ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 237-242

Error Rate Estimation Via Cross-Validation and Learning Curve Theory

Varfis A. & Corleto L.

CEC - Joint Research Center, Ispra Establishment
Institute for System Engineering and Informatics (ISEI).
Neural Network Laboratory - TP 361, 21020 ISPRA (VA), Italy.

Abstract. Let $\mathcal{L}_n$ be the expected error rate for a neural network trained with n examples. We present three novel estimators for $\mathcal{L}_n$ that do not require additional examples. They operate in two steps. Firstly, $\mathcal{L}_{kn}$ is estimated by training the model on subsets of size kn ($0<k<1$) of the original training sample and measuring error rates on the remaining examples available. Secondly, adjustment from $\mathcal{L}_{kn}$ to $\mathcal{L}_n$ is provided by learning curve theory. For k=0.632, the proposed estimation process bears some resemblance to the "632" bootstrap estimator, acknowledged as one of the best performing. Comparison using a real-life classification task suggests the new estimators are effective and worth further investigation.

1. Genesis

1.1 Error rates

Let us consider a learning machine - e.g. a neural network - which would implement a mapping f between a *predictor vector* x and a *response vector* y. Let F be the distribution of $z = (x,y)$ predictor-response pairs. Via i.i.d. sampling from F, a training set of size n, $T_n = \{z_1,...,z_n\}$, is available. In order to indicate easily the integrand of expectation operators in the forthcoming equations, it will be convenient to use a different notation, say F_n ($=\otimes^n F$), for the T_n distribution. Observations will be related to their parent distribution with the ~ symbol, i.e. $z \sim F$ and $T_n \sim F_n$.
Based on the T_n training set, the learning machine implements the f_{T_n} mapping. This paper focuses on estimating the (average) prediction error on additional samples $z_0 = (x_0,y_0) \sim F$, as measured by some given *loss function* $l(z_0,T_n) = l(y_0, f_{T_n}(x_0))$ - e.g. the quadratic cost $(y_0 - f_{T_n}(x_0))^2$. Relevant quantities in this setting are the *actual* (or *conditional*) *error rate* $L(T_n)$ and the *expected* (*unconditional*) *error* $\mathcal{L}_n$:

$$L(T_n) = E_F\, l(z,T_n) \qquad (1)$$

$$\mathcal{L}_n = E_{F_n} E_F\, l(z,T_n) \qquad (2)$$

L refers to a specific training set, and expectation is taken over z only. $\mathcal{L}_n$ is the expected value of L for the learning machine at hand, given that n i.i.d. samples are used for training. Since practical users face a single T_n training set, they are mainly interested in the estimation of the conditional error rate $L(T_n)$ - often, quantities that might rather be deemed estimators for $\mathcal{L}_n$, are eventually utilised as estimators for L. A straight estimator for L is the *apparent error* $A(T_n)$:

$$A(T_n) = E_{\hat{F}}\, l(z,T_n) = \frac{1}{n}\sum_{i=1}^{n} l(z_i,T_n) \qquad (3)$$

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 237-242

$\hat{F}$ denotes the *empirical distribution* on T_n, which assigns a probability of $1/n$ to each of the observed z_i's. Since the same examples are used both to tune and to test the f_{T_n} mapping, $A(T_n)$ tends to underestimate the conditional error rate. Accordingly, the difference $Opt(T_n) = L(T_n) - A(T_n)$ is often referred to as *optimism*. Because actual and apparent error rates are correlated, more effective estimations of $L(T_n)$ are obtained via $\hat{O}pt\ (T_n)$ estimators of the optimism and thus have the following form:

$$\hat{L}(T_n) = A(T_n) + \hat{O}pt(T_n) \qquad (4)$$

1.2 The "632" bootstrap estimator

Many real-life estimation or optimisation methods - and connectionist learning is no exception - would rather use all n observed data for their computations. Even when dealing with observable statistics from the data (like $A(T_n)$), the problem of estimating a parameter from F_n (like $\mathcal{A}_n = E_{F_n}A(T_n)$, the expected apparent error) is cumbersome since only a single T_n sample is available from the F_n distribution. Bootstrap estimators do with the computer what the user would do in practice, if it were possible: he would repeat the experiment. The main idea of the bootstrap methodology assumes that every quantity of interest in the *real world* - corresponding to the true underlying distribution F - may be transposed (plugged-in) *mutando mutandis* in the *bootstrap world* which is driven by the $\hat{F}$ empirical distribution. Accordingly, observations $z \sim F$ become $z^* \sim \hat{F}$ and training sets $T_n \sim F_n$ become *bootstrap samples* $T_n^* \sim \hat{F}_n$. Bootstrap samples may be generated at will by sampling n times *with replacement* from $\hat{F}$. In connectionist jargon, training with a T_n^* bootstrap sample would correspond to training epochs with n presentations, but with some given examples of the parent T_n set appearing twice or thrice in each cycle, while other available examples are never recalled. The expected number of different z_i's in a given T_n^* bootstrap sample amounts to $1 - (1-\frac{1}{n})^n \approx 0.632n$. Hence the coinage of the "632" bootstrap estimator, which has something in common with and inspired the findings of this paper. The steps in the implementation of the "632" estimator are listed below:

1. Generate many bootstrap samples T_n^{*b}, $b=1,...,B$ and train the learning machine with each of these.
2. Apply the bootstrap transposition principle to out-of-training-sample error rates: for each b this yields a *bootstrap replication* $\sum_{i=1}^{n} I^b(z_i) l(z_i, T_n^{*b}) / \sum_{i=1}^{n} I^b(z_i) = Out^{*b}(T_n^{*b})$. Here I^b stands for the indicator function of $T_n - T_n^{*b}$.
3. Compute the mean of these bootstrap replications, weighted by out-of-bootstrap-sample occurrences: $\overline{Out}_B\ (T_n) = \sum_{b=1}^{B} \sum_{i=1}^{n} I^b(z_i) l(z_i, T_n^{*b}) / \sum_{b=1}^{B} \sum_{i=1}^{n} I^b(z_i)$.
4. The "632" estimator, which corresponds in (4) to $\hat{O}pt\ (T_n) = 0.632\ [\overline{Out}_B(T_n) - A(T_n)]$, is the following weighted mean between $A(T_n)$ and $\overline{Out}_B\ (T_n)$:

$$\hat{L}_{632}\ (T_n) = 0.368A(T_n) + 0.632\,\overline{Out}_B\ (T_n) \qquad (5)$$

$\hat{L}_{632}$ comes from a loose *distance* argument, whereby test samples in the real world ($z_0 \notin T_n$) are closer to their training set than test samples in the bootstrap world ($I^b(z_0) = 1$), by a factor of 0.632. After some computation, one obtains the following two approximations [1,2]:

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 237-242

$$\mathcal{L}_n \approx \mathcal{A}_n + \frac{\gamma}{n} \qquad (6)$$

$$\text{Out}^* \approx \mathcal{A}_n + \frac{\gamma}{0.632n} \qquad (7)$$

$\mathcal{A}_n = E_{F_n} A(T_n)$ is replaced by its $A(T_n)$ realisation. Out* is the expected value of out-of-bootstrap-sample errors and is estimated by $\overline{\text{Out}}_B\ (T_n)$. Formula (5) is finally obtained by eliminating γ between (6) and (7).
As from [1], the $\overline{\text{Out}}_B\ (T_n)$ bootstrap estimator can be thought of as a smoothed version of the leave-one-out cross-validation estimator, thereby trading near unbiasedness for reduced variability. Indeed, the expectation of $\overline{\text{Out}}_B\ (T_n)$ (i.e. Out*) is about $\mathcal{L}_{n/2}$ [2], which tends to be larger than the $\mathcal{L}_{n-1} \approx \mathcal{L}_n$ expectation of the leave-one-out cross-validation estimator. Regarding the upward bias magnitude, note that having an average fraction of $0.632n$ different exemplars out of n training samples is worse than using a training set limited to the $0.632n$ different elements, as the latter case would correspond to $\mathcal{L}_{0.632n}$ as expected value.

2. Novel estimators

2.1. Learning curves

Considering many theorems which have been established in a wide variety of different settings, behaviour in $1/n$ as in equations (6) and (7) appears to be a sort of "universal" property of learning curves [3]. Both the expected generalisation and apparent error are involved, yielding the following pair of companion formulae:

$$\mathcal{L}_n \approx \alpha + \frac{\beta}{n} \qquad (8)$$

$$\mathcal{A}_n \approx \alpha - \frac{\beta}{n} \qquad (9)$$

α and β depend on the model at hand, and $\alpha = \mathcal{A}_\infty = \mathcal{L}_\infty$ may be interpreted as the asymptotic error rate. For connectionist models, a relevant reference is [4] where the equations above are derived in a setting suitable for multilayer perceptron (MLP) networks: (i) unfaithful models are allowed, i.e. it is possible that no set of network parameters exists which maps the conditional distribution of y given x; (ii) general smooth loss functions are minimised via (stochastic) gradient descent algorithms.
Regarding the standard implementation of MLP classifiers - one output neuron per category and class indicator target vectors, formulae (8) and (9) apply to the expected MSE between binary target values and MLP outputs. Since explicit minimisation of misclassification rate is not embodied in the objective loss function, *the learning curve formulae are not established for the probability of classification error.* Nevertheless, extending the (8,9) twin $1/n$ behaviour - with different (α,β) coefficients - from a MSE with respect to class indicator targets to a misclassification rate loss function is a reasonable assumption we shall make as new tentative estimators of $\mathcal{L}_n$ are investigated for classification tasks.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 237-242

2.2. Three tentative estimators

A customary use of (8) and (9) is model selection within a series of nested models [4], where these equations are first subtracted to yield the expected optimism $O_n = \mathcal{L}_n - \mathcal{A}_n \approx 2\beta/n$ and then β is estimated from theoretic arguments. Our approach is quite different: first, we estimate α and/or β in the "bootstrap-like" world via replications with kn different training samples ($0<k<1$) and use of the corresponding learning curve equations ($\mathcal{L}_{kn} \approx \alpha + \beta/kn$; $\mathcal{A}_{kn} \approx \alpha - \beta/kn$); second, we plug the estimated parameters into the "real" world equation (8), possibly replacing $\mathcal{A}_n$ by its $A(T_n)$ realisation. This yields the following three tentative estimators, which result from estimation and substitution of α alone, β alone or both α and β, respectively:

$$\hat{L}_{\alpha}(T_n) = \mathcal{L}_{kn} + \mathcal{A}_{kn} - A(T_n) \qquad (10)$$

$$\hat{L}_{\beta}(T_n) = A(T_n) + k\, O_{kn} = A(T_n) + k\,(\mathcal{L}_{kn} - \mathcal{A}_{kn}) \qquad (11)$$

$$\hat{L}_{\alpha\beta}(T_n) = \tfrac{1}{2}\left[(1-k)\mathcal{A}_{kn} + (1+k)\mathcal{L}_{kn}\right] \qquad (12)$$

In each case $\mathcal{L}_{kn}$ and $\mathcal{A}_{kn}$ are estimated as follows:

1. *B partitions* of the T_n into a T_{kn}^b learning set and a $T_n - T_{kn}^b = Out^b$ test set are designed.
2. For each b, a neural network is trained on T_{kn}^b and tested on Out^b.
3. The grand averages of the learning and test set losses are used as estimators of $\mathcal{A}_{kn}$ and $\mathcal{L}_{kn}$, respectively.

In the case of $k = 0.632$, the difference with the $\overline{A}_B^* = 1/B\ \Sigma_b A(T_n^{*b})$ or $\overline{Out}_B(T_n)$ bootstrap estimators of $\mathcal{A}_{kn}$ or $\mathcal{L}_{kn}$ is twofold: (i) use of *exactly kn* different training examples is enforced in each T_{kn}^b set - and not *kn on average* as for T_n^{*b} sets; (ii) elements from T_n never occur more than once in any T_{kn}^b set. The former distinction is negligible, but the latter entails perceptible differences in the estimations, as e.g. the discrepancy between $\mathcal{L}_{0.632n}$ and $\mathcal{L}_{n/2}$ pinpointed previously.

There is no point in picking the B partitions randomly. We took T_{kn}^b to be the subset of T_n corresponding to the kn consecutive indices, modulo n, starting from (the integer part of) bn/B. A balanced experimental design of that kind ensures that each z_i is used about kB times for training. The partitions design and the estimation of $\mathcal{L}_{kn}$ resembles B-fold cross-validation, but with overlapping test sets and smaller training sets (if $(1 - k) > 1/B$).

3. Experimental Results

3.1. Data

The case studies reported here deal with MLP models which classify multidate multispectral satellite imagery into a large number of landcover classes. The input data have been extracted from 3-channel satellite images from 2 different dates. Since no context information is used, the input vector to the MLP classifier naturally consists of 6 features, i.e. the digital count values of each channel for a single pixel from the 2-date imagery. Ground truth data D was acquired by field survey. The D dataset comprised about 3800 labelled pixels representing 20 ground cover classes like

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 237-242

"vineyard", "tomatoes, "water" etc...The x_i training examples therefore belong to $\Re^6 \times \{0,1\}^{20}$ and MLP networks will have 6 input units and 20 output units.
Comparing resampling estimators is computationally demanding - more than 5000 learning sessions have been carried out in this study. To speed up the learning phase, the architecture has been restricted to a single hidden layer of small size (6 units). It is worth stating again that this paper focuses on assessing bootstrap and other resampling estimators of true error rates, so that model optimality is hardly a matter of concern. Indeed, substantial improvement is achieved with larger MLP models [5].
Classification tasks on satellite imagery offer a good testbed for assessing the *operational* capabilities of resampling techniques applied to connectionist models. The main reason is that field surveys are rather resource-consuming: collecting a dataset which is sufficiently large to represent the main features of the complex actual input-output mapping corresponds typically to several man-weeks. Therefore, computer intensive methods may easily become attractive if they spare additional test or validation datasets.

3.2. Mean Squared Errors

Some faithful measure of the true error rate must be available if various estimators - namely $\hat{L}$ = (5), (11), (10) and (12) - are to be assessed or compared. The "available" training set size was set to n = 1000 in our experiments, so that with 2800 test examples remaining the standard error of the empirical $L(T_n)$ mean is negligible. Note that the 3800 labelled examples at hand are viewed as a sample from a much larger population, not as a *census* of F; accordingly, T_{1000} training sets are generated by sampling without replacement from D, and their actual error rates $L(T_{1000})$ are best estimated from out-of-training-sample data.
The definition as well as the statistical properties of the $\hat{L}$ estimators depend on the number of replications involved; B was set to 50 throughout, which is large enough to ensure that the "resampling" variability of $\hat{L}$ is low as against its *sampling* variability. Resampling refers here to the various ways of picking 50 bootstrap samples for (5) or for designing balanced partitions for (10-12). Only the sampling variability of T_n under F_n was taken into account as the MSE of the various $\hat{L}(T_n)$ estimators were computed. Depending on whether the target quantity is $L(T_n)$ or $\mathcal{L}_n$, these measures are referred to as *Average Conditional Error (ACE)* or *Unconditional MSE (UMSE)*, respectively. Each MSE may be split into a bias and a variance term:

$$ACE(\hat{L}) = E_{f_n}[\hat{L} - L]^2 = [E_{F_n}\hat{L} - \mathcal{L}_n]^2 + Var_{F_n}[\hat{L} - L] \quad (13)$$

$$UMSE(\hat{L}) = E_{f_n}[\hat{L} - \mathcal{L}_n]^2 = [E_{F_n}\hat{L} - \mathcal{L}_n]^2 + Var_{F_n}\hat{L} \quad (14)$$

These quantities have been estimated by computing $\hat{L}(T_{1000})$ and $L(T_{1000})$ for 20 different T_{1000} subsets of D. This involves the assumption that a 3800/1000 ratio is large enough to ensure that extracting sets from D may substitute for sampling from the true F, as far as empirical MSE estimates are concerned. ACE (respectively UMSE) estimates for $\hat{L}_\alpha$, $\hat{L}_\beta$, $\hat{L}_{\alpha\beta}$ and $\hat{L}_{632}$ are displayed in bold (respectively italics) in Table 1, for three values of k. Next follows the standard error of the mean which is used to compute these estimates. With only 20 squared differences available, the figures are relatively high, so some care should be taken as the MSE are contemplated; yet carrying out enough experiments to reduce substantially the standard errors would have been quite expensive in CPU time. The third column shows the negligible squared bias from faithful estimates of $\mathcal{L}_n$ obtained via 200 replicates.

ESANN'1996 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-25-26 April 1996, D-Facto public., ISBN 2-9600049-6-5, pp. 237-242

	ACE *UMSE*								
	kn = **500**			*kn* = **632**			*kn* = **870**		
$\hat{L}_{\alpha}$	**3.09**	0.76		**3.78**	0.97		**3.61**	0.80	
	1.54	0.34	0.00	*2.11*	0.54	0.03	*2.06*	0.51	0.03
$\hat{L}_{\beta}$	**2.87**	0.81		**2.80**	0.80		**2.82**	0.82	
	2.00	0.65	0.02	*2.00*	0.67	0.00	*2.24*	0.69	0.04
$\hat{L}_{\alpha\beta}$	**2.47**	0.60		**2.81**	0.73		**2.75**	0.68	
	1.26	0.33	0.01	*1.58*	0.43	0.01	*1.68*	0.46	0.00
$\hat{L}_{632}$				**2.68**	0.75				
				1.68	0.55	0.13			

TABLE 1

4. Conclusion

Although the arguments which lead to formula (5) may admittedly be deemed to lack firm theoretical foundations - as those which lead to (10-12), the practical power of the "632" bootstrap estimator has been demonstrated across several benchmarking exercises [1,2]. Indeed, it outperformed on the same data both the straight bootstrap estimator of Opt ($\hat{\omega}^{(BOOT)}$ in [2]) and the 50-fold cross-validation estimator (980 training / 20 test samples). However, the $\hat{L}_{\beta}$ and even more the $\hat{L}_{\alpha\beta}$ estimators are appealing contenders which are nearly unbiased, yet with a variance of the same order as for $\hat{L}_{632}$. The practicality of these two estimators is worth further investigation.

References

[1] B. Efron, R. Tibshirani. Cross-Validation and the Bootstrap. Estimating the Error Rate of a Prediction Rule. In ftp-stite *ustat.toronto.edu* in *pub/bootpred.shar*.

[2] B. Efron. Estimating the Error Rate of a Prediction Rule: Improvement on Cross-Validation. *JASA* vol. 78, pp. 316-331, 1983.

[3] S-I. Amari. Mathematical methods of neurocomputing. In *Networks and Chaos - Statistical and Probabilistic Aspects*, Chapman & Hall, pp. 1-36, 1993.

[4] N. Murata, S. Yoshizawa, S-I. Amari. Network Information Criterion - Determining the Number of Hidden Units for an Artificial Neural Network Model. *IEEE Trans. On Neural Networks* vol. 5, pp. 865-872, 1994.

[5] I. Kanellopoulos, A. Varfis, G.G. Wilkinson, J. Mégier. Land-cover discrimination in SPOT HRV imagery using an artificial neural network - a 20-class experiment. *Int. J. Remote Sensing* vol. 13, pp. 917-924, 1992.