ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

Adaptive Optimization for Cross Validation

Alessandro Rudi[1] and Gabriele Chiusano[2] and Alessandro Verri[2]

1- Robotics, Brain and Cognitive Sciences Department - Istituto Italiano di Tecnologia
Via Morego 30, 16163, Genova - Italy

2- Dipartimento di Informatica e Scienza dell'Informazione - University of Genoa
Via Dodecaneso 35, 16146, Genova - Italy

Abstract. The process of model selection and assessment aims at finding a subset of parameters that minimize the expected test error for a model related to a learning algorithm. Given a subset of tuning parameters, an exhaustive grid search is typically performed. In this paper an automatic algorithm for model selection and assessment is proposed. It adaptively learns the error function in the parameters space, making use of the Scale Space theory and the Statistical Learning theory in order to estimate a reduced number of models and, at the same time, to make them more reliable. Extensive experiments are performed on the MNIST dataset.

1 Introduction

According to the No Free Lunch theorem [13], any learning algorithm is based on assumptions which affect the learning process as a strong bias while making it possible. Commonly these assumptions are modulated by meta-parameters whose correct tuning had been proved to be extremely important in practice for an effective assessment on the generalization performances [7]. The goal of model selection (MS) is to find the suitable meta-parameters of a learning algorithm on a given problem, in order to minimize the classification error over independent set of data. Cross Validation (CV) is the family of most used statistically efficient data driven MS approaches [3]. Each CV method express such statistical MS problem as the optimization one of finding a minimum over data-dependent function which is commonly noisy and computationally expensive to compute. Moreover an important requirement of the found minimum is stability: the performance of the learning algorithm should be stable w.r.t. a small meta-parameter variation. Despite different optimization methods for CV functions have been proposed in literature [1, 11], one of the most used approach to minimize the CV error function is still the Grid Search (GS). It consists in sampling the error function over a given grid of tuning parameters then selecting the point associated to the minimum error value. Up to our knowledge, it seems that no algorithms (nor the GS) satisfy the requirement of stability.

In the context of statistical learning, two different lines of research has been explored: one is coming from the statistical community, which have an extensive literature of studies related to the efficiency and the convergence of CV methods (see the work from Arlot et al. [3] for a complete review) and one is coming from the machine learning community, in which a number of works that tackle the problem of model selection optimization focusing on a specific learning algorithms has been proposed, in particular for Support Vector Machines (SVM) and Kernel Regularized Least Squares (KRLS). The works from Cherkassky [5] and Adankov et al. [1] propose two different methods for the selection of the parameter C which controls the slack variables in the SVM algorithm, while Chapelle et al. [4] apply the gradient descent method to an upper bound of the classification error function in order to find a suitable kernel parameters for C-SVM. An et al. [2] and Pahikkala et al. [11] proposed two different methods to speed up the computation of the CV error function for estimating the regularization parameter of the KRLS algorithm. The problem of model selection using CV on learning algorithms is closely related to the global optimization of error functions, sampled from few points and computationally intensive to evaluate in each point. In this setting, Osborne et al. [10] propose to learn computationally expensive functions from few sampled points and contemporarily to search its minima using Gaussian Processes. Up to our knowledge there are no algorithms that optimize the CV error using an automatic and adaptive learning process, contemporarily aiming at the stability requirements of the solution and exploiting the low dimensionality of the problem in order to sample the error function using less points as possible.

In this paper a refined model selection procedure is presented: it employs Scale Space and Statistical Learning Theories in order to exploit the stability condition of the searched minima and the low dimensionality of the CV-function requiring few evaluations of such function. The *Adaptive Optimization for Cross Validation* (ACV) learns the error function adaptively by sampling and refining the approximation only on the regions of stable minima without demanding strong computational efforts in parameters choice, compared to standard GS approach. The ACV performances are compared to the GS approach on classification problems using Support Vector Machines (SVM).

2 The proposed method

The presented paper is organized as follows: the main contribution of this paper is described in sections 2.1 and 2.2. Then the implementation details and the obtained results are reported in section 3.

2.1 Adaptive Optimization for Cross Validation

Let $A(\lambda)$ be a supervised learning algorithm which depends on the meta-parameter $\lambda \in \mathbb{R}^d$ and let $f(\lambda) = \mathcal{E}(X_n, A(\lambda))$ be the error function of a cross validation scheme computed using $A(\lambda)$ with training set X_n. The goal of an efficient model selection method is to find a suitable minimum plateau $f(\lambda)$ of volume σ^d using a small number of evaluations of f, where σ is the characteristic dimension. The ACV algorithm adaptively learns an approximation of $f(\lambda)$, in order to describe the global properties of the error function sampled over few points and then refining it progressively the approximation in the neighborhood of the minimum plateau. Given a characteristic dimension σ, a maximum number of points $\hat{n}$ per level of refinement, a subset L of the parameter space Λ and a maximum

ESANN 2012 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2012, i6doc.com publ., ISBN 978-2-87419-049-0. Available from http://www.i6doc.com/en/livre/?GCOI=28001100967420.

number of refinement steps N, the algorithm works as show in the Algorithm 1 box.

Algorithm 1 Adaptive Optimization for Cross Validation

```
$Z \leftarrow \emptyset$
while $N > 0$ and $\mathrm{vol}(L) \geq \sigma^d$ do
    $n \leftarrow \min\left(\hat{n}, \mathrm{vol}(L)\sigma^{-d}\right)$
    $\Lambda_n \leftarrow$ Sample$(L, n)$
    $Z \leftarrow Z \cup \{(\lambda, f(\lambda)) | \lambda \in \Lambda_n\}$
    $\phi \leftarrow$ SmoothKRLS$(Z, \sigma)$
    $m \leftarrow \int_L \phi(\lambda) d\lambda$
    $L \leftarrow \{\lambda \,|\, \phi(\lambda) \leq m\}$
    $N \leftarrow N - 1$
end while
return GlobalMinimum$(\phi, L)$
```

At each step the function f is sampled in a number n of points extracted from the region L and a function ϕ is learned over these sampled points. Then the mean m of ϕ in L is computed. In the end, the region L is updated restricting the sampling region only where $\phi(\lambda) \leq m$, following the simple heuristic that the global minimum of a function is always lower than its mean value. The algorithm stops if the volume of L is less than the reference volume σ^d or if the maximum number of steps N has been reached.

One of the main contributions of this paper is the study and the development of the algorithm to learn the error function, called *Smooth Kernel Regularized Least Squares* (*Smooth KRLS*), which combines the KRLS algorithm with the Linear Scale Space theory. This method is able to deal with the noise of the error function $f(\lambda)$, which is induced both by the finiteness of the training set X_n and by the sparse sampling of the $f(\lambda)$. The role of *Linear Scale-Space theory* as extensively proved by Lindeberg [8] is crucial to partially suppress the effect of the noise of the $f(\lambda)$ and to reduce the possibility of finding local or small volume minima in favor of bigger ones. To this extent, the linear Scale-Space approach suggests to learn the function $\phi(\lambda) = (\mathcal{N}_{\sigma^2} * f)(\lambda)$ over a set of samples $\{\lambda_i, f(\lambda_i)\}$, which is the original function f convoluted with a Gaussian of standard deviation σ, instead of the raw f, in order to reduce the effect of the noise-induced structures of characteristic dimension smaller than σ while substantially preserving the structures bigger than σ. To summarize, $\mathcal{N}_\sigma * f$ preserve plateaus of characteristic dimension greater or equal σ while reducing the sampling noise.

2.2 Smooth Kernel Regularized Least Squares

In this section a variation of Kernel Regularized Least Squares Regression, based on mathematical framework in [12], is presented. Given a noisy function f sampled in n points, the novel approach of *Smooth KRLS* learns the simpler function $\phi = \mathcal{N}_{\sigma^2} * f$ that is f convoluted with a Gaussian of variance σ^2.

Let the input space Λ be a metric space (e.g: $\mathbb{R}^d$) and $Z_n = \{(\lambda_i, y_i)\}_{1 \leq i \leq n} \subset \Lambda \times \mathbb{R}$ be the set of input-output couples independently and identically distributed according to an unknown probability distribution $\rho(\lambda, y) = \eta(\lambda)\psi(y|x)$. Moreover let $f(\lambda) = \int y d\psi(y|\lambda)$ be the function whose convoluted version $\phi = \mathcal{N}_{\sigma^2} * f$ we want to estimate. Thus we introduce the kernel machinery, namely the Reproducing Kernel Hilbert Space $\mathcal{H}$ associated to a given translation-invariant kernel function $K(\lambda, \mu) = k(\lambda - \mu) : \Lambda \times \Lambda \to \mathbb{R}$ on the input space Λ (e.g: the gaussian kernel) and the associated dot product $\cdot^\top \cdot$ such that $K_{\lambda_0}^\top f = f(\lambda_0)$ for any $f \in \mathcal{H}$ and $K_{\lambda_0}(\mu) \equiv k(\mu - \lambda_0) \in \mathcal{H}$ for any $\lambda_0 \in \Lambda$. Then the convolution operator $G_{\sigma^2} \in \mathcal{L}(\mathcal{H})$ is defined as

$$G_{\sigma^2} = \int_X \mathcal{F}^{-1} \left\{ \frac{\hat{\mathcal{N}_{\sigma^2}}}{\hat{K}} \right\} (t - \tau) K_t K_\tau^\top dt d\tau \tag{1}$$

where $\mathcal{F}$ is the Fourier transform operator, $\hat{\mathcal{N}_{\sigma^2}} = \mathcal{F}(\mathcal{N}_{\sigma^2})$ and $\hat{K} = \mathcal{F}(K)$. The operator acts in the following way $K_\lambda^\top G_{\sigma^2} f = (\mathcal{N}_{\sigma^2} * f)(\lambda)$.

Instead of searching the less expensive function which best approximates a given set of points according to the standard KRLS [6] approach, in this setting we aim at the function whose deconvoluted version best approximates the points, which is defined as follows:

$$\tilde{\phi} = \arg\min_{\varphi \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (K_{\lambda_i}^\top G_{\sigma^2}^{-1} \varphi - y_i)^2 + \nu \|\varphi\|_{\mathcal{H}}^2 \tag{2}$$

where $\tilde{\phi}$ is the learned version of ϕ. This equation can be rewritten as $\tilde{\phi} = \arg\min_{\varphi \in \mathcal{H}} \mathcal{L}$ with $\mathcal{L} = \varphi^\top g_n^\top g_n \varphi - 2 g_n^\top Y + Y^\top Y + \nu \varphi^\top \varphi$, $g_n = s_n G_{\sigma^2}^{-1}$ and $s_n = (K_{\lambda_1}, \ldots, K_{\lambda_n})^\top : \mathcal{H} \to \mathbb{R}^n$ and $Y = (y_1, \ldots, y_n)^\top$

The functional of the problem 2 is convex in φ, so, imposing its first derivative to zero, we obtain $\tilde{\phi} = \left(g_n^\top g_n + \nu\right)^{-1} g_n^\top Y = g_n^\top \left(G + \nu\right)^{-1} Y$ with $G \in \mathbb{R}^{n \times n}$, $(G)_{ij} = K_{\lambda_i}^\top G_{\sigma^2}^{-2} K_{\lambda_j}$, where the second equality is a consequence of spectral calculus. The smoothed learned function $\tilde{\phi}(\lambda)$ is expressed in closed finite form

$$\tilde{\phi}(\lambda) = K_\lambda^\top \tilde{\phi} = a_\lambda^\top (G + \nu)^{-1} Y \tag{3}$$

with $(a_\lambda) = (K_\lambda^\top G_{\sigma^2}^{-1} K_{\lambda_1}, \ldots, K_\lambda^\top G_{\sigma^2}^{-1} K_{\lambda_n})^\top \in \mathbb{R}^n$. The dot products can be calculated analytically. We note that when K is a gaussian kernel $K(\mu, \lambda) = C\mathcal{N}_{\theta^2}(\mu - \lambda)$ of variance θ^2 and a constant $C > 0$, we have $K_\lambda^\top G_{\sigma^2}^{-1} K_\mu = C\mathcal{N}_{\theta^2 - \sigma^2}(\lambda - \mu)$ and $K_\lambda^\top G_{\sigma^2}^{-2} K_\mu = C\mathcal{N}_{\theta^2 - 2\sigma^2}(\lambda - \mu)$. We stress the fact that, in order to have two bounded dot products, we should choose $\theta^2 > 2\sigma^2$.

3 Experiments

In this section, the proposed method was extensively evaluated on the MNIST data set [9], which is a collection of 60,000 images of handwritten digits. In

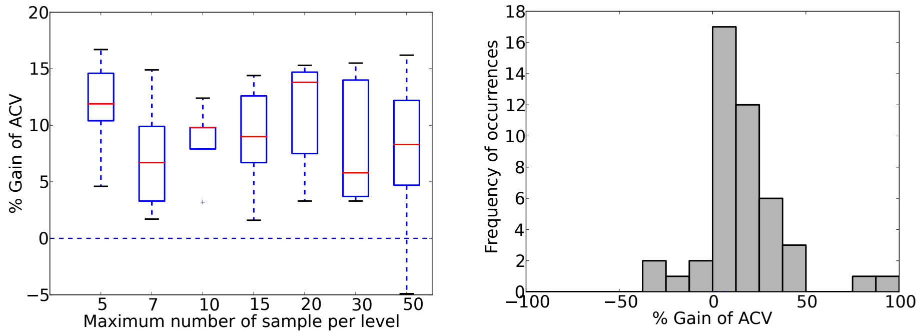

Fig. 1: Left: box and whisker plot of the $\mathbf{\Delta}\%$ accuracy gain/loss of ACV w.r.t. GS, computed varying the characteristic dimension σ while keeping fixed the maximum number of sample per level $\hat{n}$, using 50 runs per couple $(\sigma, \hat{n})$. Right: histogram of $\mathbf{\Delta}\%$ accuracy for all the possible combinations of the classes with parameters $\hat{n} = 5$ and $\sigma = 0.5$ using 20 runs per couple.

all the following experiments, 1000 training images and 1000 validation images per class were randomly sampled from the set of 50,000 training images and 1000 images per class were sampled from the set of 10,000 test images. In order to show the core idea, a very simple representations of the subregions, the samples and the global minimization function had been chosen. The region of interest L is represented as an hyperbox. The sampling function $\mathit{Sample}(L, n)$ selects n equispaced points with uniform probability in L. $\mathit{GlobalMinimum}(\phi, L)$ uniformly samples the learned function ϕ in L and then it performs a gradient descent optimization in order to find its minimum. For the implementation of the $\mathit{SmoothKRLS}(Z, \sigma)$, a Gaussian Kernel with θ^2 variance is chosen. The choice of θ is fully automatic and it is done using the mean of the distances of the points from their associated Nearest Neighbor points in the Z set. In the same way, the regularization parameter ν for the *Smooth KRLS* function is selected automatically using a Leave-One-Out Cross Validation computed over the set of points Z. For classification purposes, SVM were used. In this case, the two parameters to tune are the weight of the Slack Variables C and the parameter γ of the Gaussian Kernel: $K(x, y) = \exp^{-\gamma\|x-y\|^2}$ where γ is the variance of given Gaussian distribution. During all the experiments, the starting hyperbox in the parameters space was set to $(\log C, \log \gamma) \in L = [-6, 0] \times [0, 6]$ The comparison between the ACV and the standard GS has been done using the same setting: once the ACV algorithm is run, then the GS samples the same number of points in the parameters space over an equispaced grid, in order to maintain the results comparable between the two different methods. For all experiments, the accuracy measure $\mathbf{\Delta}\%$ is computed as follows: $\mathbf{\Delta}\% = \dfrac{100}{N} \sum_{i=1}^{N} \dfrac{e_i^s - e_i^g}{e_i^g}$ where e_i^s and e_i^g are the error of the i-th test respectively of ACV and GS method and N are the number of repetitions of the tests. The $\mathbf{\Delta}\%$ measure indicates how much the ACV performed with the respect to the maximum error of the GS method: for positive values the ACV outperformed GS, and viceversa. Figure 1(left) shows binary classification performances between a subset of digits. The box and whiskers plot indicates that ACV outperformed GS, especially in the setting with few samples per level: this means that, despite the small number of sample points used to estimate the function, the presented method refines the error function in the correct subspace L still obtaining generalized models. In Figure 1(right), the performances of binary classification for all the possible combinations of classes are shown: in this setting the ACV outperforms GS in 88.9% of the cases, with a mean accuracy of 16.7%(± 29.6%). This behavior confirm the stability of the presented method especially in finding stable minima of the error function without loosing generality over independent sets of data.

4 Conclusions

In this paper a fully automated approach for model selection is presented and validated. The *Adaptive Optimization for Cross Validation* learns the error function adaptively by sampling and refining the approximation only on the regions of stable minima with no overheading in parameters choice compared to standard GS approach. The effectiveness and the efficiency of this novel approach are showed by presenting experiments of binary classification tasks over the MNIST dataset. The proposed algorithm is able to find stable minima with few evaluations of the error function, outperforming the standard Grid Search. Future works will include an improvement and refinement of the presented method, employing the theoretical bounds of the Smooth KRLS routine.

References

[1] Adankon et al.: New formulation of svm for model selection. In: IJCNN. IEEE (2006)

[2] An, S., Liu, W., Venkatesh, S.: Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression. Pattern Recognition 40(8), 2154–2162 (2007)

[3] Arlot, S., Celisse, A.: A survey of cross-validation procedures for model selection. Statistics Surveys 4, 40–79 (2010), http://projecteuclid.org/euclid.ssu/1268143839

[4] Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine learning 46(1), 131–159 (2002)

[5] Cherkassky, V., Ma, Y.: Practical selection of svm parameters and noise estimation for svm regression. Neural Networks 17(1), 113–126 (2004)

[6] De Vito et al.: Learning from examples as an inverse problem. JMLR 6(1), 883 (2006)

[7] Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference, and prediction, vol. 66. Springer Verlag (Dec 2009)

[8] Lindeberg et al.: Linear scale-space. Journal of Mathematical (1994)

[9] The MNIST database of handwritten digits. Http://yann.lecun.com/exdb/mnist/ (1998)

[10] Osborne, M.a.a.: Gaussian processes for global optimization. In: LION3 (2009)

[11] Pahikkala, T., Boberg, J., Salakoski, T.: Fast n-fold cross-validation for regularized least-squares. In: Scandinavian conference on artificial intelligence (SCAI 2006) (2006)

[12] Smale, S., Zhou, D.X.: Learning Theory Estimates via Integral Operators and Their Approximations. Constructive Approximation 26(2), 153–172 (Mar 2007)

[13] Wolpert, D.: The lack of a priori distinctions between learning algorithms. Neural Computation 8(7), 1341–1390 (1996)