Data Set Selection

Doudou LaLoudouana* **and Mambobo Bonouliqui Tarare**
Lupano Tecallonou Center
Selacie, GUANA
doudoula3@hotmail.com, fuzzybear@yahoo.com

Abstract

We introduce the community to a new construction principle whose practical implications are very broad. Central to this research is the idea to improve the presentation of algorithms in the literature and to make them more appealing. We define a new notion of capacity for data sets and derive a methodology for selecting from them. The experiments show that even for not so good algorithms, you can show that they are significantly better than all the others. We give some experimental results, which are very promising.

1 Introduction

Learning is a marvellous subject. A year spent in artificial intelligence is enough to make one believe in God. Unfortunately, so far it has been handled only from a particular one-sided point of view. The VC-theory known only by some people does not offer what we would be in right to ask from such a theory: we want good bound for our algorithms. We offer with this article a brand new approach that allows you to present your algorithm in a much more principled and rigorous way that has been done before. Many researchers, especially in publications at NIPS, have tried to show when their algorithms are better (in some sense) than some other given set of algorithms. To do this they have employed techniques of data set selection. It is strange then that learning theorists, as they call themselves, over the last 50 years have concentrated on the model selection problem and not the data selection problem which is what people actually do. The two problems can in some sense be seen as the dual of each other, but it is not because you solve one that you can solve the other one. And vice-versa. In this article we lay down the foundations and introduce to the community of machine learning peers and other engineers a new induction principle: structural dataset minimization. Essentially we begin to formalize the sometimes ad hoc engineering approach of the selection procedure that everyone already practiced. In doing so we find concrete bounds for when the data selected really is better than other datasets and implement less ad hoc algorithms for finding such datasets. We show our approach outperforms the classical approach.

The structure of the paper contrarily to its content follows a classical trend: section 1 presents some nice bounds you can use in lots of situations, section 2 shows how to use these bounds by designing new algorithms. Section 3 describe some experiments which, of course, are good.[1] Section 4 concludes the article with smart thoughts and future work we will do.[2]

2 Bounds

Let us introduce some notations. Assume a researcher has invented an algorithm A^* and he wishes to show that his pride and joy is superior with respect to some loss function ℓ to a

[1] Contrarily to other people, we have put all of our experiments in this paper, even the bad ones, but well, we did not get any bad ones.

[2] If (and only if) the paper is accepted.

given fixed set of algorithms[3] $A_1, .., A_n$ that other researchers have made. For this purpose, the researcher selects some data sets using what is called an *empirical data set minimization* method. The latter consists in taking some fixed set of data sets $D_1, .., D_d$ and find a data set D^* in $D_1, .., D_d$ so that:

$$\ell(A^*, D^*) < \min_{i=1,..,n} \ell(A_i, D^*)$$

Note that this problem is ill-posed. A natural generalization would be to find more than one data set in which your algorithm performs well but this is a difficult problem that has not been solved so far by the community. Current efforts in solving this problem have focussed on producing more artificial data sets rather than algorithms to achieve this goal.

We have the following theorem:

Theorem 1 *Let $\mathcal{D}$ be a set of training sets, then assume that the space of algorithms is endowed with a fixed distribution $\mathbb{P}$ (which could be anything a priori), then with probability $1-\eta$ over a sampling on the algorithm, and for all $\gamma > 0$, we have:*

$$\forall D \in \mathcal{D}, \quad R_{gen}{}^{A}[D] \leq R_{emp_{\gamma}}{}^{A}(D) + O\left(\sqrt{\frac{\Phi(\mathcal{D})}{m}\log(1/\eta)}\right)$$

where $\Phi(\mathcal{D})$ is the capacity of the set of training set defined as:

$$\begin{aligned}\Phi(\mathcal{D}) = \max\{m \; s.t. \; \exists A_1, .., A_m \, algorithms \; s.t. \; \forall (r_{11}, .., r_{ij}, .., r_{mm}) \in [0,1]^{m(m-1)/2}, \\ \exists D \in \mathcal{D}, \quad \forall i \neq j |\ell(D, A_i) - \ell(D, A_j)| \leq r_{ij}\} \quad (1)\end{aligned}$$

We are proud now to supply the following elegant proof.

Proof: Let us denote by m the number of points in the training set, we see that introducing a ghost algorithm A':

$$\mathbb{P}_A\left(\sup_{D\in\mathcal{D}} \left|R_{\text{emp}}{}^{A}[D] - R_{\text{gen}}{}^{A}[D]\right| > \epsilon\right) \leq \mathbb{P}_{A,A'}\left(\sup_{D\in\mathcal{D}} \left|R_{\text{emp}}{}^{A}[D] - R_{\text{emp}}{}^{A'}[D]\right| > \epsilon\right)$$

which is trivially insensitive to permutations so that we can condition over the algorithm A and A'. Then we also have the right to play with the swapping permutation as it has been done in the theoretical but practically not used VC framework, which means that we work only with the values of (σ_1, σ_2). After some more states which we admit for brevity, this leads to the vanishing of the supremum. We are then left with a sum of two random variables whose sum can be controlled using the Bennett-bernstein inequality. This is here where the tricky part begins. It is known that averaging over two random variables does not gives you a good control of their expectation. But this can be overcome if we consider many exact replica of the first two variables. Then we have plenty of them, as much as we want! And we can then control the expectation because now the value of m is big. We call this trick, the replica trick. Note the replica trick has been used many times in the invention of algorithm: when exploring the space $\mathcal{A}$ of all possible algorithms, the same algorithm has been visited many times but with negligible variations so that if you use an ϵ-insensitive loss functions, these algorithms appear to be equivalent.[4] □

The theorem we just proved should be considered as the dual of the theorem of Vapnik and Chervonenkis. And this should be the case because it is just the dual of it. And we believe this is the more natural setting for your every day design of algorithms. Maybe our approach is complementary to the one of Vladimir and Alexei but we are one step forward because we can infer/compute the probability over the data sets just by looking at the UCI repository database. Just try to do the same with your set of functions and we will talk. Anyway, we insist that our approach shares a lot of common properties with the classical VC framework. One of them is that unfortunately we cannot say much but we try to, or say differently we have our own "no free lunch" theorem but we try to forget it. Here is our no free brunch!! theorem:

[3] Normally, a small set so that he does not have to do too many experiments.

[4] In fact, embedding nips papers into a vector space, we found a big cluster where all the points were close to each other at a distance less than 0.05 which is the classical significant threshold used in statistics. We ran k-means 50 times but kept coming up with the same single big cluster.

Theorem 2 (No Free Brunch!!) *The generalization error of two datasets for all algorithm is the same:*

$$E_A[{R_{gen}}^A[D]] = E_A[{R_{gen}}^A[D']]$$

so that there is no better dataset than the one you already have.

The consequences of the theorem are very hard: it means that if you don't do well, then you are not very skilled. We still have not worked out what the researchers had for breakfast (of course, in this remark we retain some hilarity). This leads to the natural consequence that to say anything one should restrict the set of algorithms to some natural set, in a companion paper we prove that the set of datasets restricted to all Bayesian algorithms has infinite dimension. The same is true for the set of all kernel algorithms if you leave the the following free parameters: loss function (hinge loss, pound loss, epsilon-insensitive loss, ℓ_1, ℓ_2, ℓ_2+, Huber, hyperbolic tangent, Bregman, Breiman, etc.), regularizer (RKHS, $||w||^2$, $\sum_i \alpha_i^2$, $\sum_i \alpha_i$, KL divergence, relative entropy, $\log(\max_i \alpha_i)$ etc.) and the set of functions (we do not list here all possible kernels, this will future work to examine the kernel set selection phenomena) and the optimization procedure (perceptron, gradient descent, chunking, quadratic, linear, stochastic, simulated annealing, analytic, random)[5].

The second major contribution is that our bound has exposed one of the major problems of empirical dataset minimization: the previous de facto method over ten years of nips[6] publications. Clearly, the second term in the bound (which is the confidence interval over the set of datasets) is not taken into account. This leads us to proposing a new construction principle called the Structural Data Sets Minimization (SDSM).

3 Structural Data Sets Minimization

In order to appreciate the following section, we ask the reader for a little patience and to concentrate a little. Assume you have a increasing set of datasets $\mathcal{D}_1, .., \mathcal{D}_k$ (e.g USPS, NIST, REUTERS 1, REUTERS 2, ...) which are roughly included in each other:

$$\mathcal{D}_1 \subset \mathcal{D}_2 \subset \ldots \subset \mathcal{D}_k$$

Then we would like to apply the theorem of the previous section and choose the best datasets for our algorithms (i.e. the one that will be used in the paper presenting these potentially new algorithms). Using the union bound, we found that with probability $1 - \sum_{i=1}^k \eta_i$, for all $D \in \cup \mathcal{D}_i$:

$$ {R_{\text{gen}}}^A[D] \leq \underbrace{\min_{D \in \mathcal{D}_i} {R_{\text{emp}}}_\gamma^A(D) + O\left(\sqrt{\frac{\Phi(\mathcal{D}_i)}{m}\log(1/\eta_i)}\right)}_{\Psi(i)} \quad (2)$$

So that we can pick the i^* such that $\Psi(i)$ is minimized over i and choose then the best data set on $\mathcal{D}_{i^*}$. The consistency of this procedure can be ensured by a theorem.

This section means that to find the best data set for your algorithm, you can consider many datasets and compute their capacity and then pick the best one not only in terms of empirical errors (which is strangely often called test error in many papers) but also in terms of this capacity.

Here is a nice picture (Figure 1).

4 Algorithm Ordering Machine

We have now prescribed a formal method for choosing between real datasets. However, we don't really know the capacity of these real world datasets, only approximations can be calculated (although empirical observations show that most published algorithms do well, so the capacity must be quite high). For that reason, we suggest to use toy problems. It turns out that we can prescribe an efficient algorithm for searching for the best toy problem for a given algorithm A.

[5] We do not cite the individual papers, we refer the interested reader to the NIPS volumes.

[6] www.nips.com

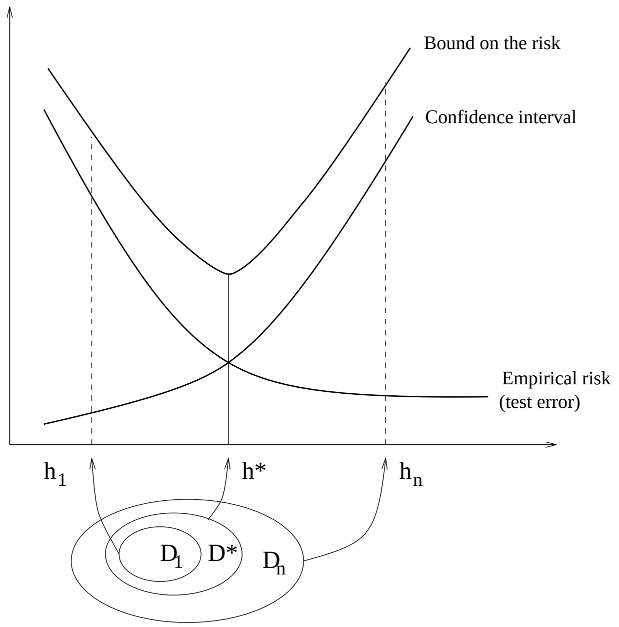

Figure 1: Graphical depiction of the strucural data sets minimization construction principle.

The highly successful and efficient algorithm prescribed is the following. Once again, assume a researcher has invented[7] an algorithm A^* and he wishes to show that it is superior with (dis)respect to some loss function ℓ to a given fixed set of algorithms $A_1, .., A_n$ that other researchers have made. Let us try to choose a fixed class of artificial datasets $\mathcal{D}$, for example artificial data that is generated by a mixture of k Gaussians with covariance matrixes $c_{1,..k}$ in a d dimensional space with a noise model e and f extra noisy features. Let us define $w = (c, d, e, f)$, we have the following theorem:

Theorem 3 *The capacity $\Phi(\mathcal{D})$ is bounded by:*

$$\Phi(\mathcal{D}) \leq \min \left(R^2 \|w\|^2, n\right)$$

where n is the number of parameters (sum of dimensions of the vectors c, d, e and f) and R is the largest possible value of any coordinates of any vectors in the data sets.

Some people would argue that this theorem is valid only for data sets parameterized by gaussian distribution, but actually it can model a lot of real life problems. The interested reader can read the literature about gaussian processes.

Now, the task is to optimize these parameters such that the algorithm A appears much better than the other ones. Let us choose A=Bruto, A_1=SVM, A_2=MARS, A_3=k-NN, A_4=C4.5 and we embed this set of algorithms with a uniform distribution to ensure no bias. This can be done with the following minimization:

$$\min_{c,d,e,f} \Phi(\mathcal{D}_{c,d,e,f})$$

$$\text{subject to :} \qquad {R_{\text{emp}}}^{A}[D] < {R_{\text{emp}}}^{A_i}[D] - \epsilon, \quad i = 1, .., 4$$

which is equivalent to:

$$\min_{w=(c,d,e,f)} \|w\|^2$$

[7] i.e, slightly altered someone else's.

Algorithms	Without noise	With noise
Linear SVM	0.5 (0.005)	0.5 (0.01)
Poly-2 SVM	0.5 (0.1)	0.5 (0.2
Poly-5 SVM	0.7 (0.2)	0.7 (0.1)
Poly-10 SVM	0.9 (0.8)	0.8 (0.6)
Mars	0.2 (0.2)	0.4 (0.1)
Bruto	**0.001 (0.00009)**	**0.002 (0.0001)**

Figure 2: Results for $w = (0.00001, 50000154839, 34, 3.14159, 2.755, 1, 2, 3, 4, 5, 6, 7, 8, 9, -1, -2, -3...)$. w corresponds to the parameters of the generated data set. Unfortunately, we have difficulties interpreting w.

$$\text{subject to :} \qquad {R_{\text{emp}}}^{A}[D] < {R_{\text{emp}}}^{A_i}[D] - \epsilon, \quad i = 1, .., 4$$

If you wish to find a dataset where your algorithm does not achieve 0% test error you can easily generalize this algorithm to the linearly inseparable case by introducing slack variables ξ_i.

The closeness to SVM is striking. Note that we are maximizing the margin between the algorithms to ensure a paper is accepted. The relation with statistical tests is open and has not been analyzed rigorously yet, but an upcoming work is statistical test selection so that even with small margin you can have a strong result.

5 Experiments

We proceed to prove our methodology by showing we can always find a good solution even for bad algorithms So we proceed with the example given above. We must admit that it has already been done in the literature but we provide a deeper analysis and also better margin. In table 2, we present the results we got with the best value for w.

Note that it is very difficult to discriminate between a linear SVM and a poly-2 SVM, both of them seem to have a similar behavior and we were not able to worsen the results of the poly-2 SVM although it would have been nice. Poly-2 SVM performs well on a large number of data sets so the optimization was difficult, this may explain why we got strange value for w. On the other hand, it is quite clear in the table that Bruto is much better than all the other algorithms even much better than MARS although MARS has some common properties with Bruto. Thus our algorithm was able to discriminate between ϵ distances in the space of $\mathcal{A}$. We omit our other experiments for brevity but the results were good.

6 Conclusion

We will now reiterate what we said before, we repeat the abstract and introduction. The problem with any unwritten law is that you don't know where to go to erase it. This is the same for other matters by the way. Consider for instance the notations, it is assumed that m or ℓ always refer to the number of training examples. Sometimes, it is n also but this occurs mainly when the authors are new in the field. Note that we also are new in the field but we did not use n. On the other hand, we have handled quite in a nice way the use of greek letters. Anyway, the question we are discussing right now is to know when and how to stop an unwritten law. We believe this could be the place and the time, and, as a mark of courage, we will not reiterate the introduction. This may then sound weird that we, as outsiders, put a stone in the sea of nips paper. We do not know the hydrodynamics laws of such a sea. We do not who discovered water but we're pretty sure that it wasn't a fish. Not even a big fish with a latin name.

Let us stop being polemic for a while and come back to our contribution. Central to our new research is the idea to improve the presentation of algorithms in literature and to make them more appealing. We defined a new notion of capacity for data sets and derived a methodology. The experiments showed that even for not so good algorithms, you can show that they are significantly better than all the other ones. The message is strong and may not be understood at a first reading so we insist here to avoid any confusion: we emplore all researchers to dig out their old failed algorithms and turn them into successful ones.

To be complete, we present in this last paragraph the future works we plan to do one day. We will make the link between dataset selection and the human neural information processing which so far researchers have shown happens, in females, in the human brain[8]. We will consider whether dataset selection is implemented via chemical stimulation of the neurons, possibly in the hippycampus. In humans it could consist of, when failing to learn a task, buggering off and learning something else instead[9].

At last, we would like to say a mild word of caution. We hope that the community learns from this break-through and applies our methodology in their future research or they will get left behind: our algorithms will far outperform theirs.[10]

References

[1] R. Shapire, Y. Freund, P. Bartlett, and W.S. Lee. *Bushing the margin: A new explanation for the effectiveness of U.S voting methods. The Anals of Statistics*, 1998.

[2] N. Cristianini and J. Shawe-Taylor. *Data Set Selection for Dummies.* Egham University Press, 2005.

[3] P. Bartlett. *The Sample Complexity of Pattern Classification with Neural Networks: the Number of Citations is More Important than the Number of Readers. Biowulf Transactions* , 1998.

[4] D. MacKay. *I did it my way. Self published*, 2002.

[5] G. Fung and O. L. Mangasarian. Data Selection for Support Vector Machine Classifiers. Proceedings of KDD'2000, 2000.

[6] J. Friedman, T. Hastie, and R. Tibshirani. *Data sets for additive models: a statistical view of bragging and boasting.* Technical report, Technical Report, Department of Statistics, Stanford University., 1998.

[7] C.M. Bishop. *Neural Not works for Pattern Recognition*, Oxford University Press, 1995

[8] D.H. Wolpert, *The lack of a priori distinctions between learning algorithm research*, Neutral Computation, 1996.

[9] V.N. Vapnik and A.Y Chervonenkis and S. Barnhill, *The VC Way: Investment Secrets from the Wizards of Venture Capital*, Biowulf Publishing (now out of print) , 2001.

[8] Note that for males, some people conjectures it should be in some unknown other place, others yet conjecture it doesn't exist at all. We will refer to this as the D-spot and note that at the least it is very hard to find, at least if the guy is fat.

[9] Notice how many people learn to juggle around exam time.

[10] Finding the occasional straw of truth awash in a great ocean of confusion and bamboozle requires intelligence, vigilance, dedication and courage. But if we don't practice these tough habits of thought, we cannot hope to solve the truly serious problems that face us – and we risk becoming a nation of suckers, up for grabs by the next charlatan who comes along.