of the one-nearest neighbor classifier
T1	Fraction of maximum covering spheres
T2	Average number of points per dimension

Table 1. Data complexity measures used in experiments.

Data set	Instances	Attributes	Classes	Imbalance ratio
HDDT breast-y	286	9	2	2.36
HDDT compustat	13657	20	2	25.26
HDDT covtype	38500	10	2	13.02
HDDT credit-g	1000	20	2	2.33
HDDT estate	5322	12	2	7.37
HDDT german-numer	1000	24	2	2.33
HDDT heart-v	200	13	2	2.92
HDDT hypo	3163	25	2	19.95
HDDT ism	11180	6	2	42.00
HDDT letter	20000	16	2	24.35
HDDT oil	937	49	2	21.85
HDDT page	5473	10	2	8.77
HDDT pendigits	10992	16	2	8.63
HDDT phoneme	5404	5	2	2.41
HDDT PhosS	11411	480	2	17.62
HDDT satimage	6430	36	2	9.29
HDDT segment	2310	19	2	6.00

Table 2. Properties of HDDT data sets used in experiments.

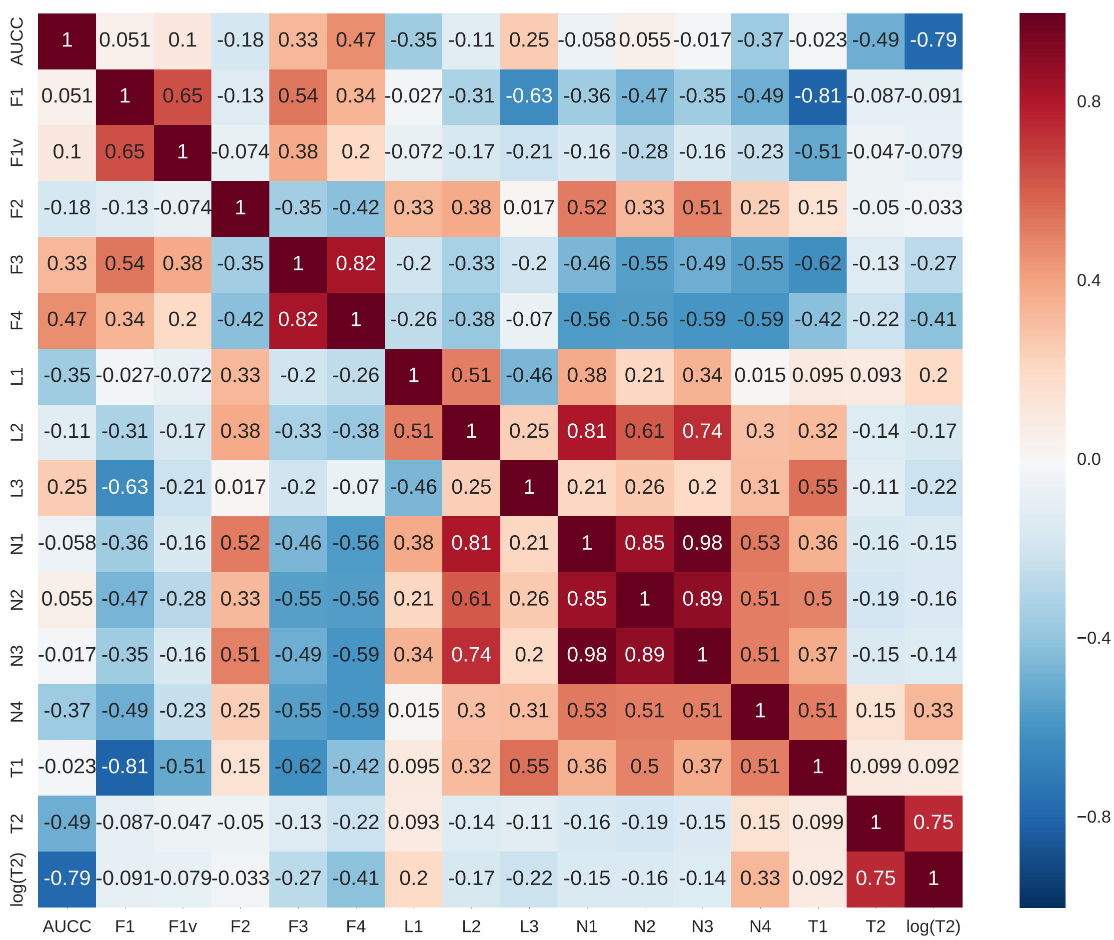

Figure 10. Pearson's correlations between complexity measures.

Data set	Instances	Attributes	Classes	Imbalance ratio
KEEL ABALONE19	4174	8	2	129.44
KEEL ABALONE9-18	731	8	2	16.40
KEEL CLEVELAND-0_VS_4	177	13	2	12.62
KEEL ECOLI-0-1-3-7_VS_2-6	281	7	2	39.14
KEEL ECOLI-0-1-4-6_VS_5	280	6	2	13.00
KEEL ECOLI-0-1-4-7_VS_2-3-5-6	336	7	2	10.59
KEEL ECOLI-0-1-4-7_VS_5-6	332	6	2	12.28
KEEL ECOLI-0-1_VS_2-3-5	244	7	2	9.17
KEEL ECOLI-0-1_VS_5	240	6	2	11.00
KEEL ECOLI-0-2-3-4_VS_5	202	7	2	9.10
KEEL ECOLI-0-2-6-7_VS_3-5	224	7	2	9.18
KEEL ECOLI-0-3-4-6_VS_5	205	7	2	9.25
KEEL ECOLI-0-3-4-7_VS_5-6	257	7	2	9.28
KEEL ECOLI-0-3-4_VS_5	200	7	2	9.00
KEEL ECOLI-0-4-6_VS_5	203	6	2	9.15
KEEL ECOLI-0-6-7_VS_3-5	222	7	2	9.09
KEEL ECOLI-0-6-7_VS_5	220	6	2	10.00
KEEL ECOLI-0_VS_1	220	7	2	1.86
KEEL ECOLI1	336	7	2	3.36
KEEL ECOLI2	336	7	2	5.46
KEEL ECOLI3	336	7	2	8.60
KEEL ECOLI4	336	7	2	15.80
KEEL GLASS-0-1-2-3_VS_4-5-6	214	9	2	3.20
KEEL GLASS-0-1-4-6_VS_2	205	9	2	11.06
KEEL GLASS-0-1-5_VS_2	172	9	2	9.12
KEEL GLASS-0-1-6_VS_2	192	9	2	10.29
KEEL GLASS-0-1-6_VS_5	184	9	2	19.44
KEEL GLASS-0-4_VS_5	92	9	2	9.22
KEEL GLASS-0-6_VS_5	108	9	2	11.00
KEEL GLASS0	214	9	2	2.06
KEEL GLASS1	214	9	2	1.82
KEEL GLASS2	214	9	2	11.59
KEEL GLASS4	214	9	2	15.46
KEEL GLASS5	214	9	2	22.78
KEEL GLASS6	214	9	2	6.38
KEEL HABERMAN	306	3	2	2.78
KEEL IRIS0	150	4	2	2.00
KEEL LED7DIGIT-0-2-4-5-6-7-8-9_VS_1	443	7	2	10.97
KEEL NEW-THYROID1	215	5	2	5.14
KEEL NEW-THYROID2	215	5	2	5.14
KEEL PAGE-BLOCKS-1-3_VS_4	472	10	2	15.86
KEEL PIMA	768	8	2	1.87
KEEL SHUTTLE-C0-VS-C4	1829	9	2	13.87
KEEL SHUTTLE-C2-VS-C4	129	9	2	20.50
KEEL VEHICLE0	846	18	2	3.25
KEEL VEHICLE1	846	18	2	2.90
KEEL VEHICLE2	846	18	2	2.88
KEEL VEHICLE3	846	18	2	2.99
KEEL VOWEL0	988	13	2	9.98
KEEL WISCONSIN	683	9	2	1.86
KEEL YEAST-0-2-5-6_VS_3-7-8-9	1004	8	2	9.14
KEEL YEAST-0-2-5-7-9_VS_3-6-8	1004	8	2	9.14
KEEL YEAST-0-3-5-9_VS_7-8	506	8	2	9.12
KEEL YEAST-0-5-6-7-9_VS_4	528	8	2	9.35
KEEL YEAST-1-2-8-9_VS_7	947	8	2	30.57
KEEL YEAST-1-4-5-8_VS_7	693	8	2	22.10
KEEL YEAST-1_VS_7	459	7	2	14.30
KEEL YEAST-2_VS_4	514	8	2	9.08
KEEL YEAST-2_VS_8	482	8	2	23.10
KEEL YEAST1	1484	8	2	2.46
KEEL YEAST3	1484	8	2	8.10
KEEL YEAST4	1484	8	2	28.10
KEEL YEAST5	1484	8	2	32.73
KEEL YEAST6	1484	8	2	41.40

Table 3. Properties of KEEL data sets used in experiments.

	AUCC	log T2
LDA	0.0489	0.0227
Logistic regression	-0.0539	**0.1103**
Naive Bayes	-0.0792	0.0889
1-NN	**-0.1256**	0.0772
3-NN	**-0.1311**	0.0863
5-NN	**-0.1275**	0.0952
10-NN	**-0.1470**	0.1225
15-NN	**-0.1730**	0.1584
20-NN	-0.1842	0.1816
25-NN	-0.1859	0.1902
30-NN	-0.1969	0.2059
35-NN	-0.2249	0.2395
Decision tree $d = 1$	0.0011	-0.0624
Decision tree $d = 3$	**-0.1472**	0.1253
Decision tree $d = 5$	-0.1670	0.1690
Decision tree $d = 10$	**-0.1035**	0.0695
Decision tree $d = 15$	**-0.0995**	0.0375
Decision tree $d = 20$	**-0.0921**	0.0394
Decision tree $d = 25$	**-0.0757**	0.0298
Decision tree $d = 30$	**-0.0677**	0.0227
Decision tree $d = \text{inf}$	**-0.0774**	0.0345

Table 4. Pearson's correlations coefficients between classifier AUC ROC performances and complexity measures. Values larger than 0.22 or smaller than -0.22 are significant at $\alpha = 0.05$ significance level.

For each data set we calculated area under the complexity curve using the previously described procedure and the values of other data complexity measures using DCOL software (Orriols-Puig et al., 2010). Pearson's correlation was then calculated for all the measures. As T2 measure seemed to have non-linear characteristics destroying the correlation additional column log T2 was added to comparison. Results are presented as Figure 10. Clearly AUCC is mostly correlated with log T2 measure. This is to be expected as both measures are concerned with sample size in relation to attribute structure. The difference is that T2 takes into account only the number of attributes while AUCC considers also the complexity of distributions of the individual attributes. Correlations of AUCC with other measures are much lower and it can be assumed that they capture different aspects of data complexity and may be potentially complementary.

The next step was to show that information captured by AUCC is useful for explaining classifier performance. In order to do so we trained a number of different classifiers on the 81 benchmark data sets and evaluated their performance using random train-test split with proportion 0.5 repeated 10 times. The performance measure used was the area under ROC curve. We selected three linear classifiers – naïve Bayes with gaussian kernel, linear discriminant analysis (LDA) and logistic regression – and two families of non-linear classifiers of varying complexity: k-nearest neighbour classifier (k-NN) with different values of parameter k and decision tree (CART) with the limit on maximal tree depth. The intuition was as follows: the linear classifiers do not model attributes interdependencies, which is in line with complexity curve assumptions. Selected non-linear classifiers on the other hand are – depending on the parametrisation – more prone to variance error, which should be captured by complexity curve.

Correlations between AUCC, log T2, and classifier performance are presented in Table 4. Most of the correlations are weak and do not reach statistical significance, however some general tendencies can be observed. As can be seen, AUC ROC scores of linear classifiers have very little correlation with AUCC and log T2. This may be explained by the high-bias and low-variance nature of these classifiers: they are not strongly affected by data scarcity but their performance depends on other factors. This is especially true for LDA classifier, which has the weakest correlation among linear classifiers.

In k-NN classifier complexity depends on k parameter: with low k values it is more prone to variance

	AUCC	log T2
LDA - Logistic regression	0.2026	-0.2025
LDA - Naive Bayes	**0.2039**	-0.1219
LDA - 1-NN	**0.2278**	-0.0893
LDA - 3-NN	**0.2482**	-0.1063
LDA - 5-NN	**0.2490**	-0.1210
LDA - 10-NN	**0.2793**	-0.1609
LDA - 15-NN	**0.3188**	-0.2148
LDA - 20-NN	**0.3365**	-0.2510
LDA - 25-NN	**0.3392**	-0.2646
LDA - 30-NN	**0.3534**	-0.2868
LDA - 35-NN	**0.3798**	-0.3259
LDA - Decision tree $d = 1$	0.0516	**0.1122**
LDA - Decision tree $d = 3$	**0.3209**	-0.1852
LDA - Decision tree $d = 5$	**0.3184**	-0.2362
LDA - Decision tree $d = 10$	**0.2175**	-0.0838
LDA - Decision tree $d = 15$	**0.2146**	-0.0356
LDA - Decision tree $d = 20$	**0.2042**	-0.0382
LDA - Decision tree $d = 25$	**0.1795**	-0.0231
LDA - Decision tree $d = 30$	**0.1636**	-0.0112
LDA - Decision tree $d = \text{inf}$	**0.1809**	-0.0303

Table 5. Pearson's correlations coefficients between classifier AUC ROC performances relative to LDA performance and complexity measures. Values larger than 0.22 or smaller than -0.22 are significant at $\alpha = 0.05$ significance level.

error, with larger k it is prone to bias if the sample size is not large enough (Domingos, 2000). Both AUCC and log T2 seem to capture the effect of sample size in case of large k value well (correlations -0.2249 and 0.2395 for 35-NN). However, for $k = 1$ the correlation with AUCC is stronger (-0.1256 vs 0.0772).

Depth parameter in decision tree also regulates complexity: the larger the depth the more classifier is prone to variance error and less to bias error. This suggests that AUCC should be more strongly correlated with performance of deeper trees. On the other hand, complex decision trees explicitly model attribute interdependencies ignored by complexity curve, which may weaken the correlation. This is observed in the obtained results: for a decision stub (tree of depth 1), which is low-variance high-bias classifier, correlation with AUCC and log T2 is very weak. For $d = 3$ and $d = 5$ it becomes visibly stronger, and then for larger tree depth it again decreases. It should be noted that with large tree depth, as with small k values in k-NN, AUCC has stronger correlation with the classifier performance than log T2.

A slightly more sophisticated way of applying data complexity measures is an attempt to explain classifier performance relative to some other classification method. In our experiments LDA is a good candidate for reference method since it is simple, has low variance and is not correlated with either AUCC or log T2. Table 5 presents correlations of both measures with classifier performance relative to LDA. Here we can see that correlations for AUCC are generally higher than for log T2 and reach significance for the majority of classifiers. Especially in the case of decision tree AUCC explains relative performance better than log T2 (correlation 0.1809 vs -0.0303 for $d = \text{inf}$).

Results of the presented correlation analyses demonstrate the potential of complexity curve to complement the existing complexity measures in explaining classifier performance. As expected from theoretical considerations, there is a relation between how well AUCC correlates with classifier performance and the classifier's position in bias-variance spectrum. It is worth noting that despite the attribute independence assumption of complexity curve method it proved useful for explaining performance of complex non-linear classifiers.

Data set	Instances	Attributes	Classes	Source
ADENOCARCINOMA	76	9868	2	Ramaswamy et al. (2003)
BREAST2	77	4769	2	van 't Veer et al. (2002)
BREAST3	95	4869	2	van 't Veer et al. (2002)
COLON	62	2001	2	Alon et al. (1999)
LEUKEMIA	38	3052	2	Golub (1999)
LYMPHOMA	62	4026	2	Alizadeh et al. (2000)
PROSTATE	38	3052	2	Singh et al. (2002)

Table 6. Properties of microarray data sets used in experiments.

Dataset	AUCC	1-NN	5-NN	DT d-10	DT d-inf	LDA	NB	LR
ADENOCARCINOMA	0.9621	0.6354	0.5542	0.5484	0.5172	0.6995	0.5021	0.7206
BREAST2	0.9822	0.5869	0.6572	0.6012	0.6032	0.6612	0.5785	0.6947
BREAST3	0.9830	0.6788	0.7344	0.6274	0.6131	0.7684	0.6840	0.7490
COLON	0.9723	0.7395	0.7870	0.6814	0.6793	0.7968	0.5495	0.8336
LEUKEMIA	0.9611	1.0000	0.9985	0.7808	0.8715	0.9615	0.8300	1.0000
LYMPHOMA	0.9781	0.9786	0.9976	0.8498	0.8660	0.9952	0.9700	1.0000
PROSTATE	0.9584	0.5931	0.4700	0.4969	0.5238	0.4908	0.5000	0.4615

Table 7. Areas under conditional complexity curve (AUCC) for microarray data sets along AUC ROC values for different classifiers. k-NN – k-nearest neighbour, DT – CART decision tree, LDA – linear discriminant analysis, NB – naïve Bayes, LR – logistic regression.

Large p, small n problems

There is a special category of machine learning problems in which the number of attributes p is large with respect to the number of samples n, perhaps even order of magnitudes larger. Many important biological data sets, most notably data from microarray experiments, fall into this category (Johnstone and Titterington, 2009). To test how our complexity measure behaves in such situations, we calculated AUCC scores for a few microarray data sets and compared them with AUC ROC scores of some simple classifiers. Classifiers were evaluated as in the previous section. Detailed information about the data sets is given by Table 6.

Results of the experiment are presented in Table 7. As expected, with the number of attributes much larger than the number of observations data is considered by our metric as extremely scarce – values of AUCC are in all cases above 0.95. On the other hand, AUC ROC classification performance is very varied between data sets with scores approaching or equal to 1.0 for LEUKEMIA and LYMPHOMA data sets, and scores around 0.5 baseline for PROSTATE. This is because despite the large number of dimensions the form of the optimal decision function can be very simple, utilising only a few of available dimensions. Complexity curve does not consider the shape of decision boundary at all and thus does not reflect differences in classification performance.

From this analysis we concluded that complexity curve is not a good predictor of classifier performance for data sets containing a large number of redundant attributes, as it does not differentiate between important and unimportant attributes. The logical way to proceed in such case would be to perform some form of feature selection or dimensionality reduction on the original data, and then calculate complexity curve in the reduced dimensions.

APPLICATIONS

Interpreting complexity curves

In order to prove the practical applicability of the proposed methodology, and show how complexity curve plot can be interpreted, we performed experiments with six simple data sets from UCI Machine Learning Repository (Frank and Asuncion, 2010). The sets were chosen only as illustrative examples. They have

	Instances	Attributes	Classes
UCI IRIS	150	4	3
UCI CAR	1728	6	4
UCI MONKS-1	556	6	2
UCI WINE	178	13	3
UCI BREAST-CANCER-WISCONSIN (BCW)	683	9	2
UCI GLASS	214	9	7

Table 8. Basic properties of the benchmark data sets.

no missing values and represent only classification problems, not regression ones. Basic properties of the data sets are given in Table 8. For each data set we calculated conditional complexity curve, as it should capture data properties in the context of classification better than standard complexity curve. The curves are presented in Figure 11.

Shape of the complexity curve portrays the learning process. The initial examples are the most important since there is a huge difference between having some information and having no information at all. After some point including additional examples still improves probability estimation, but does not introduce such a dramatic change.

Looking at the individual graphs, it is now possible to compare complexity of different sets. From the sets considered, MONKS-1 and CAR are dense data sets with a lot of instances and medium number of attributes. The information they contain can be to a large extend recovered from relatively small subsets. Such sets are natural candidates for data pruning. On the other hand, WINE and GLASS are small data sets with a larger number of attributes or classes – they can be considered complex, with no redundant information.

Besides the slope of the complexity curve we can also analyse its variability. We can see that the shape of WINE complexity curve is very regular with small variance in each point, while the GLASS curve displays much higher variance. This mean that the observations in GLASS data set are more diverse and some observations (or their combinations) are more important for representing data structure than the other.

Data pruning with complexity curves

The problem of data pruning in the context of machine learning is defined as reducing the size of training sample in order to reduce classifier training time and still achieve satisfactory performance. It becomes extremely important as the data grows and a) does not fit the memory of a single machine, b) training times of more complex algorithms become very long.

A classic method for performing data pruning is progressive sampling – training the classifier on data samples of increasing size as long as its performance increases. Provost et al. (1999) analysed various schedules for progressive sampling and recommended geometric sampling, in which sample size is multiplied by a specified constant in each iteration, as the reasonable strategy in most cases. Geometric sampling uses samples of sizes $a^i n_0$, where n_0 – initial sample size, a – multiplier, i – iteration number.

In our method instead of training classifier on the drawn data sample we are probing the complexity curve. We are not detecting convergence of classifier accuracy, but searching for a point on the curve corresponding to some reasonably small Hellinger distance value, e.g. 0.005. This point designates the smallest data subset which still contains the required amount of information.

In this setting we were not interested in calculating the whole complexity curve but just in finding the minimal data subset, which still contains most of the original information. The search procedure should be as fast as possible, since the goal of the data pruning is to save time spent on training classifiers. To comply with these requirements we constructed a criterion function of the form $f(x) = H^2(G_x, D) - t$, where D denotes a probability distribution induced by the whole data set, G_x a distribution induced by random subset of size x and t is the desired Hellinger distance. We used classic Brent method (Brent, 1973) to find a root of the criterion function. In this way data complexity was calculated only for the points visited by Brent's algorithm. To speed up the procedure even further we used standard complexity curve instead of the conditional one and settled for whitening transform as the only preprocessing technique.

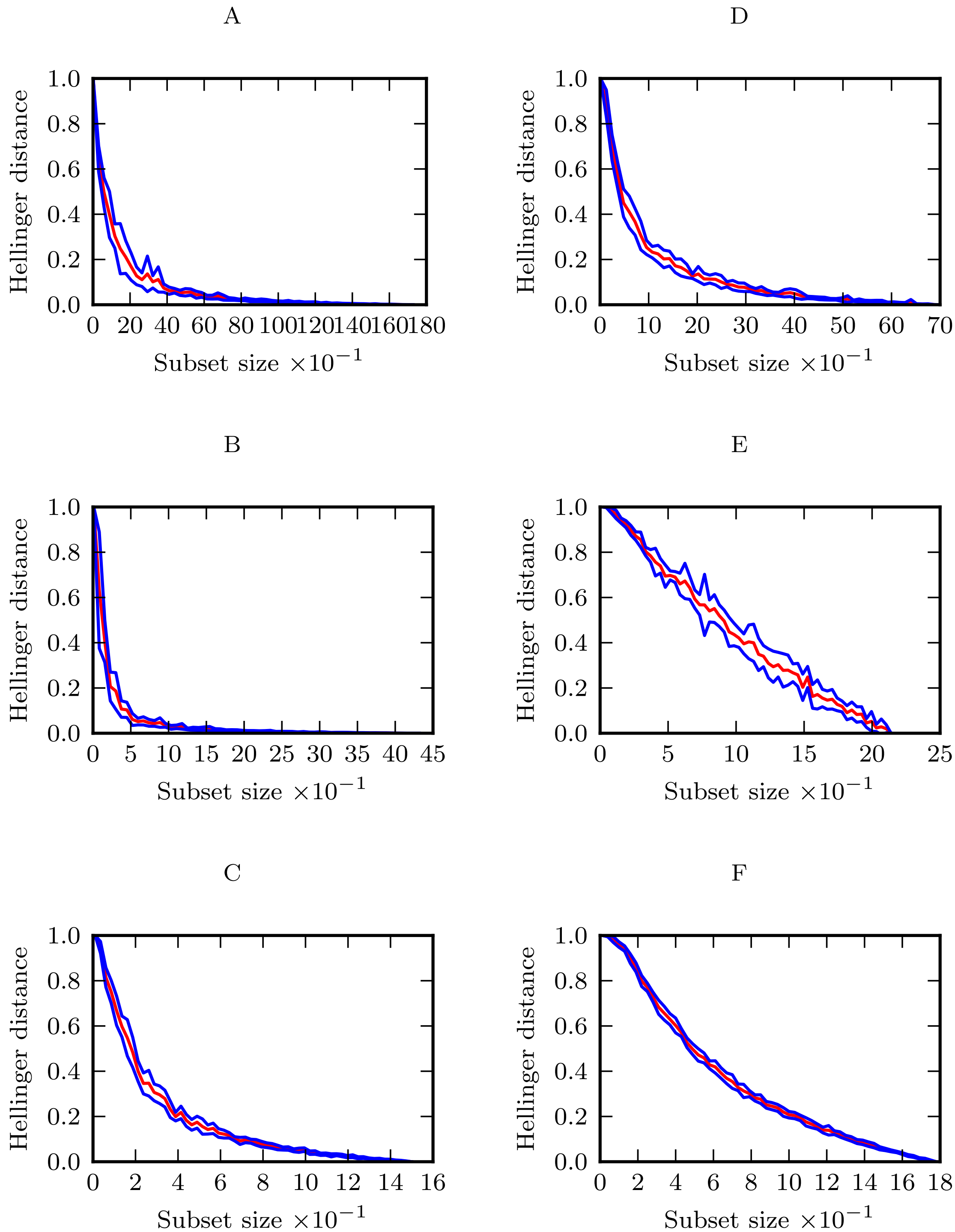

Figure 11. Conditional complexity curves for six different data sets from UCI Machine Learning repository with areas under complexity curve (AUCC) reported: A – CAR, AUCC: 0.08, B – MONKS-1, AUCC: 0.05, C – IRIS, AUCC: 0.19, D – BREAST-CANCER-WISCONSIN, AUCC: 0.13, E – GLASS, AUCC: 0.44, F – WINE, AUCC: 0.35.

	Instances	Attributes	Classes
UCI LED	100000	7	9
UCI WAVEFORM	100000	21	3
UCI ADULT	32561	14	2

Table 9. Basic properties of the data pruning benchmark data sets.

To verify if this idea is of practical use, we performed an experiment with three bigger data sets from UCI repository. Their basic properties are given by Table 9.

For all data sets we performed a stratified 10 fold cross validation experiment. The training part of a split was pruned according to our criterion function with $t = 0.005$ (CC pruning) or using geometric progressive sampling with multiplier $a = 2$ and initial sample size $n_0 = 100$ (PS pruning). Achieving the same accuracy as with CC pruning was used as a stop criterion for progressive sampling. Classifiers were trained on pruned and unpruned data and evaluated on the testing part of each cross validation split. Standard error was calculated for the obtained values. We have used machine learning algorithms from scikit-learn library (Pedregosa et al., 2011) and the rest of the procedure was implemented in Python with the help of NumPy and SciPy libraries. Calculations were done on a workstation with 8 core Intel® Core™ i7-4770 3.4 Ghz CPU working under Arch GNU/Linux.

Table 10 presents measured times and obtained accuracies. As can be seen, the difference in classification accuracies between pruned and unpruned training data is negligible. CC compression rate differs for the three data sets, which suggests that they are of different complexity: for LED data only 5% is needed to perform successful classification, while ADULT data is pruned at 33%. CC compression rate is rather stable with only small standard deviation, but PS compression rate is characterised with huge variance. In this regard, complexity curve pruning is preferable as a more stable pruning criterion.

In all cases when training a classifier on the unpruned data took more than 10 seconds, we observed huge speed-ups. With the exception of SVC on LED data set, complexity curve pruning performed better than progressive sampling in such cases. Unsurprisingly, real speed-ups were visible only for computationally intensive methods such as Support Vector Machines, Random Forest and Gradient Boosted Decision Trees. For simple methods such as Naïve Bayes, Decision Tree or Logistic Regression fitting the model on the unpruned data is often faster than applying pruning strategy.

These results present complexity curve pruning as a reasonable model-free alternative to progressive sampling. It is more stable and often less demanding computationally. It does not require additional convergence detection strategy, which is always an important consideration when applying progressive sampling in practice. What is more, complexity curve pruning can also be easily applied in the context of online learning, when the data is being collected on the fly. After appending a batch of new examples to the data set, Hellinger distance between the old data set and the extended one can be calculated. If the distance is smaller than the chosen threshold, the process of data collection can be stopped.

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 $C = 1$,

Classifier	Score	CC score	Time	CC time	PS time	PS rate
waveform						
Mean CC compression rate: 0.19 ± 0.02			Mean CC compression time: 4.01 ± 0.14			
LinearSVC	0.86 ± 0.00	0.86 ± 0.00	27.71 ± 0.35	$\mathbf{6.69 \pm 0.52}$	10.73 ± 8.65	0.55 ± 0.49
GaussianNB	0.80 ± 0.01	0.80 ± 0.01	$\mathbf{0.02 \pm 0.00}$	4.02 ± 0.14	0.03 ± 0.01	0.01 ± 0.00
RF	0.86 ± 0.00	0.85 ± 0.00	33.49 ± 0.04	$\mathbf{9.29 \pm 0.76}$	18.06 ± 10.75	0.46 ± 0.37
SVC	0.86 ± 0.00	0.86 ± 0.00	211.98 ± 0.93	$\mathbf{9.08 \pm 1.21}$	21.22 ± 28.34	0.33 ± 0.42
Tree	0.78 ± 0.00	0.77 ± 0.00	3.06 ± 0.06	4.50 ± 0.20	$\mathbf{1.40 \pm 0.70}$	0.37 ± 0.28
Logit	0.86 ± 0.00	0.86 ± 0.00	1.75 ± 0.06	4.21 ± 0.17	$\mathbf{0.60 \pm 0.62}$	0.30 ± 0.34
GBC	0.86 ± 0.00	0.86 ± 0.00	112.34 ± 0.12	$\mathbf{24.59 \pm 2.30}$	66.66 ± 37.99	0.53 ± 0.43
led						
Mean CC compression rate: 0.04 ± 0.01			Mean CC compression time: 1.38 ± 0.03			
LinearSVC	0.74 ± 0.00	0.74 ± 0.00	4.68 ± 0.10	1.49 ± 0.04	$\mathbf{0.47 \pm 1.04}$	0.13 ± 0.34
GaussianNB	0.74 ± 0.00	0.73 ± 0.00	$\mathbf{0.02 \pm 0.00}$	1.38 ± 0.03	0.07 ± 0.02	0.26 ± 0.44
RF	0.74 ± 0.00	0.73 ± 0.00	1.77 ± 0.01	1.47 ± 0.03	$\mathbf{0.83 \pm 0.25}$	0.05 ± 0.04
SVC	0.74 ± 0.00	0.74 ± 0.00	82.16 ± 0.86	$\mathbf{1.56 \pm 0.07}$	10.04 ± 17.52	0.26 ± 0.44
Tree	0.74 ± 0.00	0.73 ± 0.00	$\mathbf{0.03 \pm 0.00}$	1.38 ± 0.03	0.04 ± 0.01	0.09 ± 0.10
Logit	0.74 ± 0.00	0.74 ± 0.00	2.03 ± 0.08	1.42 ± 0.03	$\mathbf{0.30 \pm 0.44}$	0.17 ± 0.33
GBC	0.74 ± 0.00	0.73 ± 0.00	51.26 ± 0.40	$\mathbf{3.57 \pm 0.30}$	6.32 ± 4.05	0.04 ± 0.04
adult						
Mean CC compression rate: 0.33 ± 0.02			Mean CC compression time: 0.93 ± 0.05			
LinearSVC	0.69 ± 0.19	0.67 ± 0.20	1.79 ± 0.08	1.53 ± 0.08	$\mathbf{0.30 \pm 0.84}$	0.18 ± 0.52
GaussianNB	0.81 ± 0.01	0.81 ± 0.01	0.01 ± 0.00	0.93 ± 0.05	$\mathbf{0.01 \pm 0.00}$	0.02 ± 0.02
RF	0.86 ± 0.01	0.85 ± 0.01	2.04 ± 0.01	$\mathbf{1.60 \pm 0.09}$	2.11 ± 1.18	0.69 ± 0.59
SVC	0.76 ± 0.00	0.76 ± 0.00	81.70 ± 0.56	10.52 ± 2.31	$\mathbf{5.06 \pm 7.17}$	0.16 ± 0.19
Tree	0.81 ± 0.00	0.81 ± 0.01	0.12 ± 0.00	0.97 ± 0.05	$\mathbf{0.10 \pm 0.08}$	0.72 ± 0.72
Logit	0.80 ± 0.00	0.80 ± 0.00	0.08 ± 0.01	0.96 ± 0.05	$\mathbf{0.05 \pm 0.07}$	0.42 ± 0.68
GBC	0.86 ± 0.00	0.86 ± 0.00	2.33 ± 0.01	$\mathbf{1.80 \pm 0.09}$	2.37 ± 1.22	0.67 ± 0.57

Table 10. Obtained accuracies and training times of different classification algorithms on unpruned and pruned data sets. Score corresponds to classifier accuracy, time to classifier training time (including pruning procedure), rate to compression rate. CC corresponds to data pruning with complexity curves, PS to data pruning with progressive sampling. LinearSVC – linear support vector machine, GaussianNB – naïve Bayes with gaussian kernel, RF – random forest 100 CART trees, SVC – support vector machine with radial basis function kernel, Tree – CART decision tree, Logit – logistic regression, GBC – gradient boosting classifier with 100 CART trees.

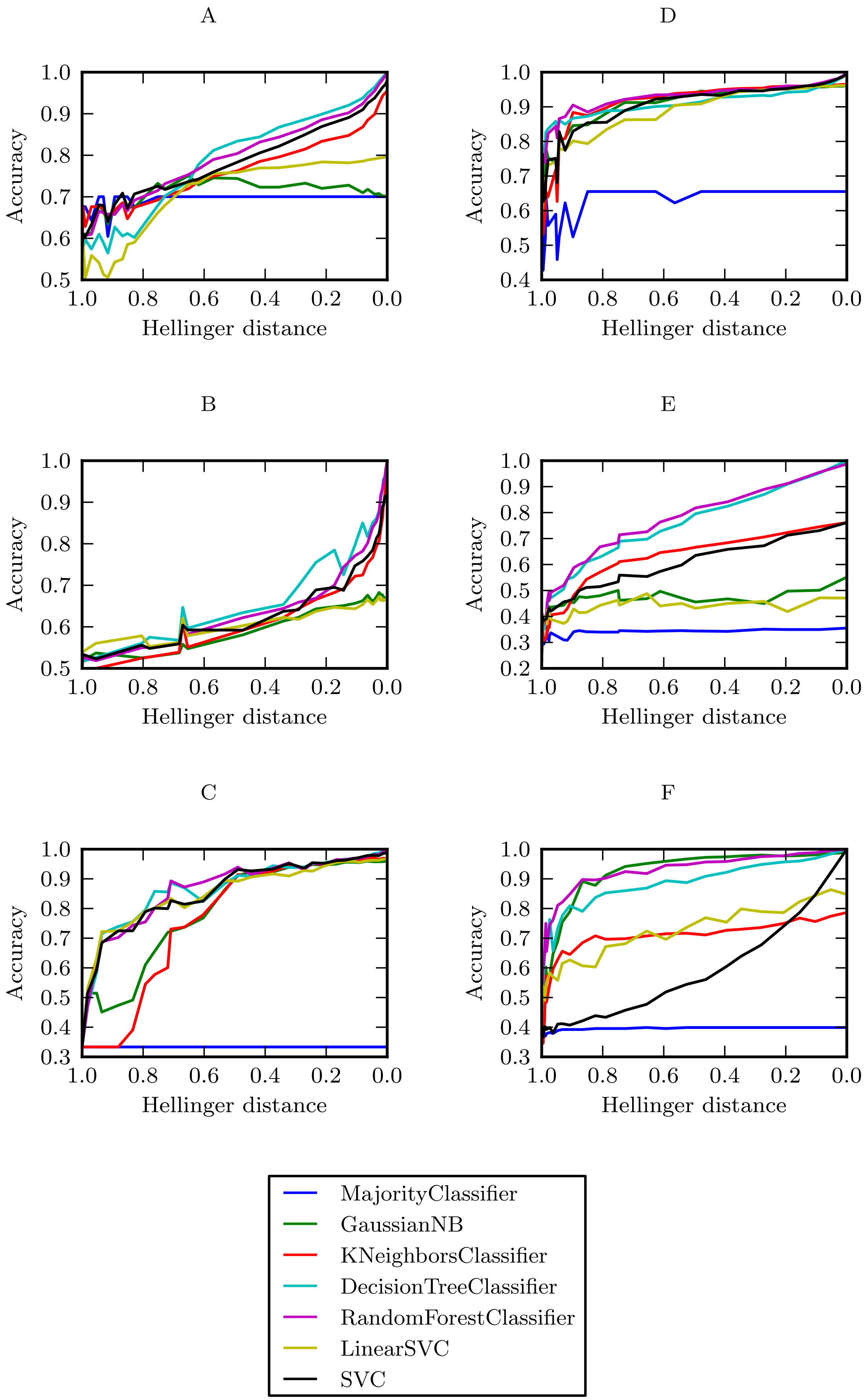

Figure 12. Generalisation curves for various data sets and classification algorithms. A – CAR, B – MONKS-1, C – IRIS, D – BREAST-CANCER-WISCONSIN, E – GLASS, F – WINE.

- 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 12.

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 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 11. 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

	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 11. 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.

simpler model is adequate. It takes into account algorithm properties ignored by standard performance metrics.

CONCLUSIONS

In this article we introduced a measure of data complexity targeted specifically at data sparsity. This distinguish it from other measures focusing mostly on the shape of optimal decision boundary in classification problems. The introduced measure has a form of graphical plot – complexity curve. We showed that it exhibits desirable properties through a series of experiments on both artificially constructed and real-world data sets. We proved that complexity curve capture non-trivial characteristics of the data sets and is useful for explaining the performance of high-variance classifiers. With conditional complexity curve it was possible to perform a meaningful analysis even with heavily imbalanced data sets.

Then we demonstrated how complexity curve can be used in practice for data pruning (reducing the size of training set) and that it is a feasible alternative to progressive sampling technique. This result is immediately applicable to all the situations when data overabundance starts to pose a problem. For instance, it is possible to perform a quick exploration study on a pruned data set before fitting computationally expensive models on the whole set. Pruning result may also provide a suggestion for choosing proper train-test split ratio or number of folds of cross-validation in the evaluation procedure.

Knowing data sparseness is useful both for evaluating the trained classifiers and classification algorithms in general. Using the concept of the complexity curve, we developed a new performance measure – an extension of learning curve called generalisation curve. This method presents classifier generalisation capabilities in a way that depends on the data set information content rather than its size. It provided more insights into classification algorithm dynamics than commonly used approaches.

We argue that new performance metrics, such as generalisation curves, are needed to move away from a relatively static view of classification task to a more dynamic one. It is worth to investigate how various algorithms are affected by certain data manipulations, for example when new data become available or the underlying distribution shifts. This would facilitate the development of more adaptive and universal algorithms capable of working in a dynamically changing environment.

Experiments showed that in the presence of large number of redundant attributes not contributing to the classification task complexity curve does not correlate well with classifier performance. It correctly identifies dimensional sparseness of the data, but that is misleading since the actual decision boundary may still be very simple. Because of this as the next step in our research we plan to apply similar probabilistic approach to measure information content of different attributes in a data set and use that knowledge for performing feature selection. Graphs analogical to complexity curves and generalisation curves would be valuable tools for understanding characteristics of data sets and classification algorithms related to attribute structure.

Our long-term goal is to gain a better understanding of the impact of data set structure, both in terms of contained examples and attributes, and use that knowledge to build heterogeneous classification ensembles. We hope that a better control over data sets used in experiments will allow to perform a more systematic study of classifier diversity and consensus methods.

REFERENCES

Alcalá, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., and Herrera, F. (2010). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(255-287):11.

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Hudson, J., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O., and Staudt, L. M. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750.

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*.

Brent, R. P. (1973). *Algorithms for Minimization Without Derivatives*. Dover Publications.

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.

Chvátal, V. (1979). The tail of the hypergeometric distribution. *Discrete Mathematics*, 25(3):285–287.

Cieslak, D. A., Hoens, T. R., Chawla, N. V., and Kegelmeyer, W. P. (2011). Hellinger distance decision trees are robust and skew-insensitive. *Data Mining and Knowledge Discovery*, 24(1):136–158.

Domingos, P. (2000). A Unified Bias-Variance Decomposition for Zero-One and Squared Loss. In *AAAI/IAAI*, pages 564–569. AAAI Press.

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.

Díez-Pastor, J. F., Rodríguez, J. J., García-Osorio, C. I., and Kuncheva, L. I. (2015). Diversity techniques improve the performance of the best imbalance learning ensembles. *Information Sciences*, 325:98–117.

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

Frank, A. and Asuncion, A. (2010). *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences.

Golub, T. R. (1999). Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286(5439):531–537.

Ho, T. K. (2008). Data Complexity Analysis: Linkage between Context and Solution in Classification. In Lobo, N. d. V., Kasparis, T., Roli, F., Kwok, J. T., Georgiopoulos, M., Anagnostopoulos, G. C., and Loog, M., editors, *Structural, Syntactic, and Statistical Pattern Recognition*, number 5342 in Lecture Notes in Computer Science, pages 1–1. Springer Berlin Heidelberg.

Ho, T. K. and Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300.

Ho, T. K., Basu, M., and Law, M. H. C. (2006). Measures of geometrical complexity in classification problems. In *Data complexity in pattern recognition*, pages 1–23. Springer.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks: the official journal of the International Neural Network Society*, 13(4-5):411–430.

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.

Johnstone, I. M. and Titterington, D. M. (2009). Statistical challenges of high-dimensional data. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 367(1906):4237–4253.

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.

Li, L. and Abu-Mostafa, Y. S. (2006). Data complexity in machine learning. Technical report, California Institute of Technology.

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

Luengo, J. and Herrera, F. (2013). An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. *Knowledge and Information Systems*, 42(1):147–180.

Macià, N., Bernadó-Mansilla, E., Orriols-Puig, A., and Kam Ho, T. (2013). Learner excellence biased by data set selection: A case for data characterisation and artificial data sets. *Pattern Recognition*, 46(3):1054–1066.

Mantovani, R. G., Rossi, A. L. D., Vanschoren, J., Bischl, B., and Carvalho, A. C. P. L. F. (2015). To tune or not to tune: Recommending when to adjust SVM hyper-parameters via meta-learning. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

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

Orriols-Puig, A., Macià, N., and Ho, T. K. (2010). DCoL: Data Complexity Library in C++ (Documentation). *ResearchGate*.

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.

Provost, F., Jensen, D., and Oates, T. (1999). Efficient Progressive Sampling. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 23–32, New York, NY, USA. ACM.

Ramaswamy, S., Ross, K. N., Lander, E. S., and Golub, T. R. (2003). A molecular signature of metastasis in primary solid tumors. *Nature Genetics*, 33(1):49–54.

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.

Scott, D. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley Series in Probability and Statistics. Wiley.

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

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., Lander, E. S., Loda, M., Kantoff, P. W., Golub, T. R., and Sellers, W. R. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1(2):203–209.

Skala, M. (2013). Hypergeometric tail inequalities: ending the insanity. *arXiv preprint arXiv:1311.5939*.

Smith, M. R., Martinez, T., and Giraud-Carrier, C. (2013). An instance level analysis of data complexity. *Machine Learning*, 95(2):225–256.

Smith-Miles, K., Baatar, D., Wreford, B., and Lewis, R. (2014). Towards objective measures of algorithm performance across instance space. *Computers & Operations Research*, 45:12–24.

Smith-Miles, K. and Lopes, L. (2012). Measuring instance difficulty for combinatorial optimization problems. *Computers & Operations Research*, 39(5):875–889.

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.

van 't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536.

Yin, L., Ge, Y., Xiao, K., Wang, X., and Quan, X. (2013). Feature selection for high-dimensional imbalanced data. *Neurocomputing*, 105:3–11.